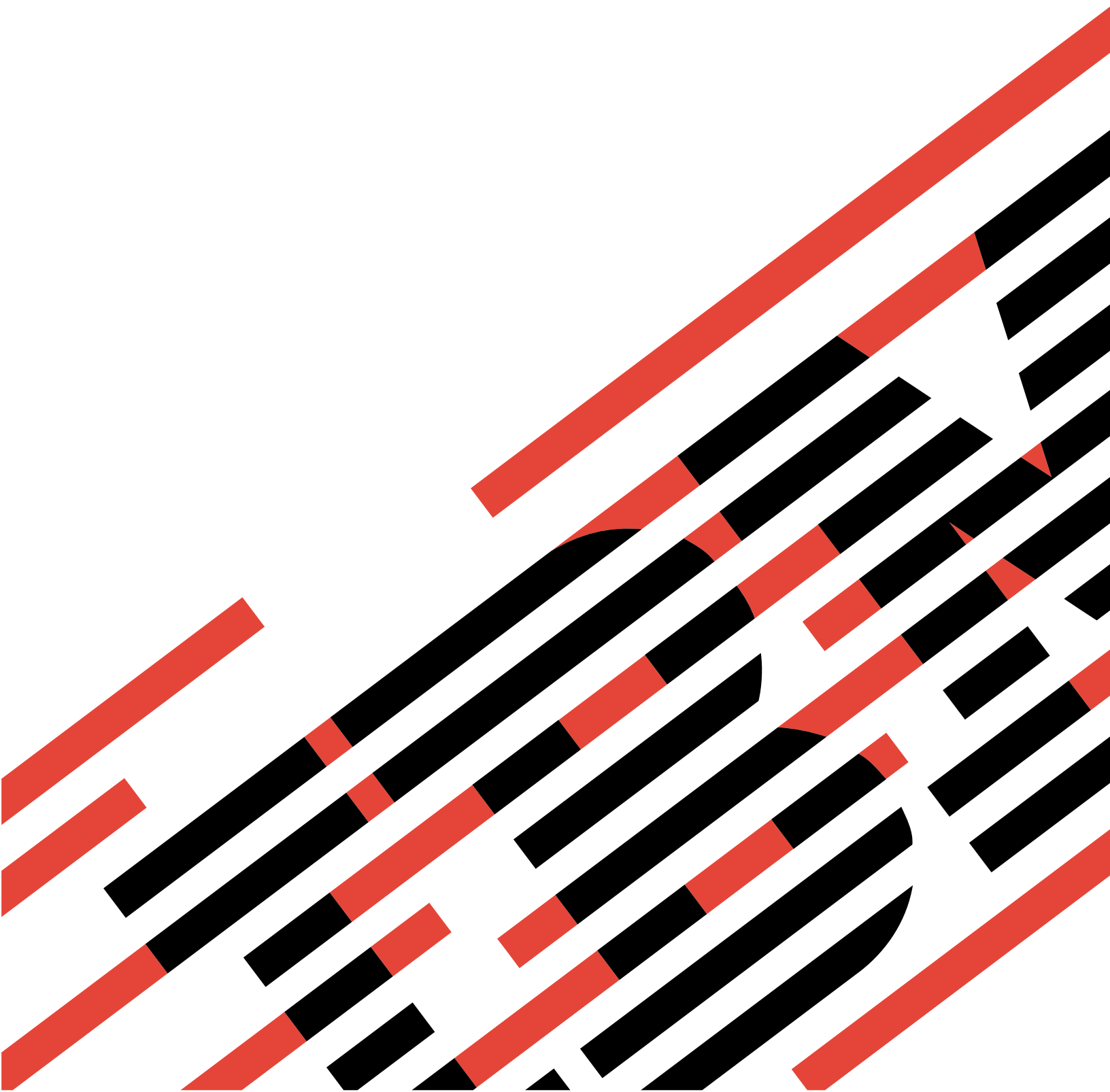


IBM

@server

iSeries

성능





@server

iSeries

성능

목차

성능	1
V5R2의 새로운 사항	2
새로운 사항: 모니터	2
새로운 사항: 콜렉션 서비스	3
새로운 사항: 성능 분석 툴 사용권 프로그램	4
새로운 사항: Performance Management/400	6
새로운 사항: Performance Explorer	7
이 주제 인쇄	9
성능 계획	10
시스템 벤치마크 설정	11
시스템 확장 시기와 방법 결정	12
Capacity Upgrade on Demand	13
Capacity Upgrade on Demand 개념	13
Capacity Upgrade on Demand 시험 기간	14
Capacity Upgrade on Demand의 소프트웨어 사용권 부여 고려사항	14
Processors On Demand 활성화 코드	15
Electronic Service Agent 요구사항	15
Capacity Upgrade on Demand 준비	15
Capacity Upgrade on Demand에서의 용량 계획	16
프로세서 활성화 시기 결정	16
Capacity Upgrade on Demand 환경 설정	17
Capacity Upgrade on Demand 주문	18
IBM에 필수 제품 자료 송신	18
대기 프로세서 활성화	19
시험 기간 동안 대기 프로세서 활성화	19
영구적으로 대기 프로세서 활성화	20
시나리오: Capacity Upgrade on Demand	21
성능 관리 전략 선택	22
성능 관리를 위한 환경 설정	24
iSeries 성능 관리	25
성능 추적	25
성능 문제 연구	26
성능 문제 식별	27
일반적인 성능 문제 식별 및 분석	28
시스템 성능 자료 수집	28
시스템 자원 활용에 대한 정보 수집	29
어플리케이션의 성능에 대한 정보 수집	30
추적 자료 덤프	31
시나리오: 업그레이드 또는 마이그레이션 후 시스템 성능 개선	31
성능 자료 표시	32
성능 조정	33
기본 성능 조정	33
자동 성능 조정	35

e-business 성능 관리	36
클라이언트 성능	37
네트워크 성능	37
OS/400에서의 Java 성능	38
HTTP Server 성능	38
WebSphere 성능	39
성능 관리를 위한 어플리케이션	40
컬렉션 서비스	43
컬렉션 서비스 자료로 데이터베이스 파일 작성	46
기존 컬렉션 오브젝트로 데이터베이스 파일 작성	47
자료 컬렉션 사용자 정의	48
컬렉션 서비스의 시간대 고려사항	49
컬렉션 서비스의 사용자 정의 범주	50
컬렉션 프로그램 권장사항 및 요구사항	51
예: 사용자 정의 범주 구현	52
컬렉션 오브젝트 관리	59
사용자 정의 트랜잭션	60
C++ 예: 사용자 정의 트랜잭션을 컬렉션 서비스에 통합	61
Java 예: 사용자 정의 트랜잭션을 컬렉션 서비스에 통합	65
성능 자료 파일	68
시간 간격 자료가 들어 있는 성능 자료 파일	69
성능 자료 파일: 파일 약어	70
성능 자료 파일: 컬렉션 서비스 시스템 범주 및 파일 관계	71
성능 자료 파일: 구성 데이터베이스 파일의 필드 자료	73
성능 데이터베이스 파일: 추적 데이터베이스 파일의 필드 자료	73
iSeries Navigator 모니터	73
모니터 개념	74
모니터 구성	75
모니터 미터법	76
시나리오: iSeries Navigator 모니터	77
시나리오: 시스템 모니터	77
시나리오: 작업 모니터	79
시나리오: 메세지 모니터	80
그래프 이력	81
그래프 이력 개념	81
그래프 이력 사용	82
Performance Management/400	83
PM/400 개념	83
PM/400의 장점	84
PM/400e에 대한 운영 지원 서비스	84
PM/400 자료 수집 고려사항	85
PM/400 구성	85
Performance Management/400 활성화	86
PM/400 사용자 정의	94
PM/400 관리	96
PM/400 비활성화	97

PM/400 문의처 정보 변경	97
PM/400을 사용한 작업 스케줄링	98
PM/400 분석에서 항목 생략	99
PM/400 일시 정지	99
PM/400 상태 표시	100
PM/400 보고서 보기	100
성능 분석 툴	100
성능 분석 툴 개념	101
성능 분석 툴에 제공되는 기능	102
관리자 및 에이전트 피처 비교	102
성능 정보 표시	103
성능 분석 툴 설치 및 구성	104
성능 분석 툴 보고서	105
Performance Explorer	107
Performance Explorer 개념	108
Performance Explorer 정의	109
Performance Explorer 데이터베이스 파일	111
Performance Explorer 보고서	113
Performance Explorer의 장점	113
Performance Explorer 구성	114
Performance Explorer 종료	114
iDoctor for iSeries	115
PTDV(Performance Trace Data Visualizer)	116
성능 관리 API	116
OS/400 성능 명령에 대한 “작업”	117
Extended Adaptive Cache	117
Extended Adaptive Cache 개념	118
제한사항 및 고려사항 - Extended Adaptive Cache	119
Extended Adaptive Cache 시작	120
Extended Adaptive Cache Simulator	120
Extended Adaptive Cache 확보	120
iSeries용 작업부하 예측기	121
무선 iSeries Navigator	121
PATROL for iSeries(AS/400) - Predict	122
시나리오: 성능	122
관련 정보	122

성능

시스템 성능 관리에 얼마나 투자하고 계십니까? 기업의 요구사항은 생각했던 것보다 더 빠르게 변하고 있습니다. 이러한 기업 환경 변화에 효율적으로 대처하기 위해서는 시스템 역시 바뀌어야 합니다. 시스템 관리를 시간이 걸리는 작업으로 생각하기 쉽습니다. 그러나 시스템이 더 효율적으로 수행되기 때문에 투자는 곧 결과를 얻게 되고 이것이 다시 기업 활동에 반영됩니다. 변화가 계획되고 관리되기 때문에 효율적인 것입니다.

iSeries 서버의 성능 관리는 iSeries 작업 관리에 대한 충분한 이해를 요하는 복잡한 작업입니다. 시스템 성능에 영향을 주는 다양한 모든 프로세스에 대해 이해한다는 것이 경험이 없는 사용자에게는 어려울 수 있습니다. 성능 문제를 분석하려면 각각 요구사항과 지원되는 기능이 고유한 수많은 툴을 효율적으로 사용할 수 있어야 합니다. 성능 자료를 수집하여 분석한 경우라도 이러한 정보를 이용하는 차원에서는 위축될 수 있습니다.

이 주제에서는 성능 관리와 관련한 작업과 툴에 대해 안내합니다.

V5R2의 새로운 사항

이 주제에서는 이 릴리스에서 새로운 정보는 무엇이고 크게 변경된 내용은 어떤 것인지 소개합니다.

이 주제 인쇄

이 정보의 인쇄본을 원할 경우 여기로 가서 PDF를 인쇄하십시오.

성능 계획

iSeries 서버에 대한 성능 목표를 설정하면 성능 자료를 비교하기 위한 측정 가능한 성능 벤치마크를 보유할 수 있습니다. 이 주제에서는 이러한 벤치마크를 설정하는 방법과 차후 이를 사용하는 방법에 대해 설명합니다.

성능 관리를 위한 환경 설정

iSeries 서버에는 시스템 성능을 관리하기 위한 강력한 어플리케이션이 포함되어 있습니다. 그러나 고유한 기업 환경의 특정한 요구를 충족시키기 위해서는 이러한 어플리케이션을 적절히 구성해야 합니다. 이를 위해 성능 자료를 정기적으로 수집, 모니터 및 분석하기 위한 어플리케이션의 구성 방법을 배우십시오.

성능 관리

성능 관리는 현재 성능을 측정하고 경향을 인식하며, 일반 사용자 및 관리 요구사항(예: 응답 시간 또는 작업 처리량)을 충족시키도록 적절히 조정하는 등의 방법으로 사용자의 컴퓨터 시스템에 대한 활용도를 최적화하기 위해 필요합니다. 기업의 효율성을 제고하고 정상적인 기업 활동이 장기간 지연되는 것을 막기 위해 필요합니다. 따라서 성능 관리는 매일 수행해야 하는 작업 중 하나입니다.

성능 관리를 위한 어플리케이션

iSeries 시스템에서 성능을 관리하려면 여러 가지 특화된 어플리케이션을 사용해야 합니다. 이들 각 어플리케이션은 시스템 성능을 고유하게 파악합니다. 이 주제에서는 여러 가지 어플리케이션과 이들 각 어플리케이션의 용도에 대해 설명합니다.

시나리오: 성능

성능 관리에 대해 배울 수 있는 가장 좋은 방법 중 하나는 기업 환경에서 이들 어플리케이션이나 툴이 어떻게 사용되는지를 보여주는 예를 참조하는 것입니다. 이러한 예를 찾아 보십시오.

관련 정보

IBM^(R) related information에는 기술 정보, 노하우 및 “방법” 정보가 들어 있습니다.

주: 이 주제에는 코드 예가 들어 있습니다. 중요한 법적 고지사항에 관해 언급하는 코드 면책사항 관련 정보를 참조하십시오.

V5R2의 새로운 사항

다음은 이번 릴리스에서의 새로운 기능입니다.

- **iSeries Navigator 모니터**
새로운 모니터 지원과 기존 모니터에 대한 변경사항을 찾아 보십시오.
- **콜렉션 서비스**
성능 데이터베이스 파일에 대한 변경사항과 새 기능에 대해 읽으십시오.
- **성능 분석 툴**
보고서에 대한 변경사항을 읽으십시오.
- **Performance Management/400**
PM/400이 자동으로 자료를 수집하는 방식을 읽으십시오.
- **Performance Explorer**
Performance Explorer 데이터베이스 파일에 대한 변경사항과 새 기능에 대해 읽으십시오.
- **Capacity Upgrade on Demand**
Capacity Upgrade on Demand의 iSeries 지원에 대해 읽도록 하십시오.
- **e-business 성능 관리**
e-business 환경에서 iSeries 성능을 관리하기 위해 사용할 수 있는 타스크, 고려사항 및 자원에 대해 읽도록 하십시오.
- **주 기억장치 유사성**
프로세스 및 스택드가 메모리 및 프로세스 자원의 유사성을 개선할 수 있는 방법에 대해 읽으십시오.

새로운 사항: 모니터

다음은 이번 릴리스에서의 새로운 기능입니다.

- **파일 모니터:** 다중 플랫폼 환경을 관리할 수 있도록 하는 기능. iSeries Navigator에서 명령을 실행하고 파일을 모니터링하며 다중 플랫폼 종료점 시스템을 관리할 수 있습니다.
- **B2B 활동 모니터:** 지정된 통합 시스템에 관한 상세 정보와 함께 트랜잭션 자료를 보고 OS/400 명령을 자동으로 실행하여 전반적인 트랜잭션 계수나 트랜잭션 시간을 제어할 수 있도록 하는 활동 모니터.

- 이제 시스템 모니터 미터법에 지점 간 프로토콜과 연관된 정보가 들어 있습니다. CPU 활용 등록 정보에 몇 가지 미터법이 추가되었습니다. 이 새 미터법은 파일 시스템, 스트림 파일, 저널 조작 및 카운터 세트와 관련됩니다.

새로운 사항: 콜렉션 서비스

다음은 이번 릴리스에서의 새로운 기능입니다.

새 릴리스 설치

새 릴리스로 업그레이드할 때 성능 자료의 처리 방법에 대해 여러 가지를 선택할 수 있습니다. 계속 CVTTPFRDTA(성능 자료 변환) 명령을 사용하여 콜렉션 라이브러리의 자료를 변환할 수 있습니다. 새로운 자동 자료 콜렉션 지원을 사용할 수 있습니다. 성능 데이터베이스 파일을 변환하지 않은 경우와, 콜렉션 서비스가 시작되어 사용자가 데이터베이스 파일을 자동으로 작성하는 옵션을 지정한 경우, 이전 릴리스 파일에 대한 성능 라이브러리가 작성됩니다. 성능 데이터베이스 파일은 이 라이브러리로 이동됩니다. 이 조치로 새 파일이 작성되고 기존 자료 파일을 이전 릴리스 레벨로부터 보호할 수 있습니다. 마지막으로 성능 데이터베이스 파일을 변환하지 않고 콜렉션 서비스를 실행하지 않은 경우 DLTF(파일 삭제) 명령(DLTF 라이브러리/QAPM*)을 사용하여 QAPMxxxx 파일을 삭제할 수 있습니다.

사용자 정의 범주

V5R2 이전에는 모든 성능 콜렉션 범주가 사전 정의되었고 모든 자료 콜렉션 프로그램이 시스템에 제공되었습니다. V5R2에서부터 사용자 자신의 성능 콜렉션 범주를 정의할 수 있게 되었습니다. 사용자 정의 성능 콜렉션 범주로 다음을 수행할 수 있습니다.

- 새로운 성능 콜렉션 범주를 콜렉터에 추가합니다. QypsRegCollectorDataCategory(콜렉터 자료 범주 등록)가 사용자 정의 자료 범주를 중앙 관리의 콜렉션 서비스 기능 중 하나 이상의 콜렉터 정의에 추가합니다.
- 자료 콜렉션을 위해 자료 콜렉션 프로그램을 새로운 범주에 제공합니다.
- 범주에 대해 수집된 자료를 관리 콜렉션 오브젝트에 저장합니다.
- 시스템 정의 범주에서와 같은 방식으로 범주에 대해 자료 콜렉션을 스케줄하여 실행합니다.
- 관리 콜렉션 오브젝트 API를 사용하여 관리 콜렉션 오브젝트에 있는 새 범주에 대해 콜렉션된 자료에 액세스합니다.

Domino 및 HTTP server(Apache로 구동)는 이 피처를 활용하여 자신의 성능 자료를 콜렉션 서비스에 통합합니다.

사용자 정의 트랜잭션

V5R2에서부터 콜렉션 서비스는 Performance Explorer 트랜잭션 경계 API를 활용하여 어플리케이션 사용자에게 자신의 트랜잭션을 정의할 수 있는 능력을 부여합니다. 콜렉션 서비스 자료 범주는 USRTNS이고 사용자 정의 트랜잭션과 연관된 성능 데이터베이스 파일은 QAPMUSRTNS입니다. 이러한 API에는 다음이 있습니다.

- 트랜잭션 시작
- 트랜잭션 종료

성능 데이터베이스 파일

QAPMAPPN을 제외한 모든 데이터베이스 파일은 이제 대화식입니다. 즉 열의 텍스트를 알파벳순으로 정렬할 수 있고 표에서 특정 단어를 탐색할 수 있습니다.

다음 표는 새롭게 변경된 데이터베이스 파일을 보여줍니다.

데이터베이스 파일	설명
QAPMJOBWT	작업, 태스크 및 스레드 대기 조건 자료를 포함하는 새 파일.
QAPMJOBWTD	QAPMJOBWT 파일에 있는 카운터 세트의 설명이 들어 있는 새 파일.
QAPMHTTPB	IBM HTTP Server(Apache로 구동)에 대한 기본 자료가 들어 있는 새 파일.
QAPMHTTPD	IBM HTTP Server(Apache로 구동)에 대한 상세 자료가 들어 있는 새 파일.
QAPMDOMINO	iSeries용 Domino에 대한 자료가 들어 있는 새 파일.
QAPMPPP	지점 간 프로토콜에 대한 자료가 들어 있는 새 파일.
QAPMUSRTNS	사용자 정의 트랜잭션에 대한 자료가 들어 있는 새 파일.
QAPMSYSTEM	파일 시스템 카운터 및 저널 카운터를 지원하는 새 필드.
QAPMSYS	단일 시스템의 여러 파티션에 있는 시스템 시계 사이에 차이가 있는지 판별할 수 있는 방법을 제공하는 새 필드.
QAPMECL	지원되지 않는 프로토콜에서 삭제된 프레임의 수를 보고하는 새 필드.
QAPMETH	지원되지 않는 프로토콜에서 삭제된 프레임의 수를 보고하는 새 필드.
QAPMCONF	새 레코드 키.
QAPMIOPD	I/O 어댑터 자료를 지원하는 새 필드.
QAPMMIOP	처리 시간을 지원하는 새 필드.
QAPMJOBMI	파일 시스템 카운터 및 저널 카운터를 지원하는 새 필드.
QAPMDISK	독립 ASP 관련 변경을 지원하는 새 필드 및 변경된 필드.
QAPMJOBOS	파일 시스템 카운터를 지원하는 새 필드.

새로운 사항: 성능 분석 툴 사용권 프로그램

다음은 이번 릴리스에서의 새로운 기능입니다.

- 이 릴리스에서 BEST/1은 성능 분석 툴에서 철수되었습니다. 용량 요구사항을 조정할 때 고려할 수 있는 대안으로 IBM의 작업부하 예측기(PM/400에 통합되어 있음)와, 분산 시스템용으로 BMC의 PATROL for iSeries(AS/400) - Predict 제품이 있습니다. 다음 명령이나 메뉴 옵션에 대한 지원이 V5R2의 성능 분석 툴 사용권 프로그램에서 취소됩니다.
 - Start BEST/1(BEST/1 시작)
 - ANZBESTMDL(BEST/1 모델 분석)
 - CRTBESTMDL(BEST/1 모델 작성)
 - CVTMDLSYSF(MDLSYS 파일 변환)

- DLTBESTMDL(BEST/1 모델 삭제)
- PRTBESTF(BEST/1 파일 인쇄)
- QCYCHKV(지정된 성능 자료 멤버의 버전 검사)
- QCYCVTBD(System/36 파일을 MDLSYS에서 BEST/1로 변환)
- QCYFLDT(지정된 성능 자료 멤버의 최초와 최종 날짜 찾기)
- QCYRSNDM(이전 계층의 메시지를 수신하여 호출 계층에 재송신)
- MDLSYS(MDLSYS 시작)
- 용량 계획/모델링이 PERFORM 메뉴에서 제거되었습니다.
- DSPPFRTA(성능 자료 표시) 명령과 연관된 기능을 iSeries Navigator에서 성능 분석 툴 플러그 인으로 사용할 수 있습니다. DSPPFRTA 명령과 거의 동등한 기능을 인터페이스가 제공합니다. 이 인터페이스로 성능 분석 툴 사용권 프로그램 보고서의 서브세트를 생성 및 볼 수 있습니다.
- PRTCPTPT(구성요소 보고서 인쇄) 명령이 개선되었고 그 내용은 다음과 같습니다.
 - 인쇄할 작업 유형을 선택할 수 있는 새로운 선택 범주가 있습니다. 이는 서버 모델에서 대화식 작업을 분석할 때 도움이 됩니다.
 - 보고서에 포함시킬 작업 등록 정보를 선택할 수 있는 새로운 선택 범주가 있습니다. 이는 시스템에 오버헤드가 걸릴 때 고우선순위 작업의 합계를 확인하고 모든 우선순위가 00인 작업을 찾는 데 도움이 됩니다.
 - 쓰기 캐시 과수행 백분율을 보고하는 새 열이 있습니다(% 쓰기 캐시 오버런).
 - 데이터베이스 저널링 요약 섹션에는 저널 카운터 및 조작과 관련된 정보를 보여주는 새로운 서브섹션이 있습니다. 새로운 카운터에 대한 정보는 QAPMJOBMI 파일에 저장됩니다.
- PRTSYSRPT(시스템 보고서 인쇄) 명령이 갱신되었고 개선된 내용은 다음과 같습니다.
 - 통신 요약에 지점 간 프로토콜(PPP) 정보가 포함됩니다. PPP 프로토콜이 자원 보고서의 통신 세부사항 섹션에도 표시됩니다. DSPPFRTA(성능 자료 표시) 명령은 통신 회선 세부사항 표시 화면의 지점 간 프로토콜 정보도 표시합니다.
 - HTTP Server 작업에서 처리하는 트랜잭션에 대한 정보를 보여주는 새로운 섹션이 추가되었습니다. 이 정보는 보고서의 맨 아래에 요약과 평균 행을 표시하여 각 간격에 대해 표시됩니다.
 - 비대화식 서버 작업에 대한 추가 정보를 표시하기 위해 작업부하, 자원 활용 및 자원 활용 확대 섹션에 새로운 범주가 포함되었습니다.
- 시스템 활동에 대한 작업 화면에는 현재의 처리 용량이 표시됩니다. 이 정보는 공유 프로세스 환경에서 유용한데, 특히 파티션의 용량은 변경되었지만 실질적 프로세서는 변경되지 않은 경우 유용합니다. 시스템 활동에 대한 작업 화면에는 작업이나 타스크가 대기 상태에서 소비한 시간의 전체 %도 표시됩니다. 옵션 6(대기 세부사항)을 사용하여 대기 범주의 리스트를 볼 수 있습니다. 애기 시간 계수 자료와 관련된 여러 필드가 QAITMON 파일(WRKSYSACT 명령으로 작성됨)에 추가되었습니다.
- 트랜잭션 보고서의 오브젝트별 점유/잠금 충돌 요약 섹션에는 최대 500개의 오브젝트를 표시할 수 있습니다. 추적 자료가 500개 이상의 점유/잠금 충돌을 가질 수 있습니다. 따라서 500개의 가장 중요한 오브젝트가 이 섹션에 표시됩니다.

- STRPFRTRC(성능 추적 시작) 명령이 더 큰 추적 자료표를 지원합니다.
- 보고서의 일부 필드의 크기가 더 큰 값을 표시하도록 증가되었습니다. 변경된 사항은 다음 보고서에 영향을 미칩니다.

보고서	섹션	필드
시스템	자원 활용	초 당 디스크 I/O
시스템	자원 활용 확대	실제 디스크 I/O, 논리 디스크 기본 I/O, 디스크 I/O(동기 및 비동기), DIO/Sec(동기 및 비동기)
구성요소	구성요소 간격 활동	초 당 디스크 I/O
작업 간격	비대화식 작업 요약	초 당 I/O 수
작업 간격	비대화식 작업 세부사항	초 당 Nbr I/O

새로운 사항: Performance Management/400

다음은 이번 릴리스에서의 새로운 기능입니다.

PM/400은 이전 릴리스에서 업그레이드하지 않고 이전 릴리스의 PM/400 성능 자료 콜렉션 기능을 끈 경우에도 자동으로 성능 자료를 합니다. IB에 전송하도록 권한을 부여할 때까지 자료는 IBM에 전송되지 않습니다. 성능 자료 자동 콜렉션의 장점은 필요로 할 때 자료가 수집되어 있다는 점과 PM/400을 활성화한 직후 PM/400 보고서를 받아볼 수 있다는 점입니다.

PM/400e 보고서에 대한 최신 정보는 PM/400 웹 사이트  를 참조하십시오.

성능 자료 변환

성능 데이터베이스 파일은 새 릴리스 설치 후 레벨이 낮아지므로 성능 자료 콜렉션을 계속하기 위한 조치를 수행해야 합니다. Performance Management/400, Collection Services 또는 성능 분석 툴 사용권 프로그램을 사용하며 새 릴리스를 설치하기 전에 자료를 수집한 경우 이전 릴리스의 성능 자료를 처리하는 것과 관련하여 다음과 같은 선택을 할 수 있습니다.

- 자료 변환
CVTPFRDTA(성능 자료 변환) 명령을 사용하여 콜렉션 라이브러리의 자료를 변환합니다.
- 자동 자료 콜렉션 지원 사용
성능 데이터베이스 파일을 변환하지 않은 경우와, 콜렉션 서비스가 시작되어 사용자가 데이터베이스 파일을 자동으로 작성하는 옵션을 지정한 경우, 이전 릴리스 파일에 대한 성능 라이브러리가 작성됩니다. 성능 데이터베이스 파일은 이 라이브러리로 이동됩니다. 이 조치로 새 파일이 작성되고 기존 자료 파일을 이전 릴리스 레벨로부터 보호할 수 있습니다. 다음 사항에 대해 알아야 합니다.
 - 작성되는 라이브러리명은 QPFRDvrmnn입니다. 여기서 vrm은 현재 버전, 릴리스 및 수정이고 nn은 01로 시작하는 고유한 순번입니다(예: QPFRD52001).
 - 라이브러리는 *EXCLUDE 공용 권한을 사용하여 작성됩니다. 라이브러리는 QSYS 사용자 프로파일이 소유하며 원래 라이브러리의 소유자에게는 *ALL 권한이 부여됩니다.
 - 모든 QAPMxxxx 파일이 이동됩니다.

- 이전 릴리스의 자료를 보관하지 않으려면 DLTLIB(라이브러리 삭제) 명령(DLTLIB qprfdvrnmn)을 사용하여 QPRFDvrnmn 라이브러리를 삭제할 수 있습니다.
- QAPMxxxx 파일 삭제
성능 데이터베이스 파일을 변환하지 않고 컬렉션 서비스를 실행하지 않은 경우 DLTF(파일 삭제) 명령(DLTF 라이브러리/QAPM*)을 사용하여 QAPMxxxx 파일을 삭제할 수 있습니다.

새로운 사항: Performance Explorer

다음은 이번 릴리스에서의 새로운 기능입니다.

- 이제 Performance Explorer 명령을 사용하려면 사용자는 *SERVICE 권한이 있어야 합니다.
- 새로운 CL 명령을 이용하면 캡처하는 자료의 양을 줄이기 위한 필터를 작성할 수 있습니다.
 - ADDPEXFTR(PEX 필터 추가)
 - WRKPEXFTR(PEX 필터에 대한 작업)
 - RMVPEXFTR(PEX 필터 제거)
 - STRPEX 명령은 필터를 지원하도록 개선되었습니다.
- Performance Explorer는 이제 관리 컬렉션 오브젝트(*MGTCOL)를 사용하여 수집된 자료를 이의 데이터베이스 파일로 변환합니다. CRTPEXDTA(PEX 자료 작성) 명령이 변환을 수행합니다.
- 다음 이벤트가 ADDPEXDFN(PEX 정의 추가) 명령에 추가되었습니다.
 - 어플리케이션 이벤트(APPEVT)
 - 휴대용 어플리케이션 솔루션 환경(PASE) 이벤트(PASEEVT)
 - 저널 이벤트(JRNEVT)
 - iSeries NetServer, 파일 서버 및 네트워크 파일 시스템 서버와 클라이언트 이벤트(FILSVREVT)
 - 동기화 이벤트(SYNCEVT)
 - 전문가 캐시 이벤트(EXPCCHEVT)
- 새 값이 ADDPEXDFN 명령에서 오퍼레이팅 시스템(OSEVT) 매개변수 *HOSTSVRCNN에 추가되었습니다. 이 값은 *DBSVRCNN 값과 동일합니다. *HOSTSVRCNN가 권장되는 값입니다.
- Performance Explorer 정의에 대해 작업할 수 있도록 새로운 명령(WRKPEXDFN(PEX 정의에 대한 작업) 명령)이 제공됩니다.
- Performance Explorer 세션 종료 시 새로운 컬렉션 상태인 불완전을 표시할 수 있습니다. 이 상태는 컬렉션이 예기치 않게 종료했음을 의미합니다.
- 이 릴리스의 새 데이터베이스 파일에는 다음이 있습니다.

새 파일	설명
QAYPEPROCI	이 파일에는 컬렉션 자료의 모든 명령어 주소와 연관된 프로시저 정보를 제공합니다. 이전에 QAYPEMII, QAYPELICI, QAYPENMI 및 QAYPENLIC 파일에 있었던 자료가 여기에 들어 갑니다.

새 파일	설명
QAYPECFGI	이 파일에는 콜렉션 정의 정보가 들어 있습니다. 이전에 QAYPECOFG, QAYPECICFG, QAYPESTCFG, QAYPETRCFG 및 QAYPEHWCFCG 파일에 있었던 자료가 여기에 들어 갑니다.
QAYPEFTRI	이 파일에는 필터 정의 정보가 들어 있습니다.
QAYPELTASK	이 파일에는 정의 정보의 TASKNAME와 수가 들어 있습니다.
QAYPERINF	이 파일에는 기타 분석 정보(예: 각 IP 주소의 문자 스트링 버전)가 들어 있습니다.
QAYPETSWSW	이 파일에는 TASK 전환 이벤트에 특정한 정보가 들어 있습니다. 이전에 QAYPEBASE 파일에 있었던 자료가 여기에 들어 갑니다.

- 이 릴리스에서는 다음과 같은 데이터베이스 파일이 삭제되었습니다.

삭제된 파일	자료 이동 위치
QAYPECICFG	QAYPECFGI
QAYPECOCFG	QAYPECFGI
QAYPEHWCFCG	QAYPEPROCI
QAYPELICI	QAYPECFGI
QAYPELNAMT	QAYPELTASK
QAYPELNUMT	QAYPELTASK
QAYPEMII	QAYPEPROCI
QAYPENLIC	QAYPEPROCI
QAYPENMIC	QAYPEPROCI
QAYPEPERD	정기 모드가 더 이상 지원되지 않으므로 표가 필요하지 않습니다.
QAYPEPSUM	이제 QAYPEPPANE 및 QAYPETASKI 자료로 프로파일 요약 자료를 연산할 수 있습니다.
QAYPEPWDW	이 파일에는 사용되지 않았던 프로파일 요약 자료가 들어 있습니다.
QAYPESTCFG	QAYPECFGI
QAYPES36	System/36 이벤트는 더 이상 지원되지 않습니다.
QAYPETRCFG	QAYPECFGII
QAYPETRCPT	QAYPETIDX
QAYPEUNKWN	QAYPEUSRDF

- 이 릴리스에서는 다음과 같이 데이터베이스 파일이 변경되었습니다.
 - 문자 자료 필드가 디폴트 CSID 65535를 사용합니다.
 - 8바이트 주소, 명령어 및 세그먼트가, 각 바이트를 2개의 16진 문자로 변환하여 16바이트 문자 필드나 16바이트 16진 필드에 저장되는 대신 8바이트 16진 필드에 저장됩니다.
 - 팩 10진 필드에 저장되는 참 숫자 자료가 정수 필드로 변경되었습니다.
 - 5자리 팩 10진 필드가 98 정수 필드로 변경되었습니다.
 - 10자리 팩 10진 필드가 18B 정수 필드로 변경되었습니다.

- 20자리 팩 10진 필드가 18B 정수 필드로 변경되었습니다. 단 값이 8바이트 부호화 정수보다 큰 경우에 한합니다.

이 주제 인쇄





성능 주제의 PDF 버전을 보거나 다운로드하려면 성능(약 350KB 또는 95 페이지)을 선택하십시오. 이 PDF에는 성능 데이터베이스 표 정보는 포함되지 않습니다.






성능 데이터베이스 표 정보의 PDF 버전을 보거나 다운로드하려면 성능 데이터베이스 표(약 350KB 또는 125 페이지)를 선택하십시오.

다음과 같은 관련 주제를 보거나 다운로드할 수도 있습니다.

- 중앙 관리(약 250KB 또는 55 페이지)에는 중앙 관리 종료점 시스템과 시스템 그룹을 설정하는 방법에 대한 정보와 중앙 관리가 다음과 같은 서버 관리 작업을 효과적으로 처리하는 모든 방법에 대한 정보도 들어 있습니다.
 - 사용자 및 그룹 관리
 - 자료의 패키지 및 송신
 - 명령 실행
 - 중앙 관리 스케줄러 또는 고급 작업 스케줄러를 사용한 task 또는 작업 스케줄링
- 작업 관리(약 173KB 또는 40 페이지)에서는 다음과 같은 작업 관리 개념을 설명합니다.
 - 작업 수명
 - 일일 작업 관리
 - 시스템 구조
 - 작업 수행 방식


또한 다음 PDF도 무엇이든 열람하거나 인쇄할 수 있습니다.

- 매뉴얼:
 - Performance Tools for iSeries  (약 400 페이지)
 - iSeries Performance Capabilities Reference 
이 책의 최신 버전이 들어 있는 웹 사이트에 대한 링크. 이 참조서는 성능 벤치마킹, 용량 계획 및 서버 성능 계획에 유용한 기술적인 정보를 제공합니다.
 - 이 참조서는 성능 벤치마킹, 용량 계획 및 서버 성능 계획에 유용한 기술적인 정보를 제공합니다.
- 레드북:
 - AS/400 Performance Management  (약 504 페이지)
 - AS/400 HTTP Server Performance and Capacity Planning  (약 205 페이지)

- Java and WebSphere Performance on IBM eserver iSeries Servers  (약 235 페이지)
- Managing OS/400 V5R1 with Operations Navigator  (약 550 페이지)
- DB2 UDB/WebSphere Performance Tuning Guide  (약 360 페이지)
- Lotus Domino for AS/400: Performance, Tuning, and Capacity Planning  (약 550 페이지)
- AS/400 Performance Explorer Tips and Techniques  (약 550 페이지)

보거나 인쇄하기 위해 워크스테이션에 PDF를 저장하려면 다음을 수행합니다.

1. 브라우저에서 PDF를 엽니다(위의 링크를 클릭).
2. 브라우저의 메뉴에서 파일을 클릭합니다.
3. 다른 이름으로 저장...을 클릭합니다.
4. PDF를 저장하려는 디렉토리를 탐색합니다.
5. 저장을 클릭하십시오.

이 PDF를 보거나 인쇄하기 위해 Adobe Acrobat Reader가 필요한 경우 Adobe 웹 사이트 (www.adobe.com/prodindex/acrobat/readstep.html)  에서 사본을 다운로드할 수 있습니다.

성능 계획

시스템의 성능을 계획하려면 성능 목표를 설정하고 이 목표에 기초한 벤치마크를 작성하며 시스템의 확장을 계획해야 합니다. 이 섹션에서는 시스템의 성능을 계획하는 데 있어 필요한 단계를 안내합니다.

시스템의 성능을 계획할 때 시스템이 가지고 있는 비즈니스 요구사항을 충분히 이해하고 이러한 비즈니스 요구를 성능 목표로 변환해야 합니다. 비즈니스 요구가 관련되는 한 성능 목표도 관련되어야 함을 기억하십시오.

가장 좋은 시작 방법은 최대 사용 비즈니스 기간 동안 컴퓨터 시스템에 필요한 시간당 및 일일 대화식 트랜잭션의 최대 처리량을 예측하는 것입니다. 그런 후 로컬 및 리모트 워크스테이션에서 허용 가능한 평균 응답 시간을 결정할 수 있습니다. 정규의 일괄처리 프로세스에 걸리는 시간, 비즈니스 요구사항을 달성하기 위해 이들 프로세스가 정시에 완료되도록 스케줄하는 방법에 대해 생각해야 합니다.

그런 후 기본 통계 세트를 설정할 수 있는데 이는 다음의 내용을 담고 있는 성능 목표 계획에 문서화해야 합니다.

- 시간 당 최대 트랜잭션
- 일일 최대 트랜잭션
- 로컬 워크스테이션에 수용 가능한 평균 응답 시간
- 최대 대화식 트랜잭션
- 주요 스케줄 일괄처리 작업, 실행 시기 및 예상 실행 시간 리스트

- 필요할 수도 있는 스케줄되지 않은 다른 일괄처리 작업의 리스트

성능을 계획하려면 다음 타스크를 완료하십시오.

시스템 벤치마크 설정

시스템 벤치마크를 잘 설정하면 적절히 조정된 시스템의 성능 자료를 제공받을 수 있습니다. 시스템 변경 이전과 이후의 성능 벤치마크는 문제 해결과 계획을 위한 중요한 정보를 제공합니다.

시스템 확장 시기와 방법 결정

비즈니스 요구가 변화함에 따라 시스템도 바뀌어야 합니다. 변경에 대처하기 위해서는 현재 시스템을 모델링한 후 시스템, 구성 또는 작업부하가 바뀌면 어떤 일이 발생할지를 판별해야 합니다.

Capacity Upgrade on Demand

Capacity Upgrade on Demand는 비즈니스 필요성 변경으로 더 많은 자원이 요구될 때 iSeries 서버에 프로세서를 추가할 수 있게 합니다. 시험, 임시 또는 영구 기준으로 프로세서를 주문하고 활성화할 수 있습니다.

성능 관리 전략 선택

서로 다른 비즈니스 요구는 서로 다른 성능 관리 전략을 필요로 합니다. 다음은 세 가지 기본 비즈니스 모델과 이에 대해 제안된 성능 관리 전략입니다.

시스템 벤치마크 설정

새로운 대화식 어플리케이션을 추가하거나 시스템 업그레이드를 수행하는 등 시스템 구성의 주요 내용을 변경하기 전에 시스템 벤치마크를 설정해야 합니다. 정확한 벤치마크 정보를 유지보수함으로써 필수적인 문제 해결 정보를 제공할 수 있습니다. 최소한 벤치마크에는 콜렉션 서비스의 현재 콜렉션 오브젝트가 포함되어야 합니다. 사용자 환경에 따라 Performance Explorer를 사용하여 좀 더 자세한 정보를 유지보수해야 할 수도 있습니다.

벤치마크를 설정하자면 다음이 필요합니다.

- 정확한 iSeries 구성을 사용할 수 있습니다.
- 어플리케이션과 자료가 대표적이고 유효합니다.
- 사용되는 모든 프로그램과 소프트웨어의 정확한 버전을 가지고 있습니다.
- 테스트 실행에 필요한 사용자와 워크스테이션을 사용할 수 있습니다.
- 각 사용자에 대해 트랜잭션이 잘 정의되어 있습니다.

대화식 작업부하에 대해 의미있는 벤치마크를 실행하는 것은 워크스테이션에서 사용자를 시뮬레이트할 수 있도록 하는 특별한 장비 없이는 거의 불가능합니다. 일괄처리 벤치마크를 실행하는 것은 대화식 어플리케이션의 성능을 테스트하는 것만큼 복잡하지는 않지만 위의 처음 세 가지 사항은 이러한 유형의 테스트에도 유효합니다. 그러나 동시 일괄처리 및 대화식 작업(이것이 실제 고객 환경인 경우가 많음)에서 시스템 벤치마크를 설정하는 경우에도 적절한 수의 사용자와 워크스테이션이 필요합니다.

시스템 확장 시기와 방법 결정

사업 요구가 점차 발전해가면서 사용자 시스템에 대한 요구도 더불어 발전해갑니다. 향후의 시스템 요구와 성장을 계획하려면 시스템, 구성 또는 작업부하가 바뀌면 어떤 일이 발생할지를 판별해야 합니다. 이 프로세스를 경향 분석이라고 하며 경향 분석은 매일 수행해야 합니다. 시스템이 자원 용량 가이드라인에 접근해가면 이 자료를 좀 더 자주 수집해야 할 수 있습니다.

경향 분석은 대화식 및 일괄처리 환경에 대해 별도로 수행해야 합니다. 회사에서 특정의 어플리케이션을 광범위하게 사용하는 경우 이 어플리케이션에 대해 경향 분석을 수행해야 할 수 있습니다. 추적에 중요한 또다른 환경은 월말 처리입니다. 경향 분석 자료를 지속적으로 수집하는 것이 중요합니다. 시스템의 최대 작업부하 시간이 10:00 AM에서 2:00 PM 사이이고 이 시간 기간에 대해 경향 분석 자료를 수집하는 경우 이 자료를 다른 시간 기간에서 수집된 자료와 비교하지 마십시오.

용량 계획 및 성능 분석을 올바르게 수행하려면 성능 자료를 수집, 분석, 유지보수 및 아카이브해야 합니다. IBM은 용량 계획, 자원 예측 및 크기 조정에 도움이 되는 몇 가지 툴을 제공합니다.

Performance Management/400(PM/400)

PM/400은 자료 수집, 자료 분석 및 자료 아카이브를 완전히 자동화하여 사용자에게 이해하기 쉬운 요약된 성능 및 용량 정보를 제공합니다. PM/400은 중요한 성능 인디케이터를 지속적으로 분석함으로써 시스템 자원을 계획하고 관리하는 데 도움이 됩니다. 이 기능은 OS/400 사용권 프로그램과 함께 제공됩니다. 사용자가 할 일은 이 기능을 활성화시키고 자료가 수집되어 IBM으로 전송되는지를 주기적으로 확인하는 것 뿐입니다. 모든 콜렉션 사이트는 네트워크 보안이 되어 있으며 PM/400e 서비스는 비독점적인 성능 자료만을 IBM으로 전송합니다. 전송 시간은 전적으로 사용자가 제어할 수 있습니다.

작업부하 예측기

작업부하 예측기는 특정 작업부하 유형에 대해 측정된 작업부하에 근거하여 시스템 요구를 조정할 수 있도록 돕는 툴입니다. 웹 기반 어플리케이션을 통해 PM/400이 보고한 기존 시스템의 활용, 성능 및 성장을 수용할 수 있는 iSeries 시스템에 맞게 업그레이드를 조정할 수 있습니다. 추가 옵션으로, 크기 조정에는 특정 어플리케이션(예: Domino, Java, Websphere)을 추가하기 위한 용량과 여러 AS/400 또는 iSeries OS/400 워크로드를 한 시스템에 통합하는 것도 포함할 수 있습니다. 이 기능을 이용하여 사용자 시스템에서 발생하는 기존 활용 자료에 근거한 향후의 시스템 요구사항을 계획할 수 있습니다. 이 어플리케이션은 Processor On-Demand 환경이나 논리 파티션이 있는 환경은 지원하지 않습니다.

PATROL for iSeries(AS/400) - Predict

이 제품은 고가용성 및 최적의 성능을 위해 필요한 다수의 루틴 관리 타스크를 자동화하여 iSeries 성능 관리를 도와줍니다. 또한 이 제품은 자세한 용량 계획 정보를 제공하여 사용자의 iSeries 환경에 대한 확장 계획을 도와줍니다.

성능 전략을 작성 및 구현하는 데 대한 자세한 정보는 성능 관리 전략 선택을 참조하십시오.

Capacity Upgrade on Demand

iSeries용 Capacity Upgrade on Demand(CUoD)는 서버를 다시 시작하거나 비즈니스에 영향을 주지 않고 선택 서버 모델의 하나 이상의 중앙 프로세서를 동적으로 활성화하기 위한 용량을 제공하는 피처입니다. 시험판 또는 영구 기초로 서버에 이미 설치된 대기 프로세서를 하나 이상 주문하여 활성화할 수 있습니다.

CUoD를 사용하면, 추가 프로세서를 활성화하고 필요성이 증가함에 따라 새 처리 전력에 대해서만 비용을 지불할 수 있습니다. 현재 조작에 영향을 주지 않고 프로세서 용량을 증가할 수 있습니다.

CUoD의 작동 방법과 CUoD를 이용하기 위해 필요한 사항에 대해 알려면 이 주제를 사용하십시오.

Capacity Upgrade on Demand 개념

CUoD의 개념적 이해를 원하면 여기에서 시작하십시오.

Capacity Upgrade on Demand 준비

중요한 계획 및 준비 요구사항을 고려하도록 하십시오.

Capacity Upgrade on Demand 주문

활성화 피처를 주문하여 대기 프로세서를 활성화할 활성화 코드를 받아야 합니다.

대기 프로세서 활성화

iSeries 서버에서 하나 이상의 대기 프로세서를 활성화하려면 이 정보를 사용하십시오.

시나리오: Capacity Upgrade on Demand

관리자가 추가 용량을 계획, 주문 및 활성화하기 위해 밟게 되는 단계들의 예를 보려면 이 정보를 사용하십시오.

Capacity Upgrade on Demand 개념

Capacity Upgrade on Demand(CUoD)는 무료 14일 시험판 기초에서 iSeries 모델 선택시 추가 프로세서를 활성화하거나 영구 프로세서 활성화를 구입하면 용량을 제공합니다. 이 용량은 새 작업부하에 따른 용량을 추가하여 서버가 예기치 않은 성능 요구에 적응하도록 빠르고 경제적인 방식을 가능하게 함으로써 iSeries 서버에 중요한 가치를 부가합니다.

선택 iSeries 모델은 여러 시작 프로세서와 함께 제공됩니다. 시작 프로세서는 이미 제공될 때 iSeries 서버에서 활성화된 프로세서입니다. 대기 프로세서는 서버와 함께 제공되지만 사용자가 활성화할 때까지 사용할 수 없는 프로세서입니다. 대기 프로세서는 14일 시험판 기초에서 임시로 활성화하거나, 활성화 피처를 구입하고 제공된 활성화 코드를 입력하여 영구적으로 활성화할 수 있습니다. 다음 표는 각 모델에서 사용할 수 있는 시작 및 대기 프로세서 수를 나열한 것입니다.

모델	프로세서 피처	시작 프로세서 수	대기 프로세서 수	설치된 프로세서 수
825	2473	3	3	6
830	2349	4	4	8
830	2351*	1	7	8
840	2352	8	4	12
840	2353	12	6	18

모델	프로세서 피처	시작 프로세서 수	대기 프로세서 수	설치된 프로세서 수
840	2354	18	6	24
840	2416	8	4	12
840	2417	12	6	18
840	2419	18	6	24
870	2486	8	8	16
890	2487	16	8	24
890	2488	24	8	32
890	2497	16	8	24
890	2498	24	8	32

* 제한된 사용 가능성

다음 개념은 CuoD를 이용해야 하는 정보를 제공합니다.

Capacity Upgrade on Demand 시험 기간

추가 프로세서 용량의 이점을 측정하기 위해 무료 14일 시험 기간을 이용합니다.

요구시 용량 업그레이드의 소프트웨어 사용권 부여 고려사항

CUoD 활성화가 소프트웨어 층에 미치는 영향에 대해 알아보십시오.

Processors on Demand 활성화 코드

대기 프로세서의 일부 또는 전체를 영구적으로 활성화할 것을 결정하고 나면, 하나 이상의 Processors On Demand 활성화 피처를 주문하여 구입해야 합니다. 그러면 대기 프로세서를 활성화할 수 있는 활성화 코드가 제공됩니다.

Electronic Service Agent 요구사항

CuoD 대기 프로세서를 활성화하는 데 필요한 필수 제품 자료의 빠르고 정확한 전송을 위해 Electronic Service Agent를 사용합니다.

Capacity Upgrade on Demand 시험 기간: Capacity Upgrade on Demand의 14일 시험 기간은 비용 없이 대기 프로세서의 사용을 평가하려고 하는 고객이 사용할 수 있습니다. 활성화하면, 시험 기간은 서버의 초기 시작 후 1회, 그리고 활성화 코드를 입력할 때마다 1회 사용할 수 있습니다.

시작하면, 시험 기간은 전원이 공급된 날을 기준으로 14일 동안 사용할 수 있습니다. 시험 기간 시계는 서버에 전원이 공급될 경우에만 가동됩니다. 시작 및 대기 프로세서 모두 시험 기간 동안 활성화됩니다. 시험 기간을 중단하여 재시작할 수 없습니다.

이 피처를 시도하려면 시험 기간 동안 대기 프로세서 활성화를 참조하십시오.

Capacity Upgrade on Demand의 소프트웨어 사용권 부여 고려사항: 많은 비즈니스 파트너들이 서버에서 사용할 수 있는 프로세서 피처 코드 시스템 값을 사용하여 소프트웨어의 소프트웨어 사용권 부여 비용을 설정합니다. 이 시스템 값은 대기 프로세서가 있는 서버에서 계속 사용할 수 있으며, 활성화한 대기 프로세서 수에 관계없이 프로세서 피처 코드 시스템 값은 동일하게 유지됨을 명시해야 합니다.

서버의 프로세서 수에 따라 소프트웨어 사용권 부여 비용을 산출하는 소프트웨어 제공자는 이전부터 "소프트" 준수 접근방식을 사용해 왔습니다. 각 프로세서가 활성화될 때, 프로세서는 프로세서 수에 의해 소프트웨어 사용권을 부여한 서버에 상주하는 소프트웨어와 연관되는 필수 소프트웨어 사용권 부여 비용을 알리고 지불할 고객 책임을 유지합니다.


프로세서 활성화는 소프트웨어 층을 변경하지 않습니다.

주: Processors on Demand 활성화 피처를 IBM 구성 프로그램 중 하나를 통해 주문한 경우, 프로세서에 의해 사용권이 부여된 고객 서버에 현재 설치된 소프트웨어 제품에 대해 IBM 소프트웨어 사용권 부여의 추가 비용이 부과됩니다.

Processors On Demand 활성화 코드: 대기 프로세서의 일부 또는 전체를 영구적으로 활성화할 것을 결정하고 나면, 하나 이상의 Processors On Demand(POD) 활성화 피처를 주문하여 구입해야 합니다.

주문을 제출하면, 주문 레코드는 서버에서 수집된 필수 제품 자료(VPD)와 결합합니다. 이 정보는 서버에 특정한 POD 활성화 코드를 생성하기 위해 사용됩니다. 이는 대기 프로세서를 활성화하기 위해 사용할 용량 사용권 키로 생각할 수 있습니다.


이 활성화 코드는 빠른 액세스를 위해 IBM 웹 사이트에 게시됩니다. 이는 보통 주문이 서버에서 수집된 필수 APD와 결합되었다고 가정할 때 하루(24시간) 내에 이루어집니다. 활성화 코드가 생성되면, iSeries Capacity Upgrade on Demand 웹 사이트에서 시스템 유형 및 일련 번호를 사용하여 이 코드에 액세스할 수 있습니다.

<http://www.ibm.com/servers/eserver/series/hardware/ondemand> 

POD 활성화 피처를 주문하고 POD 활성화 코드를 받는 방법에 대한 지시사항은 Capacity Upgrade on Demand 주문을 참조하십시오.

Electronic Service Agent 요구사항: Capacity Upgrade on Demand를 구입하기 위한 주문을 제출할 때, IBM은 주문 정보를 서버의 필수 제품 자료(VPD)와 결합하여 서버에서 대기 프로세서를 잠금 해제하고 활성화하는 데 필요한 활성화 코드를 작성합니다.

VPD 정보는 Extreme Support의 일부인 Electronic Service Agent, IBM의 포괄적인 기술 서비스 및 IBM iSeries에 독점된 지원 주도권을 사용하여 전자 방식으로 IBM에 보낼 수 있습니다. Electronic Service Agent는 서버에 상주하는 무료 사용권 프로그램(5798-RZG)으로, 이벤트를 모니터링하고 정기적, 고객 정의 시간표에 따라 IBM에 서버 명세 정보를 전송하도록 설계되어 있습니다.

설치 관련 정보를 포함하여 Electronic Service Agent에 대한 완전한 문서는 Electronic Service Agent for iSeries User's Guide  를 참조하십시오.

Capacity Upgrade on Demand 준비

Capacity upgrade on Demand는 서버 작동에 영향을 주지 않고 프로세서 용량을 추가할 수 있게 합니다. 그러나 새 용량을 끊어짐 없이 통합하려면 주문을 제출하기 전에 서버를 준비해야 합니다.

Capacity Upgrade on Demand에서의 용량 계획

이 주제에는 CUoD 가능 서버 모델에 대해 용량 계획을 수행하기 위한 고려사항 및 자원이 포함됩니다.

프로세서 활성화 시기 결정

이 주제에서는 자원 활용에서 경향을 모니터링할 수 있는 툴, 추가 용량이 필요할 경우의 권장사항, 그리고 대기 프로세서가 있는 서버에서 툴이 CPU 활용을 보고하는 방법에 대해 설명합니다.

Capacity Upgrade on Demand 환경 설정

새 프로세서 용량을 통합하기 위해 서버를 준비하고, 필요할 경우 추가 용량을 주문할 준비를 하십시오.

Capacity Upgrade on Demand에서의 용량 계획: 대기 프로세서가 있는 서버의 용량 계획에는 원래 다른 iSeries 모델의 크기 조절에 사용하는 것과 같은 절차 및 자원을 사용합니다. 필요한 서버 용량을 판별하는 데 도움을 줄 수 있는 포괄적인 툴, 자원 및 오퍼링 세트는 대기 프로세서가 있는 서버를 지원하도록 갱신되었습니다.

용량 계획에 관한 도움이 필요하다면, 다음 자원을 참조하십시오.

서버 확장 시기와 방법 결정

이 주제 개요는 용량을 계획하고 자원 활용에서 경향을 식별할 때 도움을 줄 수 있는 몇 가지의 툴에 대해 설명합니다.

iSeries Benchmark Center

벤치마킹 어플리케이션 환경에 대한 도움이 필요하다면 이 IBM 웹 사이트를 사용하십시오.

iSeries Solutions Center — Capacity Planning Services

이 IBM 문의 서비스는 점점 증가하는 비즈니스 요구를 충족시키는 서버 솔루션 계획에 도움을 줄 수 있습니다.

주: 작업부하 예측기는 CUoD를 지원하지는 않지만, 권장하는 활동 프로세서 수에 해당하는 예측만 리턴합니다. 대기 프로세서에서 제공하는 추가 용량은 권장사항에 제시되지 않습니다.

프로세서 활성화 시기 결정: Capacity Upgrade on Demand는 작업부하로 추가 자원이 필요할 때 시스템에 프로세서를 추가할 수 있는 기능을 제공합니다. 시스템 CPU 활용도와 CPU 활용에서의 추세를 모니터링하여 추가 프로세서를 활성화하는 시기와 필요한 프로세서 수를 결정해야 합니다.

CPU 활용 정보를 보고하기 위한 많은 성능 분석 툴이 있습니다. 특히, PM/400은 자원 활용에서 추세를 식별하고, iSeries Navigator 모니터는 사용되는 자원에 대한 자세한 정보를 제공하며, 활용이 사전에 정의된 심각 레벨에 도달할 경우 경고로 알립니다.

대기 프로세서가 있는 서버의 CPU 활용 측정

사용 가능한 모든 프로세서의 평균 사용 정도를 계산할 때, CPU 활용을 보고하는 시스템 함수는 총 CPU 용량에 대기 프로세서를 포함하지 않습니다. 대기 프로세서는 CPU 활용 백분율을 보고하는 다양한 시스템 함수 내에서 활동 중인 것으로 간주하지 않습니다. 사용된 CPU 용량 백분율(iSeries Navigator에서 CPU 활용 CPU

백분율)은 경과된 시간 내에 프로세서가 활동한 시간 양을 기초로 계산됩니다. 이는 보통 백분율로 보고되며, 100%는 프로세서가 전체 경과 시간 동안 사용되었음을 표시합니다. 여러 개의 프로세서가 존재할 경우, CPU 시간은 활용이 항상 사용 가능한 총 용량의 평균 백분율로 보고되도록 모든 프로세서의 평균 사용을 표시하도록 조정해야 합니다.

대화식 기능은 구입한 대화식 피처에 의해 판별됩니다. 이 기능은 대기 프로세서 수에 관계없으며 대기 프로세서가 활동될 때 변하지 않습니다. 대화식 기능의 백분율로 보고되는 대화식 활용은 Capacity Upgrade on Demand 기술의 영향을 받지 않습니다. iSeries Navigator 내에서, 이러한 미터법을 CPU 활용(대화식 피처)이라고 합니다.

대화식 CPU 활용은 총 시스템 CPU 백분율로도 보고됩니다. 중앙 관리 내에서는 이 미터법을 CPU 활용(대화식 작업)이라고 합니다. 이 미터법은 시스템 CPU 활용에 대해 위에서 설명한 것과 같은 방식으로 CUoD가 있는 서버에 영향을 미칩니다.

Capacity Upgrade on Demand 환경 설정: POD(Processor On Demand) 활성화 코드를 주문하기 전에, 추가 용량을 주문하고 통합하기 위해 환경을 준비해야 합니다.

주문 준비

POD 활성화 코드를 구입할 때, 서버의 필수 제품 자료(VPD)도 제공해야 합니다. 팩스에 의해 수동으로, 또는 Electronic Service Agent를 사용하여 전자 방식으로 이를 수행할 수 있습니다. Electronic Service Agent를 사용하여 VPD 자료를 보내면 수동 방법과 연관되는 지연이 발생하지 않으므로, POD 활성화 코드는 주문 수신 24시간 내에 게시됩니다. 성능에 영향이 가지 않도록 하려면 시험판 기초에서 추가 프로세서를 사용할 때 주문을 보내고 처리해야 할 충분한 시간이 있어야 합니다.

전자 방법을 사용하여 VPD 자료를 보내려면, Electronic Service Agent 설정을 수행하십시오.

추가 용량 준비


서버가 활성화된 프로세서를 완전하게 이용할 수 있도록 하기 위해, 다음 사항들을 준비할 수 있습니다.

- I/O 조건지정 수행
- 메모리 업그레이드 수행
- 논리 파티션(LPAR) 준비

LPAR을 사용하는 서버의 경우, 대기 프로세서는 항상 1차 파티션과 연관됩니다. 모든 프로세서를 파티션에 지정해야 합니다.

Electronic Service Agent 설정: Electronic Service Agent를 사용하여 VPD 데이터를 보낼 수 있습니다. Electronic Service Agent를 설정하려면 다음 단계를 수행하십시오.

1. Electronic Service Agent를 설치하십시오.

설정 및 설치 지시사항에 대해서는 Electronic Service Agent for iSeries User's Guide  를 참조하십시오.

2. TCP/IP를 설정하고 시작했는지 확인하십시오.

Electronic Service Agent에 대한 자세한 정보는, Electronic Service Agent 요구사항을 참조하십시오.

Capacity Upgrade on Demand 주문

새 서버, 모델 업그레이드 또는 설치된 서버에 대해 POD(Processors on Demand) 활성화 피처를 주문할 수 있습니다. 새 서버 또는 모델 업그레이드의 경우, 주문에 하나 이상의 POD 활성화 피처를 포함할 수 있습니다. 이러한 경우, POD 활성화 코드는 서버가 사용자에게 제공되기 전에 입력합니다.

설치된 서버의 경우, 대기 프로세서의 일부 또는 전체를 영구적으로 활성화할 것을 결정하고 나면 하나 이상의 POD 활성화 피처를 주문한 후 결과로 생성되는 POD 활성화 코드를 사용하여 대기 프로세서를 활성화해야 합니다.

주:

1. 주문을 처리하고 결과로 생성되는 POD 활성화 코드를 게시하는 데는 며칠이 소요될 수도 있습니다. 추가 용량의 영구 활성화 주문이 처리되는 동안 작업부하 요구사항을 충족하기 위해 14일 시험 기간 동안 대기 프로세서를 활성화할 수 있습니다.
2. POD 활성화 피처는 기타 피처가 주문에 포함되지 않은 경우 더 빠르게 처리할 수 있습니다.

하나 이상의 POD 활성화 피처를 주문하려면 다음 단계를 수행하십시오.

1. 활성화할 대기 프로세서 수를 결정하십시오.
관련 정보를 보려면 프로세서 활성화 시기 결정 주제를 참조하십시오.
2. IBM 비즈니스 파트너나 IBM 영업대표에 문의하여 하나 이상의 POD 활성화 피처를 주문하십시오. 미국 내의 고객은 IBM 웹 사이트에서 주문을 제출할 것을 선택할 수 있습니다.

<http://www.ibm.com> 

3. 서버에서 IBM으로 필수 제품 자료를 송신하십시오.
주문은 주문이 처리되기 전에 서버에서 수집되는 필수 제품 자료(VPD)와 결합합니다. 정보는 서버에 특정한 POD 활성화 코드를 생성하기 위해 사용됩니다. POD 활성화 코드는 빠르게 액세스할 수 있도록 메일로 알린 후 iSeries Capacity Upgrade on Demand 웹 사이트에 게시됩니다.
4. 서버에서 POD 활성화 코드를 입력하여 영구적으로 대기 프로세서를 활성화하십시오.

IBM에 필수 제품 자료 송신: 하나 이상의 POD(Processors on Demand) 활성화 피처를 주문할 경우, IBM에 iSeries 서버의 필수 제품 자료(VPD)를 제공해야 합니다. VPD를 수집하고 Electronic Service Agent^(TM)를 사용하거나 자료를 인쇄 후 팩스로 전송하여 IBM에 보낼 수 있습니다.

Electronic Service Agent를 사용하여 VPD를 수집하여 전송하려면, 다음 단계를 따르십시오.

1. Electronic Service Agent가 설정되어 있는지 확인하십시오.
2. Electronic Service Agent 마법사를 시작하십시오.
3. 필요한 VPD 정보를 수집하고 송신할 하드웨어를 선택하십시오.

VPD를 인쇄하여 팩스로 보내려면, 다음 단계를 수행하십시오.

1. 시스템 콘솔에서, 명령행에 STRSST를 입력하여 시스템 서비스 툴(SST)을 시작하십시오. SST에 사인 온 하십시오.

주: 시스템 서비스 툴을 사용하려면, 시스템 용량과 함께 유효한 서비스 툴 사용자 ID(관리자 특권)를 가지고 있어야 합니다.

2. 옵션 6(시스템 용량에 대한 작업)을 선택하고 Enter를 누르십시오.
3. 옵션 1(시스템 용량 정보 표시)을 선택하고 Enter를 누르십시오. 영구 시스템 용량 표시 화면이 나타납니다.
4. 정보를 인쇄하려면 F6을 누르십시오.
5. 다음 정보가 있는 팩스 문서를 준비하십시오.

- 팩스 수신자 정보:

- 받는 사람: Capacity on Demand 관리자(507-253-7019)
- 팩스 번호: 507-253-4553
- 위치: Rochester, Minnesota

- 팩스 송신자 정보:

- 고객 이름:
- 고객 연락처 이름:
- 고객 주소:
- 고객 전화 번호:
- 고객 팩스 번호:

6. VPD를 다음 번호로 팩스 전송하십시오.
507-253-4553

대기 프로세서 활성화

시험판 또는 영구 기초로 서버에 이미 설치된 대기 프로세서를 하나 이상 활성화할 수 있습니다.

시험 기간 동안 대기 프로세서 활성화

시험 기간 14일 동안 서버에 설치된 모든 대기 프로세서를 활성화할 수 있습니다. 이렇게 하면 추가 용량을 구입할지 혹은 추가 용량의 영구 활성화 주문이 이행되는 동안 최고 작업부하 요구사항을 충족할지 결정하기 전에, 추가 용량이 환경에 어떤 이득이 될 수 있는지를 평가할 수 있습니다.

영구적으로 대기 프로세서 활성화

지정된 수의 대기 프로세서를 영구적으로 활성화할 수 있습니다.

시험 기간 동안 대기 프로세서 활성화: 시험 기간 동안 서버에 설치된 모든 대기 프로세서를 활성화하려면 다음 단계를 수행하십시오.

1. 시스템 콘솔에서, 명령행에 STRSST를 입력하고 시스템 서비스 툴(SST)을 시작하십시오. SST에 사인 온 하십시오.

주: 시스템 서비스 툴을 사용하려면, 시스템 용량에 대한 유효한 서비스 툴 사용자 ID(관리자 특권)를 가지고 있어야 합니다.

2. 옵션 6(시스템 용량에 대한 작업)을 선택하고 Enter를 누르십시오. 시스템 용량에 대한 작업 화면이 나타납니다.
3. 옵션 3(임시 시스템 용량에 대한 작업)을 선택하고 Enter를 누르십시오. 임시 시스템 용량 활성화 시작 확인 화면이 나타납니다.

주: 이 옵션은 임시 시스템 용량 활성화가 사용 중이거나 이미 사용한 경우 나타나지 않습니다.

4. Enter를 눌러 시스템 용량 활성화를 확인하십시오.
5. 시스템 용량에 대한 작업 화면에서 나가십시오.
6. 서버가 파티션된 경우, 새 용량을 사용하기 전에 새로 사용 가능한 프로세서를 논리 파티션에 지정해야 합니다.

활성화된 프로세서를 논리 파티션에 지정하는 방법에 대해서는 처리 파워 동적 이동을 참조하십시오.

7. 서버가 파티션되지 않은 경우, 다음 단계를 수행하십시오.
 - a. 옵션 5(시스템 파티션에 대한 작업)를 선택하고 Enter를 누르십시오.
 - b. 옵션 3(파티션 구성에 대한 작업)을 선택하고 Enter를 누르십시오.
 - c. PRIMARY 옆에 2(파티션 처리 자원 변경)를 입력하고 Enter를 누르십시오.
 - d. 총 활동 프로세서 수를 나타내는 프로세서 수에 해당하는 값을 입력하십시오.

주: 화면 맨 아래에 있는 구성 오류는 기본 1차 파티션이 변경되었으므로(논리 파티션이 현재 사용 중임) 논리 파티션 계획을 참조하여 적절하게 활성화된 프로세서를 서버에 지정해야 함을 나타냅니다.

- e. 변경을 확인하려면 Enter를 누르십시오.
8. SST에서 나가십시오.

이제 새 용량을 사용할 수 있습니다.

주:

1. 시험판은 14일 후, 또는 새 POD 활성화 코드를 입력할 때 자동으로 종료합니다.
2. 서버가 파티션된 경우, 시험 기간 끝에 대기 프로세서를 2차 파티션으로 리턴해야 합니다.

영구적으로 대기 프로세서 활성화: 하나 이상의 POD(Processors on Demand) 활성화 피처를 구입한 경우, 대기 프로세서를 활성화하기 위한 POD 활성화 코드를 받게 됩니다.

대기 프로세서 중 일부 또는 전체를 영구적으로 활성화하려면, 다음 단계를 수행하십시오.

1. 다음과 같이 POD 활성화 코드를 검색하십시오.
 - a. iSeries Capacity Upgrade on Demand 웹 사이트에 액세스하십시오.

<http://www.ibm.com/servers/eserver/series/hardware/ondemand> 

- b. 시스템 유형과 서버의 일련 번호를 입력하십시오.
 - c. 웹 사이트에 표시된 POD 활성화 코드를 기록하십시오.
2. 명령행에서 STRSST를 입력하여 시스템 서비스 툴(SST)을 시작하십시오. SST에 사인 온하십시오.

주: 시스템 서비스 툴을 사용하려면, 시스템 용량에 대한 유효한 서비스 툴 사용자 ID(관리자 특권)를 가지고 있어야 합니다.
 3. 옵션 6(시스템 용량에 대한 작업)을 선택하고 Enter를 누르십시오.
 4. 옵션 2(영구 시스템 용량 활성화)를 선택하고 Enter를 누르십시오.
 5. Processor on Demand 활성화 코드 필드에 POD 활성화 코드를 입력하고 Enter를 누르십시오. 시스템 용량 활성화 확인 화면이 나타납니다.
 6. Enter를 눌러 시스템 용량 활성화를 확인하십시오.
 7. 시스템 용량에 대한 작업 화면에서 나가십시오.
 8. 서버가 파티션된 경우, 새 용량을 사용하기 전에 새로 사용 가능한 프로세서를 논리 파티션에 지정해야 합니다.

활성화된 프로세서를 논리 파티션에 지정하는 방법에 대해서는 처리 전력 동적 이동을 참조하십시오.

9. 서버가 파티션되지 않은 경우, 다음 단계를 수행하십시오.
 - a. 옵션 5(시스템 파티션에 대한 작업)를 선택하고 Enter를 누르십시오.
 - b. 옵션 3(파티션 구성에 대한 작업)을 선택하고 Enter를 누르십시오.
 - c. PRIMARY 옆에 2(파티션 처리 자원 변경)를 입력하고 Enter를 누르십시오.
 - d. 총 활동 프로세서 수를 나타내는 프로세서 수에 해당하는 값을 입력하십시오.

주: 화면 맨 아래에 있는 구성 오류는 디폴트 1차 파티션이 변경되었으므로(논리 파티션이 현재 사용 중임) 논리 파티션 계획을 참조하여 적절하게 활성화된 프로세서를 서버에 지정해야 함을 나타냅니다.
 - e. 변경을 확인하려면 Enter를 누르십시오.
10. SST에서 나가십시오.

주: POD 활성화 코드를 입력할 때, 대기 프로세서는 즉시 활성화됩니다. 그러나 서버는 POD 활성화 코드를 저장하기 위해 15분 동안 실행되어야 합니다. POD 활성화 코드를 입력하고 15분 동안 실행되기 전에 서버를 종료한 경우, 서버를 시작할 때 다시 POD 활성화 코드를 입력해야 할 수도 있습니다.

이제 새 용량을 사용할 수 있습니다.

시나리오: Capacity Upgrade on Demand

Capacity Upgrade on Demand는 고객이 작업부하로 필요로 할 경우에 대기 프로세스를 활성화할 수 있게 합니다. 다음 시나리오는 이 피처의 계획, 준비 및 이점 단계를 보여줍니다.

1. 피처 코드가 2416인 모델 840 서버는 8개의 활동 프로세서와 4개의 대기 프로세서로 작동합니다. 서버 작업부하가 증가함에 따라, 사용할 수 있는 CPU 자원 활용은 사용할 수 있는 용량의 70%에 매우 근접하거나 초과합니다. 추가 자원의 필요성을 예상하고, 관리자는 일부 대기 프로세스를 활성화할 것을 결정합니다.
2. 프로세서를 활성화하기 전에, 관리자는 Capacity Upgrade on Demand용 서버를 준비합니다. 여기에는 추가 프로세서가 어느 정도 필요한지 알아내기 위한 경향 분석 수행과, 추가 프로세서를 지지하기 위한 서버 준비, 그리고 새 용량 주문 준비 등이 포함됩니다.
3. 추가 프로세서 활성화의 이점을 조사하기 위해, 관리자는 시험 기간 동안 프로세서 활성화할 것을 결정합니다. 시험 기간은 14일입니다.
4. 추가 프로세서 보증을 활성화하여 얻게 되는 성능 개선으로 프로세서를 영구적으로 구입하는 것이 보증된다고 결정하면, 관리자는 IBM 영업대표 또는 비즈니스 파트너에게 문의하거나, www.ibm.com을 방문하여 네 개의 POD(Processor on Demand) 활성화 코드의 주문을 제출하십시오.
5. IBM 영업대표는 IBM 구성기에서 주문을 제출하고, 주문이 있는 서버로부터 필수 제품 자료(VPD)를 송신하도록 통지를 수신합니다. VPD 데이터는 팩스로 IBM에 보내거나 Electronic Service Agent를 사용하여 전자식으로 보낼 수 있습니다.
6. 관리자는 웹에서 POD 활성화 코드를 검색하고, 영구 용량을 활성화합니다. 여기에는 대상 서버에 POD 활성화 코드 입력과 프로세서를 논리 파티션에 지정하는 작업이 포함됩니다.

모델 840에는 이제 사용할 수 있는 프로세서가 총 8개 있습니다.

성능 관리 전략 선택

성능 관리 전략을 잘 개발하면 시스템의 성능을 관리하는 데 도움이 됩니다. 성능 관리 전략은 상당 부분 성능 관리에 투입할 수 있는 시간의 양에 의존합니다. 소규모 회사에서 일하는 경우 비즈니스의 여러 가지 많은 부분을 관리하게 되고 성능 관리에 많은 시간을 투입할 수 없습니다. 대다수 대기업들은 성능 관리 전문가를 고용하여 시스템을 조정하고 효율적으로 실행되도록 관리하게 합니다.

기본적인 성능 관리 전략을 결정하고 사용할 성능 어플리케이션을 식별하기 위해 회사를 세 가지 범주 중 하나 즉 소규모 기업, 중간 규모 기업 및 대기업으로 분류하십시오. 각 규모마다 기업 자원은 달라지며 이에 따라 관리 전략도 달라집니다.

소규모 기업

소규모 기업은 대기업에 비해 성능 관리를 전담할 수 있는 자원이 적은 것이 사실입니다. 이러한 이유로 가능한 많이 자동화를 이용하십시오. PM/400을 사용하면 성능 자료가 IBM으로 바로 보내지고 IBM에서는 이 자료를 컴파일하여 보고서를 생성합니다. 이 경우 시간을 절약할뿐 아니라 iSeries 서버에 업그레이드가 필요한 경우 IBM에서 제안할 수도 있습니다.

다음은 소규모 기업에 권장되는 성능 어플리케이션 리스트입니다.

콜렉션 서비스

차후 분석을 위해 사용자 정의 간격으로 샘플 자료를 수집합니다.

그래프 이력

콜렉션 서비스를 사용하여 수집한 성능 자료를 표시합니다.

PM/400

시스템 성능 자료의 수집, 아카이브 및 분석을 자동화합니다.

성능 분석 툴

시스템 성능 정보를 수집, 분석 및 유지보수합니다.

모니터

그래프로 표시된 iSeries 시스템 성능을 관찰하고 사전 정의된 이벤트나 조건에 따라 응답을 자동화합니다.

중간 규모 기업

중간 규모 기업들은 소규모 기업보다는 성능 관리에 투입되는 자원이 많습니다. 그러나 이 경우에도 가능한 많은 부분에서 자동화를 원하며 PM/400의 사용으로 도움을 받을 수도 있습니다.

다음은 중간 규모 기업에 권장되는 성능 어플리케이션 리스트입니다.

콜렉션 서비스

차후 분석을 위해 사용자 정의 간격으로 샘플 자료를 수집합니다.

그래프 이력

콜렉션 서비스를 사용하여 수집한 성능 자료를 표시합니다.

PM/400

시스템 성능 자료의 수집, 아카이브 및 분석을 자동화합니다.

성능 분석 툴

시스템 성능 정보를 수집, 분석 및 유지보수합니다.

모니터

그래프로 표시된 iSeries 시스템 성능을 관찰하고 사전 정의된 이벤트나 조건에 따라 응답을 자동화합니다.

Performance Explorer

특정 어플리케이션이나 시스템 자원에 대한 자세한 정보를 수집합니다.

대기업

대기업에서는 성능 관리에 별도의 자원을 투입합니다.

다음은 대기업에 권장되는 성능 어플리케이션 리스트입니다.

콜렉션 서비스

차후 분석을 위해 사용자 정의 간격으로 샘플 자료를 수집합니다.

그래프 이력

콜렉션 서비스를 사용하여 수집한 성능 자료를 표시합니다.

PM/400

시스템 성능 자료의 수집, 아카이브 및 분석을 자동화합니다.

성능 분석 툴

시스템 성능 정보를 수집, 분석 및 유지보수합니다.

모니터

그래프로 표시된 iSeries 시스템 성능을 관찰하고 사전 정의된 이벤트나 조건에 따라 응답을 자동화합니다.

Performance Explorer

특정 어플리케이션이나 시스템 자원에 대한 자세한 정보를 수집합니다.

iSeries용 iDoctor

시스템과 어플리케이션 성능을 향상시키기 위해 추적 자료를 분석합니다.

PTDV(Performance Trace Data Visualizer)

Java 어플리케이션에서 추적 자료를 봅니다.

성능 관리를 위한 환경 설정

iSeries 서버에는 시스템 성능 자료를 정기적으로 수집하여 시스템의 성능 경향과 잠재적 문제점을 모니터링하는 여러 가지 툴이 들어 있습니다. 사용자의 고유한 요구사항과 환경이 툴과 구성 선택사항을 결정합니다. 시스템을 효율적으로 설정하면 시스템 확장에 따른 정확한 용량 계획을 수행하고 성능 문제 발생시 이를 분석할 수 있습니다.

다음 주제에서 시스템 성능 자료를 수집, 모니터 및 분석하는 툴에 대해 배우고 구성하십시오.

컬렉션 서비스

컬렉션 서비스는 시스템 성능 자료의 정기적 수집을 관리합니다. 이 툴은 정기적으로 자료를 수집하여 컬렉션 오브젝트라고 하는 아카이브를 작성합니다. 이들 컬렉션 오브젝트에는 몇몇 툴에서 직접 액세스할 수 있으며, 사용자 자신의 조화나 다른 툴과 보고서를 사용하여 분석을 위한 데이터베이스 파일 세트로 변환할 수 있습니다. 컬렉션 서비스는 주로 다른 어플리케이션에 자료를 제공하므로 사용 중인 다른 툴은 구성 선택사항(예: 자료 수집 빈도, 수집하는 자료의 유형, 시스템에 자료를 보관하는 기간 등)에 큰 영향을 미칩니다.

Performance Management/400

PM/400은 컬렉션 서비스를 사용하여 비독점 성능 자료를 수집하여 이를 저장 및 전문가 분석을 위해 IBM에 송신합니다. 따라서 사용자 스스로 자료를 저장 및 유지보수할 필요가 없어집니다. 사용자는 웹 브라우저를 사용하여 시스템 성능에 대한 상세 보고서와 권장사항에 액세스할 수 있습니다.

iSeries Navigator 모니터

iSeries Navigator에 포함된 모니터는 컬렉션 서비스 자료를 사용하여 관심있는 시스템 성능의 요소를 추적합니다. 더욱이 CPU 활용 비율이나 작업 상태와 같은 특정 이벤트가 발생할 경우 지정된 조치를 수행할 수 있습니다. 이 주제에서 모니터 사용 방법과 시스템에서 모니터를 설정하는 방법에 대해 배우십시오.

iSeries 성능 관리

성공적인 성능 관리는 시스템이 자원을 효율적으로 사용하도록 하고 iSeries 서버가 사용자 및 사용자의 비즈니스 요구에 가능한 최상의 서비스를 제공하도록 보장해줍니다. 더욱이 효율적 성능 관리를 통해 시스템의 변경에 신속하게 응답할 수 있고 고비용의 업그레이드를 지연시킴으로써 금전과 서비스를 절약할 수 있습니다.

시스템 성능에 영향을 주는 요소를 이해함으로써 문제에 응대하고 좀 더 나은 장기 계획을 수립할 수 있습니다. 효율적인 계획은 잠재적 성능 문제가 발생되지 않도록 하고 현재 및 점증하는 작업부하를 처리할 수 있는 시스템 용량을 보유할 수 있도록 보장합니다.

다음 주제를 통해 시스템의 성능을 유지보수하고 성능 문제를 분석하는 방법을 익히십시오.

성능 추적

일정 기간 동안 시스템 성능을 추적하면 시스템의 성장을 계획할 수 있고 성능 문제의 원인을 분리시켜 식별하는 데 도움이 되는 자료를 가질 수 있게 됩니다. 사용할 어플리케이션과 성능 자료를 정기적으로 수집하는 방법을 익히십시오.

성능 문제 연구

성능 문제를 식별하여 분석하는 데 사용할 수 있는 여러 가지 옵션이 있습니다. 성능 문제의 소스를 찾는 데 도움이 되는 툴과 보고서를 찾아내는 방법을 익히십시오.

성능 자료 표시

성능 자료를 수집한 후 목적에 가장 적합한 툴을 사용하여 자료를 표시하는 방법을 익히십시오.

성능 조정

성능 문제를 식별하였으면 시스템을 조정하여 이를 수정해볼 수 있습니다.

e-business 성능 관리

e-business 환경에서의 성능 관리는 iSeries 관리자를 위한 몇 가지의 새 문제점을 소개합니다. e-business 어플리케이션의 성능을 계획, 추적 및 개선하는 데 도움이 되는 정보 및 자원에 대해서는 이 주제를 참조하십시오.

성능 추적

iSeries 서버의 시스템 성능을 추적하면 시스템 구성을 조정하고 시스템 업그레이드 시기와 방법을 가장 잘 선택하는 데 도움이 되는 경향을 식별해낼 수 있습니다. 더욱이 문제 발생 시 사고 이전과 이후의 성능 자료를 가지고 성능 문제의 원인을 좁혀나가 적절한 해결책을 찾는 것은 필수적입니다.

iSeries 서버에는 성능 경향을 추적하고 iSeries 성능 자료의 이력 레코드를 유지보수하기 위한 여러 개의 어플리케이션이 포함되어 있습니다. 이들 대다수의 어플리케이션은 콜렉션 서비스가 수집한 자료를 사용합니다. 콜렉션 서비스를 사용하여 다음 영역에서 경향을 파악할 수 있습니다.

- 시스템 자원 활용 경향. 이 정보를 이용하여 시스템 구성 변경 및 업그레이드를 계획하고 특별하게 조정할 수 있습니다.
- 구성의 실제 구성요소에 대한 스트레스 식별.

- 최대 및 정상 사용 중 대화식 작업과 일괄처리 작업 사이의 시스템 자원 사용에 대한 균형.
- 구성 변경사항. 콜렉션 서비스를 사용하여 변경(예: 사용자 그룹 추가, 대화식 작업 증가 및 기타 변경)의 영향을 정확하게 예측할 수 있습니다.
- 시스템의 다른 활동에서 문제를 초래할 수 있는 작업의 식별.

다음 툴은 일정 기간 동안 시스템 성능을 모니터링하는 데 도움이 됩니다.

콜렉션 서비스

콜렉션 서비스는 사용자 정의 시간 간격으로 성능 자료를 수집한 다음 이 정보를 사용자 시스템의 콜렉션 오브젝트에 저장합니다. 모니터, 그래프 이력, PM/400 및 성능 분석 툴 사용권 프로그램의 여러 기능 등 다른 많은 툴은 이러한 콜렉션 서비스에 의존하여 자료를 얻습니다.

그래프 이력

그래프 이력은 지정된 기간 동안 콜렉션 서비스가 수집한 성능 자료를 GUI(그래픽 사용자 인터페이스)를 사용하여 표시합니다. 표시에 사용할 수 있는 시간 길이는 콜렉션 오브젝트의 보유 기간과 PM/400의 사용 여부에 따라 달라집니다.

PM/400

PM/400은 시스템 성능 자료의 수집, 아카이브 및 분석을 자동화하며 시스템 자원과 용량 관리를 돕기 위해 보고서를 리턴합니다.

성능 문제 연구

성능 자료를 수집하거나 분석하는 대다수 툴은 추적 또는 샘플 자료를 사용합니다. 콜렉션 서비스는 다양한 시스템 자원에 대해 샘플 자료를 정기적으로 수집합니다. 여러 가지 툴이 이러한 샘플 자료를 분석 및 보고하며 사용자는 이를 사용하여 시스템 자원 활용에 대해 좀 더 폭넓은 시각을 얻게 되며 일반적인 성능 문제에 응답할 수 있습니다. 좀 더 자세한 성능 정보를 제공하기 위해 몇몇 툴은 추적 레벨 자료를 생성합니다. 추적 레벨 자료는 시스템 상의 작업과 어플리케이션의 작동 및 자원 소비에 대한 상세 정보를 제공할 수 있습니다. Performance Explorer 및 STRPFTRC 명령이 추적 자료를 생성하는 일반적인 툴입니다.

예를 들어 시스템의 실행 속도가 느린 경우 시스템 모니터를 사용하여 문제를 찾을 수 있습니다. CPU 활용이 너무 고도할 경우 과도하게 많은 양의 자원을 사용하고 있는 것으로 보이는 작업을 식별할 수 있을 것입니다. 그런 후 구성을 변경하여 문제를 정정할 수 있습니다. 그러나 일부 문제는 추가 정보를 필요로 합니다. 이 작업의 성능에 대한 상세 정보를 얻으려면 Performance Explorer 세션을 시작하고 iSeries 시스템에서 해당 작업의 작동에 대한 상세 정보를 수집하고 원본 프로그램을 변경할 수 있습니다.

성능 자료 수집에 대해 더 자세히 알려면, 다음 주제에서 성능 관리 어플리케이션의 사용 방법과 시기에 대해 배우십시오.

성능 문제 식별

성능 문제를 식별하는 것과 관련한 일반적인 단계에 대해 배우십시오.

일반적인 성능 문제 식별 및 분석

여러 가지 서로 다른 성능 문제가 iSeries 시스템의 공통 영역에 영향을 줍니다. 공통 영역에서 문제를 연구 및 분석하는 방법에 대해 배우십시오.

시스템 성능 정보 수집

컬렉션 서비스는 시스템 성능에 대한 정보를 정기적으로 수집합니다. 성능 자료 분석은 이 정보에서 시작됩니다.

시스템 자원 활용에 대한 정보 수집

여러 가지 톨이 자원(CPU, 디스크 공간, 대화식 용량 및 기타 요소)의 사용 방식을 모니터링합니다. 이러한 톨을 사용하여 문제 영역 식별을 시작할 수 있습니다.

어플리케이션의 성능에 대한 정보 수집

어플리케이션은 여러 가지 이유로 수행 속도가 느려질 수 있습니다. OS/400 및 기타 사용권 프로그램에 들어 있는 여러 가지 톨을 사용하여 자세한 정보를 얻을 수 있습니다.

시나리오: 업그레이드 또는 마이그레이션 후 시스템 성능 개선

이 시나리오에서는 시스템을 막 업그레이드 또는 마이그레이트하였고 이전 보다 실행 속도가 느려진 것처럼 보입니다. 이 시나리오는 성능 문제를 식별하고 수정하도록 돕습니다.

성능 문제 식별

성능 문제를 식별하려면 하드웨어 구성이 작업부하를 감당할만한지 평가하는 것이 중요합니다. CPU 용량은 충분합니까? 여러 어플리케이션이 수행되기에 주 기억장치 용량은 충분합니까? 먼저 용량 모델링 기법을 통해 이 질문에 대한 답을 구하는 것이 나중에 있을 불필요한 수고를 막아줍니다.

문제의 증상을 이해하고 목표가 설정되었다면 분석자는 문제의 원인을 설명할 수 있는 가정을 만들어볼 수 있습니다. 분석자는 OS/400 오퍼레이팅 시스템의 명령 및 톨과 성능 분석 톨 사용권 프로그램을 사용하여 시스템 성능을 측정할 수 있습니다.

측정된 자료를 검토하여 문제를 더 깊게 정의하고 현재의 가정을 검증하거나 거부할 수 있습니다. 명백한 원인이 분리되었다면 그에 대한 솔루션도 제시될 수 있습니다. 한 번에 하나의 솔루션을 처리하면 프로그램을 다시 설계하여 테스트할 수 있습니다. 분석자 톨을 사용하면 여러가지로 솔루션의 효율성을 측정할 수 있고 부작용을 찾아낼 수 있습니다.

최적의 성능을 달성하기 위해서는 중요한 시스템 자원간의 상호 관계를 인식하고 이 자원들 즉, CPU, 디스크, 주 기억장치 및 통신용 리모트 회선 등의 균형을 맞추도록 해야 합니다. 이러한 각각의 자원들이 성능 저하를 유발할 수도 있습니다.

시스템 성능을 향상시키는 것은 이것이 대화식 처리량이건, 대화식 응답 시간이건, 일괄처리 처리량이건 또는 이들의 조합이건 단순히 활동 레벨이나 풀 크기를 조정하는 경우에서부터 어플리케이션 코드 자체를 변경해야 하는 경우에 이르기까지 다양합니다. 이러한 경우 활동 레벨은 처리 장치를 두고 동시에 경쟁할 수 있는 최대 작업 수를 지정하는 서브시스템의 특성입니다.

일반적인 성능 문제 식별 및 분석

iSeries 서버에서 성능 문제가 발생하면 이들은 먼저 시스템의 특정 영역에 영향을 미칩니다. 다음 표에 시스템 영역의 성능 탐색 시 사용할 수 있는 여러 가지 방법이 나와 있습니다. 이들 영역의 대다수는 시스템 모니터 미터법으로 제공됩니다. 그러나 다른 방법으로 이의 정보에 액세스할 수도 있습니다.

영역	설명	사용할 수 있는 툴
프로세서 로드	시스템에 작업이 너무 많지 않은지 또는 일부 작업이 너무 많은 프로세서 시간을 차지하는 않는지 판별합니다.	<ul style="list-style-type: none"> WRKACTJOB(활동 작업에 대한 작업) 명령 WRKSYSACT(시스템 활동에 대한 작업) 명령 - 성능 분석 툴 사용권 프로그램의 일부 iSeries Navigator의 작업 관리 기능 iSeries Navigator 시스템 모니터 내의 CPU 활용 미터법
주 기억장치	결함 및 대기에서 부적합으로 변환 조사.	<ul style="list-style-type: none"> WRKDSKSTS(디스크 상태에 대한 작업) 명령 iSeries Navigator 시스템 모니터 내의 디스크 기억장치 미터법 WRKSYSSTS(시스템 상태에 대한 작업) 명령 iSeries Navigator에서 작업 관리 아래의 메모리 풀(pool) 기능
디스크	암이 너무 적은지 또는 암이 너무 느린지 판별합니다.	<ul style="list-style-type: none"> WRKDSKSTS(디스크 상태에 대한 작업) 명령 iSeries Navigator 시스템 모니터 내의 디스크 암(arm) 활용 미터법 성능 분석 툴 및 구성요소 보고서
통신	늦은 회선, 회선 상의 오류 또는 해당 회선에 사용자가 너무 많은지를 확인합니다.	<ul style="list-style-type: none"> 성능 분석 툴 구성요소 보고서 iSeries Navigator 시스템 모니터 내의 LAN 활용 미터법
IOP	균형이 맞지 않은 IOP가 있는지 또는 IOP가 충분하지 않은지 판별합니다.	<ul style="list-style-type: none"> 성능 분석 툴 구성요소 보고서 iSeries Navigator 시스템 모니터 내의 IOP 활용 미터법
소프트웨어	잠금 및 상호 제외(mutex) 조사.	<ul style="list-style-type: none"> 성능 분석 툴 잠금 보고서 성능 분석 툴 추적 보고서 WRKOBJLCK(오브젝트 잠금에 대한 작업) 명령 iSeries Navigator에서 작업 관리 아래의 의심나는 작업을 마우스 오른쪽 버튼으로 클릭하고 세부사항 —> 잠긴 오브젝트를 선택하십시오.

시스템 성능 자료 수집

자료 수집은 성능을 향상시키기 위한 중요한 단계입니다. 성능 자료를 수집할 때 응답 시간 및 처리량을 이해하는 데 사용할 수 있는 서버에 대한 정보를 수집합니다. 이것은 작업 수행에 관련된 서버 또는 서버 집합의

성능 상태를 캡처하기 위한 한 방법입니다. 자료 컬렉션에서는 나중에 수행될 비교 또는 분석을 위한 문맥이나 시작점을 제공합니다. 첫 번째 자료 컬렉션을 사용할 때 앞으로의 성능을 위한 기준 및 성능 향상을 위한 현재의 시작점이 있습니다. 수집한 성능 자료를 사용하여 응답 시간을 조정 및 향상하고 시스템이 최고의 성능을 달성할 수 있도록 합니다. 성능 문제점 분석은 종종 “무엇이 바뀌었는가?”라는 간단한 질문으로부터 시작됩니다. 성능 자료가 이 질문에 대한 답을 구하는 데 도움이 됩니다.

컬렉션 서비스를 통해 성능 자료를 수집하고, CRTPFDRDTA(성능 자료 작성) 명령을 사용하여 성능 파일을 작성하여 CVTPFRDTA(성능 자료 변환) 명령이나 iSeries의 성능 분석 툴 플러그 인을 통해 이 파일을 현재 릴리스로 변환한 후, 성능 데이터베이스 파일에 들어 있는 정보를 사용하여 보고서를 작성하거나 조회를 작성할 수 있습니다.

성능 자료에 대한 자세한 정보는 다음을 참조하십시오.

컬렉션 서비스

분석을 위한 성능 자료 수집 방법 및 컬렉션 사용자 정의 방법을 참조하십시오.

성능 데이터베이스 파일

사용할 수 있는 성능 데이터베이스 파일의 개요를 찾아 각 성능 데이터베이스 파일에 대한 자세한 필드 자료를 참조하십시오.

또한 성능 관리 API를 사용하여 컬렉션을 시작, 종료 및 순환하고 수집된 자료에 대한 시스템 매개변수를 변경 및 검색할 수 있습니다. 다른 iSeries 성능 분석 툴 및 기술에 대한 자세한 정보는 성능 페이지를 참조하십시오.

시스템 자원 활용에 대한 정보 수집

iSeries 서버와 사용자 애플리케이션이 사용 가능한 자원을 사용하는 방법을 모니터링하고 추적할 수 있도록 돕는 여러 가지 툴이 있습니다. 이 정보를 문제점 분석의 시작점으로 사용하여 용량 계획 및 시스템의 성장을 관리하는 데 도움이 되는 경향을 식별할 수 있습니다.

이들 툴의 사용 방법과 사용 시기에 대해 배우려면 다음 주제를 참조하십시오.

iSeries Navigator 모니터

iSeries Navigator에 들어 있는 모니터는 다양한 미터법으로 현재 및 최신 자료를 제공합니다. 또한 특정 이벤트 발생 시 지정된 조치를 취하도록 이들 모니터를 구성할 수도 있습니다.

OS/400 성능 명령

OS/400에는 시스템 성능을 관리 및 유지보수하도록 돕는 여러 가지 중요한 기능이 포함되어 있습니다.

PM/400

PM/400은 컬렉션 서비스를 사용하여 비독점 성능 자료를 수집하여 이를 저장 및 전문가 분석을 위해 IBM에 송신합니다. 따라서 사용자 스스로 자료를 저장 및 유지보수할 필요가 없어집니다. 사용자는 웹 브라우저 사용하여 시스템 성능과 경 분석에 대한 상세 보고서와 권장사항에 액세스할 수 있습니다.

어플리케이션의 성능에 대한 정보 수집

어플리케이션의 성능에 대한 정보 수집은 시스템 성능에 대한 정보를 수집하는 것과 많이 다릅니다. 어플리케이션 정보를 수집하는 것은 특정의 성능 어플리케이션(예: Performance Explorer, PTDV(Performance Trace Data Visualizer) 및 iDoctor)을 사용해야만 수행할 수 있습니다. 대안으로 개별 서버 성능을 추적하기 위한 작업 모니터와 서버 작업을 추적 및 분석하기 위한 성능 분석 툴을 사용하여 어플리케이션 성능의 개요를 얻을 수 있습니다.

주: 어플리케이션의 성능 자료를 수집하는 것은 시스템의 성능에 중대한 영향을 줄 수 있습니다. 콜렉션을 시작하기 전에 다른 모든 콜렉션 옵션을 시도해 보도록 하십시오.


Performance Explorer

이 툴은 일반적인 성능 모니터링을 수행하는 툴을 사용하여서는 식별할 수 없는 성능 문제의 원인을 찾는 데 도움이 됩니다. 컴퓨터 환경은 크기와 복잡성이 점차 커지므로 성능 분석에서도 복잡성이 증대되는 것은 당연합니다. Performance Explorer는 복잡한 성능 문제에 대한 자료를 수집하여 복잡성 증대에 따른 문제를 해결합니다.

Performance Explorer는 프로그램의 성능에 대한 이해 및 개선에 관심이 있는 어플리케이션 개발자를 위해 고안되었습니다. 또한 복잡한 성능 문제를 식별하여 분리해낼 수 있는 성능 관리에 대한 지식이 있는 사용자에게도 도움이 됩니다.

iSeries용 PTDV(Performance Trace Data Visualizer)

이 툴은 iSeries에서 실행 중인 어플리케이션의 성능 분석에 사용할 수 있는 Java 어플리케이션입니다. PTDV는 OS/400 오퍼레이팅 시스템의 Performance Explorer 기능에 대해 작업하여 분석자가 프로그램 흐름을 보고 추적, 작업, 스레드 및 프로시저어별로 요약된 세부사항(예: CPU 시간, 현재 시스템 시간, 순환 수, 명령어 수)을 얻을 수 있도록 합니다. Java 어플리케이션 추적을 가시화할 때 추가 세부사항(작성된 오브젝트의 수 및 유형)과 Java 잠금 작동에 대한 정보를 표시할 수 있습니다. 또한 WebSphere Application Server가 생성하는 Performance Explorer 이벤트에 대한 지원도 들어 있습니다. PTDV를 이용하면 열의 정렬, 자료 내보내기 및 다양한 레벨의 자료 요약도 가능합니다.

추가 정보는 Performance Trace Data Visualizer  웹 사이트를 참조하십시오.

iDoctor for iSeries

iDoctor의 Performance Explorer Analyzer 기능에는 시스템 및 어플리케이션 성능을 개선하기 위한 추적 자료 분석에 중점을 둔 소프트웨어 툴이 들어 있습니다. 이러한 상세 분석을 통해 디스크 조작, 파일 열기 조작, 기계 인터페이스(MI) 프로그램, 대기 상태, 디스크 공간 소비량 등에 대한 자세한 요약이 제공됩니다. 클라이언트 구성요소는 iSeries 추적 자료를 집약하여 그래프로 표시할 수 있도록 하는 iSeries Navigator 플러그인입니다.

STRPFRTRC(성능 추적 시작) 명령

OS/400에는 멀티 프로그래밍 및 트랜잭션 자료를 수집하기 위한 명령이 들어 있습니다. 이 명령은 이전 릴리스에서 STRPFRMON 명령이 수집했던 자료를 수집합니다. 이 명령을 실행한 후 DMPTRC(추적 덤프) 명령을 사용하여 자료를 데이터베이스 파일로 내보낼 수 있습니다.

추적 자료 덤프: 덤프가 시스템 성능에 영향을 미치므로 추적 자료의 덤프 시기를 결정하는 것은 아주 중요한 결정사항입니다. DMPTRC(추적 덤프) 명령은 내부 추적 표의 정보를 데이터베이스 파일에 넣습니다. 작업 부하가 큰 시스템이나 고 우선순위(대화식) 작업 내에서 최대 활동 시간에 추적 자료를 덤프하는 것은 좋지 않습니다. 추적 덤프를 지연시킬 수는 있지만 자료가 존재한다는 사실을 잊기 전에 자료를 덤프하도록 하십시오. 어떤 이유로 추적표가 지워진 경우 추적 자료를 유실하게 됩니다. 그러나 덤프를 약간 지연시킨 후 DMPTRC 명령을 사용하여 추적 자료를 일괄처리 작업으로 덤프하면 성능을 유지할 수 있습니다.

추적 자료를 덤프하려면 다음 명령을 발행하십시오.

```
DMPTRC MBR(member-name) LIB(library-name)
```

자료를 저장할 라이브러리명과 멤버명을 지정해야 합니다. 추적 정보를 수집할 때 컬렉션 서비스를 사용하여 샘플 자료를 수집할 수 있습니다. 이와 같이 샘플 자료와 추적 자료를 함께 수집하는 경우 동일하게 이름이 붙여진 멤버에 이의 자료를 배치해야 합니다. 다시 말해 CRTPFRTDA TOMBR 및 TOLIB 매개변수에 제공하는 이름이 DMPTRC MBR 및 LIB 매개변수에 제공하는 이름과 같아야 합니다.

시나리오: 업그레이드 또는 마이그레이션 후 시스템 성능 개선

최근에 iSeries 서버를 최신 릴리스로 업그레이드하였습니다. 업그레이드를 완료한 후 정상 조작을 재개한 후 시스템 성능이 크게 저하되었습니다. 성능 문제의 원인을 식별하고 시스템을 정상 성능 레벨로 복원하고자 합니다.

변경 분리

몇 가지 문제가 오퍼레이팅 시스템을 업그레이드한 후 성능 저하를 초래할 수 있습니다. OS/400에 들어 있는 성능 관리 툴과 성능 관리 툴 사용권 프로그램(5722-PT1)을 사용하여 성능 문제에 대한 좀 더 자세한 정보를 얻고 예상되는 문제의 원인을 좁혀나갈 수 있습니다.

1. CPU 활용을 검사합니다. 업그레이드 후 작업에 필요한 일부 자원에 액세스할 수 없는 경우가 종종 있습니다. 이 결과 하나의 작업이 용납할 수 없을 정도로 많은 양의 CPU 자원을 소비하게 됩니다.
 - WRKSYSACT, WRKSYSSTS, WRKACTJOB 또는 iSeries Navigator 시스템 모니터를 사용하여 전체적인 CPU 활용을 확인하십시오.
 - CPU 활용이 높을 경우(예를 들어 90% 이상), 활동 작업에서 사용하는 CPU의 양을 검사하십시오. CPU 자원의 30% 이상을 하나의 작업에서 소비하는 경우 파일 호출이나 오브젝트를 누락할 수 있습니다. 벤더 공급 프로그램에 대해서는 벤더에게 문의하고 작업의 소유자나 프로그래머에게 추가 지원을 요청하십시오.
2. STRPFRTRC 명령을 시작한 후 시스템 및 구성요소 보고서를 사용하여 다음의 가능한 문제를 식별하고 수정하십시오.
 - 기계 풀(pool)에 대한 페이지 결함 비율이 초당 10건 이상을 경우 결함 비율이 이 레벨 아래로 떨어질 때까지 기계 풀에 더 많은 메모리를 제공하십시오.
 - 디스크 활용이 40% 이상일 경우 대기 및 서비스 시간을 살펴보십시오. 이들 값이 수용 가능한 경우 우선순위를 관리하기 위해 작업부하를 줄여야 할 수 있습니다.
 - IOP 활용이 60% 이상일 경우, IOP를 추가하고 여기에 일부 디스크 자원을 할당하십시오.

- 사용자 풀(pool)의 페이지 결합이 허용 불가능할 정도로 높을 경우, 성능 자동 조정 주제를 참조하십시오.
3. 작업 요약 보고서를 실행하고 **점유 또는 잠금 충돌 보고서**를 참조하십시오. 점유 또는 잠금 충돌의 수가 클 경우 액세스 경로 크기가 1TB로 설정되어 있는지 확인하십시오. 점유 또는 잠금 충돌이 사용자 프로파일에 있고 참조된 사용자 프로파일이 여러 개의 오브젝트를 소유하는 경우 이 프로파일이 소유하는 오브젝트의 수를 줄이십시오.
 4. **타스크 전환 옵션**을 사용하여 5분 동안 iDoctor를 실행하십시오. 그런 후 얻어진 추적 자료를 **타스크 전환 모니터**를 사용하여 분석하십시오. 다음을 식별 및 분석하십시오.
 - 작업의 CPU 대기
 - 작업 결합
 - 점유 충돌

성능 자료 표시

성능 자료를 표시하면 시스템의 성능을 좀 더 정확하게 분석할 수 있습니다. 성능 자료는 여러 가지 다른 방식으로 표시할 수 있습니다. 그러나 몇몇 상황에서 더 적합한 특성의 성능 어플리케이션이 있을 것입니다. 대다수 어플리케이션은 콜렉션 서비스를 사용하거나 성능 추적을 통해 수집된 자료를 표시합니다. 이 자료에 액세스하는 최상의 방법은 성능 문제를 분석할지의 여부, 차후의 시스템 성장을 계획하기 위해 시스템 성능을 모니터링하는지의 여부 또는 경향을 식별하는 중인지의 여부에 따라 달라집니다.

실시간 성능 자료 표시

다음 툴을 이용하여 현재 또는 아주 최근의 성능 정보를 표시할 수 있습니다.

OS/400 명령

기본 오퍼레이팅 시스템에는 시스템 성능의 특정 영역에 대한 현재 정보를 표시하는 다수의 명령이 있습니다.

성능 분석 툴 표시 화면

성능 분석 툴 사용권 프로그램에는 콜렉션 서비스 콜렉션 오브젝트의 성능 자료를 표시하는 iSeries Navigator용 플러그 인이 들어 있습니다. 또한 시스템 상의 작업에 대한 자세한 정보를 보고 성능 분석 툴 보고서를 인쇄할 수도 있습니다.

시스템 및 작업 모니터

이들 모니터는 많은 시스템 요소에 대한 성능 자료를 표시합니다. 모니터 자료는 콜렉션 오브젝트에 기반하며 콜렉션 서비스의 콜렉션 간격에 따라 수집되는 자료를 표시합니다.

이력 성능 자료 표시

다음 툴을 사용하여 시스템에 저장된 자료를 볼 수 있습니다.

PM/400

PM/400은 시스템 성능 자료의 수집, 아카이브 및 분석을 자동화하며 시스템 자원과 용량 관리를 돕기 위해 보고서를 리턴합니다.

그래프 이력

그래프 이력은 콜렉션 서비스에서의 보유 기간에 따라 최대 일주일간 효력을 갖는 성능 자료를 그래픽 화면으로 제공합니다. PM/400에서 그래프 이력은 더 긴 기간의 자료 콜렉션을 표시할 수 있습니다.

성능 조정

성능 조정의 주된 목적은 서버가 시스템 자원을 최대한 활용할 수 있도록 하고 작업부하를 가능한 효율적으로 실행할 수 있도록 하는 것입니다. 성능 조정은 시스템 성능을 수작업으로 또는 자동으로 조정하는 방법입니다. 시스템을 조정하는 옵션은 여러가지가 있습니다. 각 시스템 환경은 고유하므로 성능을 관찰하여 사용자의 환경에 가장 적합하도록 조정을 해야 합니다. 즉 일상적인 성능 모니터링을 수행해야 합니다. 성능 조정 이전에 수행해야 하는 성능 모니터링에 대한 자세한 정보는 성능 관리를 참조하십시오.

IBM에서는 디스크에서 읽혀지는 실제 입출력 요구 수를 줄임으로써 입출력 서브시스템과 시스템 응답 시간 모두를 향상시킬 수 있는 툴을 제공합니다. Extended Adaptive Cache를 사용하여 시스템 성능을 향상시키는 방법을 익히십시오.

성능 조정에 대한 자세한 정보는 다음 주제에서 선택하십시오.

기본 성능 조정

시스템의 성능을 조정하려면 초기 성능 조정 값을 설정하고 시스템 성능을 관찰한 후 값을 검토하여 조정할 내용을 결정해야 합니다.

자동 성능 조정

대부분의 사용자가 자동으로 성능 조정이 이루어지도록 시스템을 설정해야 합니다. 새로운 시스템이 출하될 때 자동으로 조정되도록 구성됩니다.

기본 성능 조정

성능 조정을 시작하려면 먼저 초기 기계와 사용자 풀 크기를 판별하여 초기 조정 값을 설정해야 합니다. 그런 후 시스템 성능을 관찰할 수 있습니다.

초기 조정 값 설정

초기 조정 값 설정에는 시스템을 효율적으로 조정하기 위해 시스템 풀 크기와 활동 레벨을 처음 구성할 때 수행하는 단계가 포함됩니다. 초기 값은 산정값에 기준하므로 시스템이 사용 중일 때 추가 조정이 필요할 수 있습니다. 다음 단계는 초기 조정 값을 설정합니다.

- 초기 기계 풀 크기 판별
- 초기 사용자 풀 크기 판별

시스템 성능 관찰

시스템 성능을 관찰하기 위해 WRKSYSSTS(시스템 상태에 대한 작업), WRKDSKSTS(디스크 상태에 대한

작업) 및 WRKACTJOB(활성 작업에 대한 작업) 명령을 사용할 수 있습니다. 각 관찰 시마다 성능 목표와 비교하여 시스템 성능 측정치를 검사 및 평가해야 합니다.

1. 비정기적 시스템 활동을 제거합니다. 서버 성능의 저하를 초래할 수 있는 비정기적 활동으로는 대화식 프로시저 컴파일, 통신 오류 복구 프로시저(ERP), 조회 파일 열기(OPNQRYF), 어플리케이션 오류 및 사인 오프 활동 등이 있습니다.
2. WRKSYSSTS, WRKDSKSTS 및 WRKACTJOB 명령을 사용하여 성능 자료를 표시할 수 있습니다. 성능 분석 툴 명령, WRKSYSACT(시스템 활동에 대한 작업)를 사용하여 성능 자료를 표시할 수도 있습니다.
3. 시스템이 최소 5분간 자료를 수집할 수 있습니다.
4. 성능 목표와 비교하여 성능 측정치를 평가합니다. 일반적으로 측정에는 다음이 포함됩니다.
 - WRKACTJOB 화면에서 사용할 수 있는 대화식 처리량 및 응답 시간
 - 일괄처리 처리량. 활동 일괄처리 작업에 대한 보조 입/출력(AuxIO) 및 CPU 비율(CPU%) 값을 관찰합니다.
 - 스푼 처리량. 활동 출력기 작업에 대한 보조 입/출력(AuxIO) 및 CPU 비율(CPU%) 값을 관찰합니다.
5. 기대에 부합하지 않는 성능 자료가 관찰되면 새 자료를 근거로 시스템을 조정하십시오. 다음에 유의하십시오.
 - 모든 키 성능 측정치를 측정 및 비교합니다.
 - 한 번에 하나씩 조정하고 평가합니다.

성능 검토

적절한 성능 값을 설정하였으면 시스템이 지속적으로 잘 수행되는지 확인하기 위해 정기적으로 시스템을 점검해야 합니다. 지속적인 성능 조정에는 시스템의 성능 관찰, 권장 값으로 조정 등이 포함됩니다.

의미있는 통계를 수집하려면 일반 활동 레벨에서 시스템 성능을 관찰해야 합니다. 예를 들어 시스템에 실행 중인 작업이 없는 동안 수집된 통계는 시스템 성능 평가에 있어서 별 의미가 없습니다. 최선을 다했음에도 불구하고 성능이 만족스럽지 못하면 구성을 평가해야 합니다. 목표 충족을 위해 다음을 고려하십시오.

- 프로세서 업그레이드
- 추가 기억장치 및 제어기
- 추가 주 기억장치
- 어플리케이션 수정

이 접근방식 중 하나 이상을 적용하여 원하는 성능 목표를 달성해야 합니다. 노력에도 불구하고 여전히 만족할만한 성능 목표가 얻어지지 않으면 수행 중인 작업 유형에 비해 성능 목표가 타당한지 검토해야 합니다.

조정 내용 결정

시스템 성능이 저하되어 조정이 필요한 경우 성능 문제 연구를 참조하여 성능 문제의 원인을 식별하고 정정하십시오.

자동 성능 조정

시스템은 효율적인 시스템 자원의 사용을 제고하기 위해 성능 값을 자동적으로 설정합니다. 다음을 수행하여 시스템 성능을 자동으로 조정하도록 시스템을 설정할 수 있습니다.

- 기억장치 풀 크기 및 활동 레벨 조정
- 기억장치 풀 페이지징 조정

기억장치 풀 크기 및 활동 레벨 조정

QPFRADJ 시스템 값을 사용하여 기억장치 풀과 활동 레벨의 자동 조정을 제어할 수 있습니다. 이 값은 시스템 재시작(IPL) 시 값을 조정해야 하는지 재시작 후 정기적으로 값을 조정해야 하는지를 나타냅니다.

IPL 시 성능 조정 또는 동적으로 조정 중 한 방식으로 성능을 조정하도록 시스템을 설정할 수 있습니다.

- 시스템 재시작(IPL) 시에만 조정하도록 시스템을 설정하려면 iSeries Navigator에서 구성 및 서비스 -> 시스템 값 -> 성능을 선택하십시오. 메모리 풀 탭을 마우스 오른쪽 버튼으로 클릭하고 메모리 풀 및 활동 레벨 자동 조정 아래의 시스템 재시작 시를 선택하십시오. 이것은 QPFRADJ 시스템 값을 1로 설정하는 것과 같습니다.
- 시스템 재시작(IPL) 시에 기억장치 풀을 조정하고 재시작 후 정기적으로 기억장치 풀을 정기적으로 조정하도록 시스템을 설정하려면 iSeries Navigator에서 구성 및 서비스 -> 시스템 값 -> 성능을 선택하십시오. 메모리 풀 탭을 클릭하고 메모리 풀 및 활동 레벨 자동 조정 아래의 시스템 재시작 시와 재시작 후 정기적으로를 모두 선택하십시오. 이것은 QPFRADJ 시스템 값을 2로 설정하는 것과 같습니다.
- 재시작 후에 정기적으로 기억장치 풀을 조정하고 시스템 재시작(IPL) 시에는 조정하지 않도록 시스템을 설정하려면 iSeries Navigator에서 구성 및 서비스 -> 시스템 값 -> 성능을 선택하십시오. 메모리 풀 탭을 클릭하고 메모리 풀 및 활동 레벨 자동 조정 아래의 재시작 후 정기적으로를 선택하십시오. 이것은 QPFRADJ 시스템 값을 3으로 설정하는 것과 같습니다.

기억장치 풀 값은 시스템 재시작(IPL) 시 초기 값으로 재설정되지 않습니다.

기억장치 풀 페이지징 조정

시스템이 제공하는 동적 조정 지원은 시스템의 성능을 개선하기 위해 공유 풀에 대한 풀 크기와 활동 레벨을 자동으로 조정합니다. 이러한 조정은 사용이 적은 기억장치 풀을 더 많은 기억장치를 필요로 하는 풀로 이동하여 수행됩니다. 또한 이러한 조정은 풀에 있는 스레드의 수와 풀에 할당된 기억장치가 균형을 이루도록 활동 레벨을 설정합니다. 시스템을 조정하기 위해 튜너는 스레드의 수에 따라 계산된 가이드라인을 참조합니다.

동적 조정이 사용 중인 경우 다음의 성능 값은 자동으로 적절한 설정으로 변경됩니다.

- 기계(*MACHINE) 메모리 풀 크기(QMCHPOOL 시스템 값)
- 기본(*BASE) 메모리 풀 활동 레벨(QBASACTLVL 시스템 값)
- 공유 풀 *INTERACT에 대한 풀 크기와 활동 레벨
- 공유 풀 *SHRPOOL에 대한 풀 크기와 활동 레벨
- 공유 풀 *SHRPOOL1-*SHRPOOL60에 대한 풀 크기와 활동 레벨

동적 조정이 사용 중일 경우(QPFRADJ 시스템 값이 2 또는 3으로 설정됨) 프로파일 QSYS 아래에서 실행되는 QPFRADJ 작업은 시스템에서 사용 중으로 나타납니다.

메모리 풀에 대한 자세한 정보는 메모리 풀을 참조하십시오.

e-business 성능 관리

e-business 환경의 성능은 iSeries 시스템 관리자에게 몇 가지의 복잡한 문제점이 있습니다. iSeries 서버를 켜는 루틴 외에도, 관리자는 또한 e-business 트랜잭션을 지원하는 하드웨어 및 서비스를 모니터링하고 최적화해야 합니다.

다음 주제는 서버의 e-business 성능을 최대화하기 위한 중요한 고려사항 중 일부에 익숙하게 되는 데 도움을 줄 수 있으며, 자세한 권장사항 및 예에 추가 자원 링크를 제공합니다.

클라이언트 성능

iSeries 시스템 관리자가 종종 e-business 네트워크의 클라이언트측 제어를 약간 받지만, 이들 권장사항을 사용하여 클라이언트 장치가 eBusiness 환경에 최적화되도록 할 수 있습니다.

네트워크 성능

네트워크 설계, 하드웨어 자원 및 통신 압력은 종종 e-business 어플리케이션의 성능에 중요한 영향을 미칩니다. 네트워크 성능을 최적화하고 iSeries 통신 자원을 조정하는 방법에 대해서는 이 주제를 사용할 수 있습니다.

OS/400에서의 Java 성능

OS/400은 iSeries 서버에서의 Java 어플리케이션이나 서비스의 성능을 최적화하기 위한 몇 가지의 구성 옵션 및 자원을 제공합니다. OS/400에서 Java 환경에 대해 배우고 Java 기반 어플리케이션으로부터 최상의 가능한 성능을 얻는 방법을 배우려면 이 주제를 사용하십시오.


HTTP Server 성능

HTTP 서버는 종종 iSeries 서버에서 e-business 성능의 중요한 부분입니다. IBM은 이 서버의 최외곽을 확보할 수 있도록 하는 몇 가지의 옵션 및 구성 선택사항을 제공합니다.

WebSphere 성능

WebSphere Application Server는 iSeries 서버의 e-business 어플리케이션 전개 환경 선택사항입니다. WebSphere 환경에서 성능을 계획하고 최적화하는 방법을 배우려면 이 주제를 사용하십시오.

이러한 특정 권장사항 외에도, 관리자는 다음 주제에도 익숙해야 합니다.

- 작업 관리
- Java for iSeries
- HTTP 서버
- Domino for iSeries sizing and performance tuning 

클라이언트 성능

웹 브라우저가 있는 PC로 구성되는 클라이언트는 종종 관리자가 최소의 직접 제어를 가지고 있는 eBusiness 구성요소를 제시합니다. 그러나 이러한 구성요소는 계속해서 웹 어플리케이션의 엔드투엔드 응답에 중요한 영향을 줍니다.

최고급의 성능을 얻을 수 있도록 하려면, 클라이언트 PC가 다음을 갖추고 있어야 합니다.

- 적절한 메모리를 가지고 있어야 합니다. 복잡한 양식 및 그래픽을 사용하는 인터페이스 및 자원 집약 애플릿은 클라이언트 프로세서에 요구를 제시할 수도 있습니다.
- 고속의 최적화된 네트워크 연결을 이용하십시오. 클라이언트 PC의 많은 통신 어댑터는 네트워크 환경에 맞게 최적화되지 않아도 기능할 수 있습니다. 자세한 정보는 통신 하드웨어에 관한 문서를 참조하십시오.
- 필요한 기술을 완전히 지원하는 브라우저를 사용하십시오. 또한, 브라우저 지원 및 성능은 웹 인터페이스를 설계할 때 특히 주의를 기울여야 합니다.

네트워크 성능

네트워크는 종종 웹 어플리케이션의 응답 시간에 주요 역할을 수행합니다. 또한, 네트워크 구성요소의 성능 영향은 네트워크 통신량 및 사용할 수 있는 대역폭이 자주 변경되고 시스템 관리자가 직접 제어할 수 없는 영향을 받으므로 종종 측정하기가 복잡하면서 어렵습니다. 그러나 iSeries 서버에서 통신 자원을 모니터링하고 조정하는 데 도움을 주기 위해 사용할 수 있는 몇 가지의 자원이 있습니다.

자세한 정보는 다음 주제를 참조하십시오.

콜렉션 서비스

콜렉션 서비스는 정기적으로 통신 자원의 성능 자료를 수집합니다. 특히 관심이 가는 것은, 성능 자료 파일 QAPMTCP 및 QAPMTCPIFC가 TCP 서버의 정보를 저장한다는 것입니다. 파일을 직접 조회하거나 성능 분석 툴 사용권 프로그램에 포함된 보고서를 사용하여 이 데이터를 참조할 수 있습니다.

시스템 모니터

시스템 모니터를 사용하여 통신 하드웨어를 포함한 시스템 자원을 iSeries 서버에서 사용하는 방법에 대한 정보를 제공할 수 있습니다. 특히, 시스템 모니터 내에서 회선 활용 및 IOP 미터법은 네트워크 성능에 관한 유익한 자료를 제공할 수 있습니다.

성능 추적

몇 가지의 어플리케이션 및 툴에서는 iSeries 서버의 통신 자원에 필요한 성능 데이터를 정기적으로 수집하고 시간이 지남에 따라 성능을 모니터링할 수 있습니다.

iSeries Performance Capabilities Reference

Performance Capabilities Reference는 최적의 성능이 나올 수 있도록 iSeries 서버를 구성하거나 조정하는 데 도움이 될 수 있는 자세한 정보, 보고서 및 예를 제공합니다. 특히, iSeries 통신 자원 계획 및 관리에 도움이 되는 제 5 장: Communications Performance를 참조하십시오.

iSeries Network.com 

이 웹 사이트는 네트워크 계획 및 자원을 최적화하는 데 필요한 많은 자원을 제공합니다. 특히, “Cultivate your AS/400 Networks” 및 “8 tools for better network performance”를 참조하십시오.

OS/400에서의 Java 성능

Java는 종종 웹 기반 애플리케이션의 언어 선택사항입니다. 그러나 Java 애플리케이션은 최적의 성능을 얻기 위해 OS/400 실행 환경 및 Java 애플리케이션 둘 다의 최적화를 다소 요구할 수 있습니다.

OS/400에서 Java 환경에 대해 배우고 성능 분석 및 개선을 위해 사용할 수 있는 추가 정보 및 툴에 대해 배우려면 다음 자원을 참조하십시오.

Java 성능

Javav 기반 애플리케이션으로부터 최상의 성능을 얻을 수 있도록 도와주는 몇 가지의 중요한 구성 선택 사항 및 툴이 있습니다.

애플리케이션의 성능에 대한 정보 수집

OS/400에서 애플리케이션 성능을 모니터링하고 조정하는 데 사용할 수 있는 몇 가지의 툴이 있습니다. 애플리케이션 성능을 측정하고 개선하는 데 도움이 될 성능 추적사항, Performance Explorer(PEX) 및 유사한 툴을 사용하는 방법을 배우려면 이 주제를 참조하십시오.

iSeries Performance Capabilities Reference

Performance Capabilities Reference는 최적의 성능이 나올 수 있도록 iSeries 서버를 구성하거나 조정하는 데 도움이 될 수 있는 자세한 정보, 보고서 및 예를 제공합니다. 특히, Java 애플리케이션의 성능을 최적화하고 Java 프로그래밍에 관한 성능 추가 정보를 배우려면 7장: Java Performance를 참조하십시오.

Java and WebSphere performance in OS/400

Java 및 WebSphere 성능을 최대화하고, 성능 데이터를 수집하여 분석하는 데 도움이 되도록 운영 환경을 계획하고 구성하는 방법에 대해 배우려면 이 레드북을 참조하십시오.

WebSphere J2EE application development for the IBM eServer iSeries server

이 레드북은 J2EE 소개와, iSeries 서버에서 J2EE 애플리케이션을 성공적으로 구현하는 데 도움이 될 제안사항 및 예를 제공합니다.

성능 정보 외에도, Java 주제는 iSeries 서버에서 Java 애플리케이션을 개발하고 전개하는 데 필요한 자원을 제공합니다.

HTTP Server 성능

HTTP 서버는 웹 기반 애플리케이션의 엔드투엔드 성능에서 중요한 역할을 수행하며, 몇 가지 새 개선점으로 인해 웹 서버 성능을 효율적으로 모니터링하고 개선시킬 수 있습니다. 특히, 새 FRCA(Fast Response Caching Accelerator)는 탁월한 정적 환경에서 특히 HTTP 서버 성능을 개선할 수 있게 합니다.

HTTP 서버 성능을 최대화하는 방법에 대해서는 다음 자원을 참조하십시오.

콜렉션 서비스

콜렉션 서비스를 사용하여 HTTP 서버 성능 자료를 수집하고 시간이 지남에 따라 결과를 모니터링할 수 있습니다. 성능 데이터 파일 QAPMHTTPB 및 QAPMHTTPD는 수집 간격마다 HTTP 서버 자료를 저장합니다. QAPMHHTB는 기본 정보를 제공하고, QAPMHTTPD는 더 자세한 통계를 제공합니다. 이러한 자료 파일을 직접 조회하거나, 성능 분석 툴 사용권 프로그램에서 시스템 및 구성요소 보고서를 참조할 수 있습니다.

iSeries용 IBM HTTP Server

iSeries에서 HTTP 서버를 설정, 구성 및 관리하는 방법에 대한 정보는 이 주제를 참조하십시오. 이 주제에는 또한 이 제품에 대한 FRCA(Fast Response Caching Accelerator)와 같은 최신 확장 기능에 관한 설명도 있습니다.

iSeries Performance Capabilities Reference

Performance Capabilities Reference는 최적의 성능이 나올 수 있도록 iSeries 서버를 구성하거나 조정하는 데 도움이 될 수 있는 자세한 정보, 보고서 및 예를 제공합니다. 특히, HTTP 서버 성능 스펙, 계획 정보 및 성능 추가 정보에 대해서는 제 6 장: Web Server and Web Commerce를 참조하십시오.

HTTP server(Powered by Apache)

공동 사용 시나리오에서의 HTTP Server 구성을 위한 예를 포함하여, OS/400에서의 HTTP Server(Apache로 구동)에 관한 자세한 설명을 보려면 이 레드북을 사용하십시오.

AS/400 HTTP Server Performance and Capacity Planning

HTTP 서버가 성능 조정 및 계획에 미치는 영향에 대해 배우려면 이 레드북을 사용하십시오. 이 서적에는 또한 웹 서버 성능 자료를 수집, 해석하고 응답하기 위해 iSeries 성능 관리 툴을 사용할 경우의 제안 사항도 있습니다.

WebSphere 성능

WebSphere 환경에서 iSeries 서버 성능을 관리하는 경우, iSeries 관리자에게 몇 가지의 도전이 있습니다. 웹 기반 트랜잭션은 더 많은 자원을 소비할 수 있고, 일반적인 통신 작업부하와는 다르게 소비할 수 있습니다.

최적의 성능을 계획하고 WebSphere 환경에서 서버 자원을 조절하는 방법에 대해 배우려면 다음 주제 및 자원을 참조하십시오.

WebSphere Application Server performance considerations

이 웹 사이트는 iSeries 서버에서 각 WebSphere Application Server 버전에 필요한 자원과 많은 유용한 성능 추가 정보 및 권장사항을 제공합니다. 이러한 자원은 특히 servlet, JSP(Java Server Page) 및 EJB(Enterprise Java Bean)를 사용하는 환경에 유익합니다.

DB2 UDB/WebSphere Performance Tuning Guide

이 레드북은 WebSphere 및 DB2 환경 모두를 소개하고, WebSphere 및 DB2 성능을 최적화하도록 도울 수 있는 공통 성능 문제점의 제안사항, 예 및 해결책을 제공합니다.

Java and WebSphere performance in OS/400

Java 및 WebSphere 성능을 최대화하고, 성능 데이터를 수집하며 분석하는 데 도움이 되도록 운영 환경을 계획하고 구성하는 방법에 대해 배우려면 이 레드북을 사용하십시오.

WebSphere V3 Performance Tuning Guide

이 레드북은 iSeries 서버에서 WebSphere V3 성능을 최적화하기 위한 자세한 권장사항 및 예를 제공합니다.

iSeries Performance Capabilities Reference

Performance Capabilities Reference는 최적의 성능이 나올 수 있도록 iSeries 서버를 구성하거나 조정하는 데 도움이 될 수 있는 자세한 정보, 보고서 및 예를 제공합니다. 특히, WebSphere 특정 성능 추가 정보에 대해서는 제 6 장: Web Server and Web Commerce를 참조하십시오.

기타 WebSphere 및 e-business 정보 자원에 대해서는 주제 WebSphere e-business 관리를 참조하십시오.

성능 관리를 위한 어플리케이션

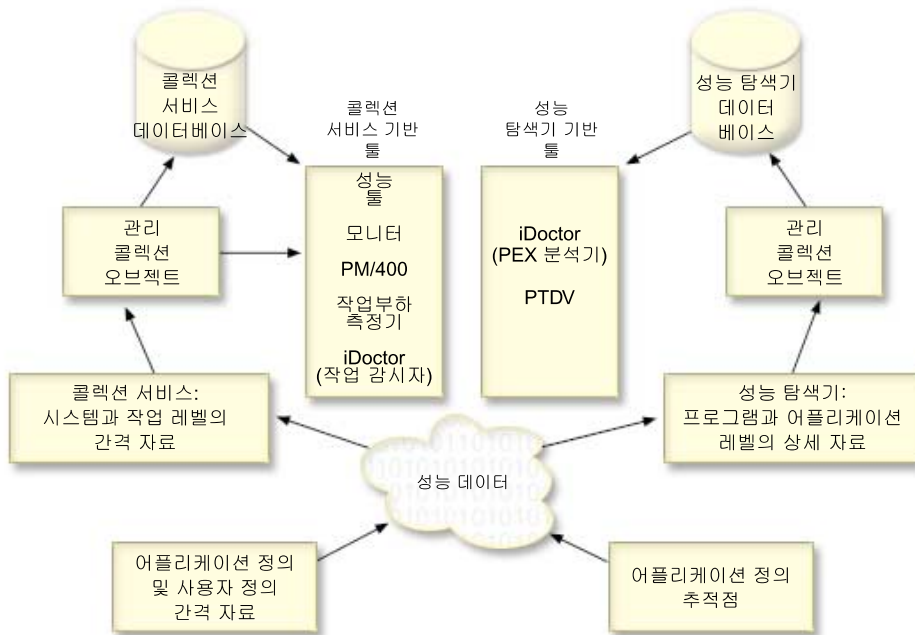
대다수의 성능 관리용 어플리케이션은 여러 가지 기능을 갖습니다. 사용할 수 있는 어플리케이션의 구성요소 중 주어진 상황에 가장 잘 맞는 구성요소를 정확하게 알아내는 것은 복잡할 수 있습니다. 다음 주제에서는 각 성능 관리 어플리케이션에 대한 자세한 정보(선택, 사용 및 구성)를 제공합니다.

다음 그림에 보인 것처럼 iSeries 서버에는 기본적으로 두 개의 성능 콜렉션 기능이 있습니다.

- 콜렉션 서비스 - 시스템과 작업 레벨에서 간격 자료를 수집합니다. 이 기능을 지속적으로 실행하면 시스템에서 발생하는 일을 알 수 있습니다. 수집되는 간격 자료는 어플리케이션 정의 자료이거나 사용자 정의 자료입니다.
- Performance Explorer - 프로그램 및 어플리케이션 레벨에서 자세한 자료를 수집합니다. 또한 어플리케이션에서 작업의 흐름을 추적하며 까다로운 성능 문제를 진단하는 데에도 사용할 수 있습니다. 수집되는 자료는 어플리케이션이 정의한 Performance Explorer 추적점(예: Domino, NetServer 또는 WebSphere)에 기준합니다.

이들 두 콜렉션 기능은 자료를 관리 콜렉션 오브젝트에 배치합니다. 콜렉션 서비스 자료의 경우에는 CRTPFRTA(Create Performance Data) 명령을 사용하고 Performance Explorer 자료의 경우에는 CRTPEXDTA(Create Performance Explorer Data) 명령을 사용하여 관리 콜렉션 오브젝트의 자료를 변환할 수 있습니다.

이 주제에서는 컬렉션 서비스 자료 또는 Performance Explorer 자료에 대해 작업할 때 사용할 수 있는 성능 관리 어플리케이션에 대해 소개합니다.



컬렉션 서비스

컬렉션 서비스는 사용자 정의 시간 간격으로 성능 자료를 수집한 다음 이 정보를 사용자 시스템의 컬렉션 오브젝트에 저장합니다. 모니터, 그래프 이력, PM/400 및 성능 분석 툴 사용권 프로그램의 여러 기능 등 다른 많은 툴은 이러한 컬렉션 서비스에 의존하여 자료를 얻습니다.

성능 데이터베이스 파일

컬렉션 오브젝트가 유지보수하는 컬렉션 오브젝트를 이용하여 데이터베이스 파일을 작성할 수 있습니다. 이 주제를 통해 데이터베이스 파일의 이름, 설명 및 속성을 찾아 보십시오.

모니터

모니터는 시스템의 성능에 대한 현재 정보를 표시합니다. 또한 모니터를 사용하여 특정 이벤트 발생 시 사전정의된 조치를 수행할 수도 있습니다. 시스템, 메시지, 작업, 파일 및 B2B 트랜잭션 모니터를 사용하여 사용자 시스템에 대한 정보를 표시하고 모니터할 수 있습니다. 시스템 및 작업 모니터는 컬렉션 서비스가 수집한 성능 자료를 사용합니다.

그래프 이력

그래프 이력은 지정된 기간 동안 컬렉션 서비스가 수집한 성능 자료를 그래픽 화면으로 제공합니다.

PM/400

PM/400은 시스템 성능 자료의 수집, 아카이브 및 분석을 자동화하며 시스템 자원과 용량 관리를 돕기 위해 보고서를 리턴합니다. PM/400은 컬렉션 서비스가 수집한 성능 자료를 사용합니다.

성능 분석 툴

성능 분석 툴 사용권 프로그램에는 시스템 성능 정보를 수집, 분석 및 유지보수할 수 있도록 도와주는 많은 기능이 들어 있습니다. 즉, 분산 네트워크상의 성능 관리, 요약 및 추적 자료의 수집 및 보고, 용량 계획 등에 대한 지원이 포함됩니다. 성능 분석 툴은 콜렉션 서비스가 수집한 성능 자료(샘플 자료)와, STRPFTRC(성능 추적 시작) 명령 및 ENDPFTRC(성능 추적 종료) 명령을 통해 얻은 추적 자료를 사용합니다.

Performance Explorer

Performance Explorer는 특정 어플리케이션, 프로그램 또는 시스템 자원에 대해 좀 더 자세한 정보를 수집하여 특정 성능 문제를 좀 더 잘 이해할 수 있도록 해줍니다. 여기에는 여러 가지 유형과 레벨의 추적을 수행하고 상세 보고서를 실행하기 위한 기능이 포함됩니다.

iDoctor for iSeries

iDoctor for iSeries 플러그 인은 성능 관리를 위한 세 개의 소프트웨어 툴로 구성되는데, 자세한 추적 자료 분석을 위한 Performance Explorer Analyzer, 작업의 작동에 대한 추적 레벨 정보를 제공하는 Job Watcher, 시스템 상의 오브젝트 조회 및 관리를 돕기 위한 Object Explorer가 그것입니다.

PTDV(Performance Trace Data Visualizer)

iSeries용 PTDV(Performance Trace Data Visualizer)는 iSeries에서 실행 중인 어플리케이션의 성능 분석에 사용할 수 있는 Java 어플리케이션입니다.

성능 관리 API

성능 관리 API는 콜렉션을 관리하기 위한 서비스를 제공합니다. 성능 관리 API는 콜렉션을 시작, 종료 및 순환하고 수집된 자료에 대한 시스템 매개변수를 변경 및 검색합니다. 대다수의 성능 관리 API는 콜렉션 서비스가 수집한 성능 자료를 사용합니다.

OS/400 성능 명령

OS/400에는 시스템 성능을 관리 및 유지보수하도록 돕는 여러 가지 중요한 기능이 포함되어 있습니다.

Extended Adaptive Cache

Extended Adaptive Cache는 디스크 사용 자료를 수집한 다음 이 통계를 사용하여 대형 캐시를 작성함으로써 디스크에 대한 실제 I/O 요구를 효과적으로 줄여 주어 시스템 성능을 개선할 수 있습니다.

iSeries용 작업부하 예측기

작업부하 예측기는 다음 업그레이드의 크기와 타이밍 요구사항을 계획하는 데 도움이 됩니다. 이 툴은 시스템 성능의 경향을 분석하기 위해 PM/400에 자주 사용되며 iSeries 서버의 성장과 확장을 효율적으로 관리할 수 있도록 해줍니다.

무선 iSeries Navigator

무선 iSeries Navigator를 이용하면 PDA, 인터넷폰 또는 전통적인 웹 브라우저를 사용하여 무선 연결을 통해 성능 자료를 모니터할 수 있습니다. 무선 iSeries Navigator는 콜렉션 서비스가 수집한 성능 자료를 사용합니다.

PATROL for iSeries(AS/400) - Predict

PATROL for iSeries(AS/400) - Predict는 고가용성 및 최적의 성능을 위해 필요한 다수의 루틴 관리 타스크를 자동화하여 iSeries 성능 관리를 도와줍니다. 또한 이 제품은 자세한 용량 계획 정보를 제공하여 사용자의 iSeries 환경에 대한 확장 계획을 도와줍니다.

컬렉션 서비스

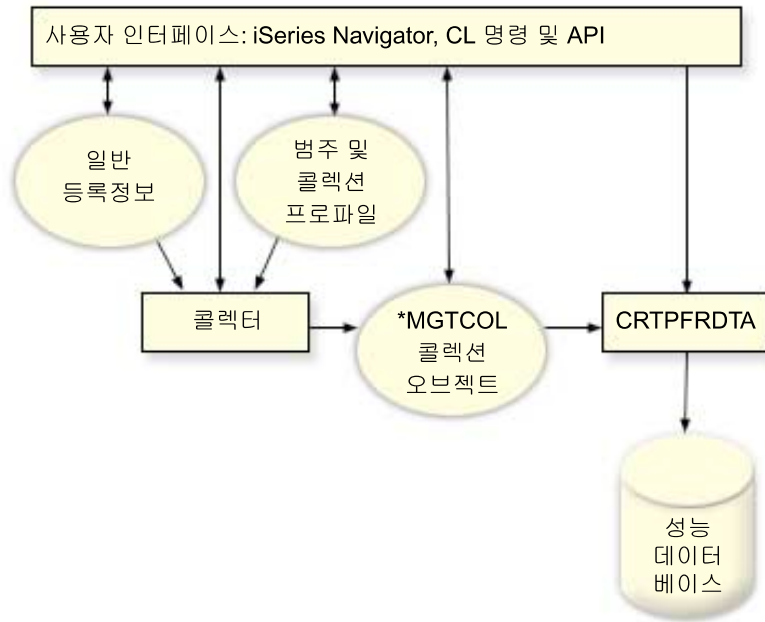
컬렉션 서비스를 사용하여 성능 자료를 수집한 후 차후 iSeries용 성능 분석 툴 사용권 프로그램이나 다른 성능 보고서 어플리케이션, iSeries Navigator 모니터 및 그래프 이력 기능 등을 사용한 분석에 이 자료를 사용할 수 있습니다. (실시간으로 성능 자료를 보고자 하는 경우 시스템 모니터는 시스템 성능을 모니터하도록 사용이 쉬운 그래픽 인터페이스를 제공합니다.) 컬렉션 서비스는 시스템의 서로 다른 영역들이 사용한 시스템 자원의 상대적인 양을 보여주는 자료를 수집합니다. 컬렉션 서비스를 사용하여 다음을 수행할 수 있습니다.

- 컬렉션 오브젝트를 쉽게 관리합니다.
- 최소의 시스템 오버헤드로 성능 자료를 지속적으로 자동 수집합니다.
- 어떤 자료를 수집하고 그 자료가 어떻게 사용되는지를 제어합니다.
- 자료를 변환하지 않고 릴리스간에 성능 자료를 이동합니다.
- 성능 분석 툴이 사용하는 성능 자료 파일을 작성합니다.
- 사용자 정의 성능 자료를 수집하기 위한 사용자 자신의 프로그램을 컬렉션 서비스로 통합합니다.

컬렉션 서비스의 작동 방식

컬렉션 서비스는 STRPFRMON(성능 모니터 시작) 명령으로 호출되던 OS/400 성능 모니터를 대체하는 것입니다. V4R5 이후에는 성능 모니터(STRPFRMON 명령)를 사용할 수 없습니다. OS/400 성능 모니터를 사용할 때는 자료가 30개의 데이터베이스 파일로 수집되었습니다.

컬렉션 서비스 기능에서는 성능 자료를 수집하는 새로운 프로세스가 도입되었습니다. 컬렉션 서비스는 각 컬렉션에 대한 자료를 하나의 컬렉션 오브젝트에 저장하고, 여기에서 필요한 수만큼의 데이터베이스 파일 세트를 작성할 수 있습니다. 이것은 성능 자료를 수집할 때 시스템 오버헤드가 낮아진다는 의미입니다. 수집 중에 데이터베이스 파일 작성을 선택하는 경우에도 컬렉션 서비스는 이들 파일을 갱신하는 데 낮은 우선순위(50)의 일괄처리 작업을 사용하기 때문에 OS/400 성능 모니터에 비해 더 우수한 성능을 경험할 수 있습니다. 컬렉션 오버헤드가 줄어들었으므로 성능 자료 수집을 훨씬 더 자세하게 그리고 지속적으로 훨씬 더 짧은 간격으로 하는 것이 가능해졌습니다. 컬렉션 서비스를 사용하여 성능 자료의 수집 및 보유에 대한 네트워크 전체적인 시스템 정책을 수립하고 그 정책을 자동으로 구현할 수 있습니다. 관리 컬렉션 오브젝트를 보유하고 있는 한, 필요가 발생하면 성능 관련 이벤트를 검토하여 사용자가 수집한 상세 레벨까지 분석할 수 있습니다.



컬렉션 서비스를 이용하면 시스템 성능에 미치는 영향은 거의 없거나 눈에 띄지 않게 하여 성능 자료를 수집할 수 있습니다. iSeries Navigator를 사용하여 사용자가 원하는 빈도대로 자료를 수집하도록 컬렉션 서비스를 구성할 수 있습니다. 컬렉션 오브젝트(*MGTCOL)는 다량의 성능 자료를 담기 위한 효율적인 저장 매체가 됩니다. 컬렉션 서비스를 구성하여 시작하면 지속적으로 성능 자료가 수집됩니다. 성능 자료에 대해 작업해야 하는 경우 필요한 자료를 성능 데이터베이스 파일 세트에 복사할 수 있습니다.

위의 그림은 아래의 컬렉션 서비스 요소에 대한 개요를 제공합니다.

사용자 인터페이스

컬렉션 서비스의 여러 요소에 액세스하기 위한 몇 가지 방법이 있습니다. 예를 들어 CL 명령, API 및 iSeries Navigator 인터페이스를 사용할 수 있습니다.

일반 등록 정보

일반 등록 정보는 컬렉션을 수행하는 방법을 정의하고 자동 컬렉션 속성을 제어합니다.

자료 범주

자료 범주는 수집할 자료의 유형을 식별합니다. 수집되는 자료와 자료 수집 빈도를 제어하기 위한 범주를 독립적으로 구성할 수 있습니다.

컬렉션 프로파일

컬렉션 프로파일은 특정의 범주 구성을 저장 및 활성화하기 위한 수단을 제공합니다.

성능 컬렉터

성능 컬렉터는 일반 등록 정보와 범주 정보를 사용하여 성능 자료의 수집을 제어합니다. 성능 컬렉터를 시작 및 중단할 수 있고 자동으로 실행되도록 구성할 수도 있습니다.

컬렉션 오브젝트

컬렉션 오브젝트(*MGTCOL)는 다량의 성능 자료를 담기 위한 효율적인 저장 매체가 됩니다.

CRTPRFDTA(성능 자료 작성) 명령


CRTPRFDTA 명령은 관리 컬렉션 오브젝트에 저장된 자료를 처리하고 성능 데이터베이스 파일을 생성합니다.

성능 데이터베이스

데이터베이스 파일은 CRTPRFDTA 명령이 처리한 자료를 저장합니다. 이 파일은 시간 간격 자료를 담은 성능 자료 파일, 구성 자료 파일 및 추적 자료 파일의 세 가지 범주로 나눌 수 있습니다.

컬렉션 서비스를 시작하는 방법

다음 방법 중 하나를 사용하여 컬렉션 서비스를 시작합니다. 그러나 성능 주제의 정보는 iSeries Navigator 매소드에 초점이 맞추어져 있습니다.

시작 방법	설명
iSeries Navigator	아래 섹션에서는 iSeries Navigator를 사용하여 다양한 컬렉션 서비스의 작업을 수행하는 방법을 설명합니다.
성능 관리 API	성능 관리 API를 사용하여 컬렉션을 시작, 사용자 정의, 종료 및 순환할 수 있습니다. 또한 API를 사용하여 관리 컬렉션 오브젝트에 대해 작업하거나 사용자 자신의 트랜잭션을 정의할 수 있습니다.
종래의 메뉴 옵션	문자 기반의 인터페이스에서 GO PERFORM 을 입력한 후 성능 분석 툴 기본 메뉴에서 옵션 2(성능 자료 수집)를 선택합니다. 추가 정보는 성능 분석 툴 책  을 참조하십시오.
Performance Management/400	컬렉션 서비스 시작을 자동화하고 컬렉션 중에 데이터베이스 파일을 작성하는 PM/400을 활성화할 수 있습니다.

컬렉션 서비스 태스크

컬렉션 서비스와 iSeries Navigator를 사용하여 아래 표에 보인 것과 같이 다양한 자료 수집 작업을 수행할 수 있습니다.

태스크	설명
다양한 방법으로 컬렉션 서비스를 시작합니다.	특정 성능 계량법을 사용하여 개별 시스템이나 시스템 그룹에서 사용자 정의된 성능 자료 컬렉션을 작성합니다. 또는 시작 프로그램에서 Start Collector API를 사용하여 성능 자료 수집을 자동으로 시작할 수 있습니다. 이들 작업을 수행하는 방법에 대한 자세한 정보는 온라인 도움말을 참조하십시오. 자세한 태스크 도움말은 iSeries Navigator 창에 제공됩니다. 메뉴 바에서 도움말을 클릭하고 도움말 주제를 선택하십시오. 수행할 수 있는 작업 . . . ?을 선택하여 사용자가 수행할 수 있는 작업과 이 작업을 수행하기 위한 iSeries Navigator에서의 위치를 찾으십시오.


타스크	설명
데이터베이스 파일 작성	컬렉션 서비스를 사용하여 성능 데이터베이스 파일 작성을 자동화합니다. 또한 자료가 수집된 후 저장되는 컬렉션 오브젝트로부터 데이터베이스 파일을 작성할 수도 있습니다. PM/400이나 성능 분석 툴 사용권 프로그램에서 이 데이터베이스 파일을 사용할 수 있으며 조회를 작성하여 이 파일에 대해 실행할 수도 있습니다. 데이터베이스 파일을 작성하면서 수집되는 자료를 제어하는 방법을 배우십시오. 사용자가 사용할 수 있는 데이터베이스 파일 및 각 파일에 들어 있는 필드 레벨 자료에 대한 내용은 성능 데이터베이스 파일을 참조하십시오.
자료 수집 사용자 정의	자료 수집 사용자 정의 어떤 성능 자료를 수집하고 얼마나 자주 수집할 것인지를 제어하는 방법에 대한 정보를 찾아 보십시오. 또한 중요한 시간대 고려사항에 대한 정보도 찾아 보십시오.
컬렉션 서비스에 사용자 정의 범주 추가	나감 프로그램을 작성한 후 이를 컬렉션 서비스에 통합하여 사용자 어플리케이션의 성능 자료를 수집할 수 있습니다. 루틴 컬렉션 간격 중에 이 자료를 수집하여 컬렉션 오브젝트에 이를 저장할 수 있습니다. 이를 구현하는 방법에 대한 정보는 사용자 정의 범주를 참조하십시오.
컬렉션 오브젝트 관리	컬렉션 오브젝트의 내용, 컬렉션 오브젝트의 저장 기간 및 컬렉션 오브젝트에 대하여 할 수 있는 작업 등 컬렉션 오브젝트를 관리하는 데 필요한 정보를 찾아 보십시오.
추적 자료 수집	컬렉션 서비스는 샘플 자료를 수집합니다. 그러나 추적 자료를 수집하지 않습니다. 추적 자료를 수집하는 방법을 찾아 보십시오.
사용자 정의 트랜잭션에 대한 성능 자료 수집	컬렉션 서비스는 사용자 자신의 트랜잭션을 정의할 수 있는 API를 제공합니다. 사용자 정의 트랜잭션에서 이 타스크를 수행하는 방법을 찾아 보십시오.

컬렉션 서비스 자료로 데이터베이스 파일 작성

컬렉션 서비스는 사용자가 수집한 자료를 관리 컬렉션 오브젝트에 넣습니다. 이 자료를 이용하려면 먼저 특별한 데이터베이스 파일 세트에 자료를 넣어야 합니다. 자료가 수집될 때 데이터베이스 파일을 자동으로 작성하려면 컬렉션 서비스 시작 대화상자에서 컬렉션 서비스 시작을 선택하십시오. 또한 기존의 관리 컬렉션 오브젝트에서 데이터베이스 파일로 자료를 내보내고자 할 때 데이터베이스 파일을 작성할 수도 있습니다.

데이터베이스 파일을 작성할 수 있는 옵션은 여러 가지가 있습니다.

- 컬렉션 서비스를 사용하여 성능 자료를 수집할 때 자동으로 데이터베이스 파일을 작성할 수 있습니다.
- 자료가 수집된 후 저장되는 컬렉션 오브젝트로부터 데이터베이스 파일을 작성할 수도 있습니다.
CRTPFDRDTA(성능 자료 작성) 명령을 사용하여 관리 컬렉션(*MGTCOL) 오브젝트에 저장된 성능 정보를 가지고 성능 데이터베이스 파일 세트를 작성할 수 있습니다. iSeries Navigator 또는 CRTPFDRDTA 명령을 사용할 수 있습니다.
- 컬렉션 서비스를 자동으로 시작하고 컬렉션 중에 데이터베이스 파일을 작성하는 Performance Management/400을 활성화할 수 있습니다.

사용자가 작성한 데이터베이스 파일을 iSeries용 성능 분석 툴 사용권 프로그램 또는 다른 어플리케이션과 함께 사용하여 성능 보고서를 생성할 수 있습니다. 성능 자료를 한 시스템에 수집한 후 관리 컬렉션 오브젝트(*MGTCOL)를 또다른 시스템으로 이동하여 성능 자료 파일을 생성하고 성능 분석 툴 보고서를 실행할 수 있습니다. 이러한 조치를 통해 소스 시스템의 성능에 영향을 주지 않으면서 다른 시스템의 자료를 분석할 수 있습니다. 성능 분석 툴에 대한 자세한 정보는 성능 분석 툴 책  을 참조하십시오.

데이터베이스 파일 대신 관리 컬렉션 오브젝트에 자료 저장

보고서 실행을 위해 필요로 하는 자료를 데이터베이스 파일 대신 관리 컬렉션 오브젝트에 저장해야 하는 이유는 무엇입니까? 관리 컬렉션 오브젝트는 데이터베이스 파일과 별도로 관리할 수 있으므로 짧은 컬렉션 간격(예: 5분 간격)으로 성능 자료를 수집한 다음 좀 더 긴 샘플링 간격(예: 15분 간격)으로 데이터베이스 파일을 작성할 수 있습니다.

하나의 관리 컬렉션 오브젝트에서, 자료 범주, 시간 범위 및 샘플링 간격을 각각 서로 다르게 지정하여 여러 가지 용도로 서로 다른 데이터베이스 파일 세트를 작성할 수 있습니다.

예를 들어 24시간 동안 컬렉션 간격을 5분으로 하여 전체 범주 (모든 자료 또는 **Standard** 플러스 프로토콜 프로파일)에 대해 성능 자료를 수집할 수도 있습니다. 이 관리 컬렉션 오브젝트에서 서로 다른 용도의 여러 가지 데이터베이스 파일 세트를 작성할 수 있습니다. 보통의 일일 성능 보고서를 실행하기 위해 데이터베이스 파일 세트를 작성할 수 있을 것입니다. 이들 파일에는 샘플링 간격을 15분으로 한 모든 범주의 자료가 들어 있을 것입니다. 따라서 특정한 성능 문제를 분석하려는 경우에는 또다른 데이터베이스 파일 세트를 작성할 수 있습니다. 이들 파일에는 분석하고자 하는 범주, 24시간 중 특정 시간 또는 좀 더 조밀한 샘플링 간격(5분)에 대한 자료가 들어 있을 것입니다.

또한 단일 관리 컬렉션 오브젝트를 이용하면 자료를 여러 개의 파일이 아니라 하나의 오브젝트로 관리할 수 있습니다. 단일 컬렉션 오브젝트를 이용하면 자료를 변환하지 않고도 릴리스들 간에서 성능 자료를 이동할 수 있습니다. 컬렉션 오브젝트를 보유하고 있는 한 성능 관련 이벤트를 검토하여 사용자가 수집한 상세 레벨까지 분석할 수 있습니다.

수집된 성능 자료 내보내기

관리 컬렉션 오브젝트에서 데이터베이스 파일로 성능 자료를 내보내려면 다음 단계를 수행하십시오.

1. iSeries Navigator에서 중앙 관리 아래의 종료점 시스템을 선택하거나 사용자 연결 (또는 사용자의 활동 환경) 아래에서 직접 연결되어 있는 시스템을 선택하십시오.
2. 구성 및 서비스를 확장하십시오.
3. 컬렉션 서비스를 클릭하십시오.
4. 데이터베이스 파일로 내보내려는 관리 컬렉션 오브젝트를 마우스 오른쪽 버튼으로 클릭하고 데이터베이스 파일 작성을 선택하십시오.
5. 데이터베이스 파일 작성 대화상자에서 데이터베이스 파일에 포함시키려는 컬렉션 오브젝트의 범주를 선택하십시오. 또한 컬렉션 오브젝트에 사용자의 선택을 지원할 자료가 들어 있다면 다른 시간대와 샘플링 간격을 선택할 수도 있습니다.
6. 확인을 클릭하십시오.

기존 컬렉션 오브젝트로 데이터베이스 파일 작성: 성능 자료를 기존의 관리 컬렉션 오브젝트에서 데이터베이스 파일로 내보낼 수 있습니다. 다음 단계를 수행하십시오.

1. 성능 자료를 수집하고 있는 시스템에 대해 구성 및 서비스를 확장하십시오.
2. 컬렉션 서비스를 선택하십시오.

3. 데이터베이스 파일로 내보내려는 자료가 들어 있는 관리 콜렉션 오브젝트를 마우스 오른쪽 버튼으로 클릭하십시오.
4. 먼저 등록 정보를 선택하여 콜렉션 오브젝트에 있는 자료의 특성을 표시할 수 있습니다. 자료 등록 정보 페이지에는 이 콜렉션 오브젝트에서 수집된 자료의 범주와, 이들 자료의 수집 간격이 표시됩니다. 내보내려는 자료를 선택할 때 이 자료를 이용할 수 있습니다. 이 정보를 모두 검토하였으면 확인을 클릭하십시오.
5. 다시 관리 콜렉션 오브젝트를 마우스 오른쪽 버튼으로 클릭하고 데이터베이스 파일 작성을 선택하십시오. 온라인 도움말을 이용하여 필드를 완료하십시오.
6. 확인을 클릭하십시오.

데이터베이스 파일의 자료를 변환한 후, iSeries 사용권 프로그램용 성능 분석 툴 또는 다른 어플리케이션을 사용하여 성능 보고서를 생성할 수 있습니다.

자료 콜렉션 사용자 정의

콜렉션 서비스를 사용하여 성능 자료를 콜렉션하는 경우 수집되는 자료와 수집 간격을 제어할 수 있습니다. 즉, 제공되는 콜렉션 프로파일에서 선택할 수 있습니다. 표준 프로파일은 이전 릴리스의 STRPFRMON (성능 모니터 시작) 명령이 제공하였던 OS/400 성능 모니터 기능에서 시스템 자료에 대한 설정에 해당합니다. **Standard 플러스 프로토콜** 프로파일은 모든 자료에 대한 STRPFRMON 명령 설정에 해당합니다. 또는 사용자 정의를 선택하여 사용자 자신이 정의한 프로파일을 작성할 수도 있습니다. 사용할 수 있는 여러 가지 다른 프로파일이 있으므로 자세한 설명은 온라인 도움말을 참조하십시오. 사용자 정의 프로파일의 경우, 사용할 수 있는 자료 범주(예: 시스템 CPU, 로컬 응답 시간, 디스크 기억장치 및 입/출력 프로세서) 리스트에서 선택할 수 있습니다.

콜렉션하는 각 자료 범주에 대해 자료 콜렉션 간격을 지정할 수 있습니다. 대다수의 범주에 대해 15초에서 60분 사이로 사전 정의된 설정 중에서 선택할 수 있는 디폴트 콜렉션 간격을 선택하는 것이 좋을 것입니다. (권장되는 설정은 15분입니다.)

주: 디폴트 값이 지정된 시간으로 설정되면 명시적 시간 간격을 갖는 범주(예: 디스크 기억장치, 입/출력 프로세서 및 통신 관련 범주)를 제외한 모든 범주는 지정된 시간을 사용합니다.

콜렉션된 자료는 콜렉션이라고 하는 관리 콜렉션 오브젝트(유형 *MGTCOL)에 저장됩니다. 이들 관리 콜렉션 오브젝트가 너무 커지는 것을 방지하려면 일정한 간격으로 콜렉션을 순환해야 합니다. 콜렉션 순환이란 새로운 콜렉션 오브젝트를 작성하여 여기에 자료 저장을 시작하고 이와 동시에 원래 콜렉션 오브젝트에서는 자료 콜렉션이 중단되는 것을 의미합니다. 자료 사용 계획에 따라 1시간에서 24시간 사이의 간격을 지정할 수 있습니다.

시스템에서 콜렉션 서비스를 사용자 정의하려면 다음 단계를 수행하십시오.

1. iSeries Navigator에서 중앙 관리 아래의 종료점 시스템을 선택하거나 사용자 연결 (또는 사용자의 활동 환경) 아래에서 직접 연결되어 있는 시스템을 선택하십시오.
2. 구성 및 서비스를 확장하십시오.
3. 콜렉션 서비스를 마우스 오른쪽 버튼으로 클릭하고 등록 정보를 선택하십시오.

4. 일반 페이지에서 디폴트값(1일)보다 긴 보유 기간을 지정해야 합니다. 콜렉션 서비스는 보유 기간이 만기된 후에는 언제라도 관리 콜렉션 오브젝트와 이의 자료를 시스템에서 삭제할 수 있습니다. 관리 콜렉션 오브젝트가 작성되면 여기에 만기일이 지정됩니다. 콜렉션 오브젝트를 다른 라이브러리로 이동한 경우에도 콜렉션 서비스는 오브젝트가 만기된 후에는 이를 삭제합니다. 콜렉션 서비스가 새로운 콜렉션 오브젝트에 만기일을 지정하지 않게 하려면 영구를 지정하십시오. 이 경우에는 콜렉션 오브젝트를 수동으로 삭제해야 합니다.
 그래프 이력 창을 보려면, 그래프 또는 요약의 콜렉션 보유 기간을 지정해야 합니다. 이들 옵션을 지정하는 경우에는 이력 보고 기능을 활용할 수 있는데 이 기능을 이용하면 장시간에 걸쳐 여러 시스템에 대한 수치 비교를 수행할 수 있습니다.
 또한 콜렉션을 저장하려는 위치의 경로, 콜렉션 순환 빈도 및 디폴트 콜렉션 간격을 지정할 수도 있습니다. 콜렉션 중 자동으로 데이터베이스 파일을 작성하도록 선택할 수 있습니다.
5. 수집할 자료 탭을 클릭하십시오.
6. 사용할 콜렉션 프로파일에 대해서는 사용자 정의를 선택하십시오. 사용자 정의된 리스트에 대해 선택하는 각 범주에 대해 콜렉션 간격을 지정할 수 있습니다.
7. 확인을 클릭하여 사용자 정의된 값을 저장하십시오.

콜렉션 서비스를 원하는 설정으로 사용자 정의한 후 콜렉션 서비스를 마우스 오른쪽 버튼으로 클릭하고 콜렉션 서비스 시작을 선택하여 성능 자료 수집을 시작할 수 있습니다.

콜렉션 서비스의 시간대 고려사항

성능 자료를 검토하고 분석할 경우 콜렉션의 실제 현지 시간은 중요할 수 있습니다. 예를 들어 검토 시 최대 작업부하가 시스템에 부과된 것으로 나타나는 하루 중 가장 바쁜 시간 기간에 어떤 자료가 수집되었는지를 확인해야 할 수 있습니다. 성능 자료를 수집하는 시스템 중 일부가 다른 시간대에 위치할 경우 다음 사항에 대해 알아야 합니다.

- 시스템 그룹에 대해 콜렉션 서비스를 시작할 때 그룹 내의 모든 시스템에서 동시에 콜렉션 서비스를 시작하십시오. 일부 시스템이 다른 시간대에 위치하기 때문에 발생하는 시스템 시간과 날짜 설정의 차이는 고려되지 않습니다.
- 중앙 관리 스케줄러를 사용하여 콜렉션 서비스를 시작하는 경우 스케줄러가 타스크를 시작하는 시간은 중앙 관리에서 중앙 시스템의 시스템 시간과 날짜에 기반합니다.
- 각 종료점 시스템의 관리 콜렉션 오브젝트는 해당 종료점 시스템과 중앙 시스템의 QTIME 및 QUTCOFFSET(협정 세계시 오프셋) 시스템 값에 기초한 시작 및 종료 시간을 반영합니다. 종료점 시스템이 중앙 시스템과 다른 시간대에 있고 이들 시스템 값이 두 시스템에서 정확하게 설정된 경우 콜렉션 오브젝트에 대해 보고되는 시작 및 종료 시간은 종료점 시스템의 실제 시간입니다. 다시 말해 시작 및 종료 시간은 이벤트가 발생할 당시 종료점 시스템의 QTIME 값을 반영합니다.
- 성능 콜렉션의 스케줄링은 표준 시간제에서 일광 절약 시간제 또는 일광 절약 시간제에서 표준 시간제로 경계를 교차할 수 있습니다. 그럴 경우 시작 시간을 스케줄링할 때 이러한 시간 차이를 고려해야 합니다. 그렇지 않으면 실제 시작 및 종료 시간이 예상보다 1시간 늦거나 빠를 수 있습니다. 또한 일광 절약 시간제와 표준 시간제 사이에서 변경이 발생할 때마다 QUTCOFFSET 시스템 값을 조정하지 않으면 관리 콜렉션 오브젝트에 대해 보고되는 시작 및 종료 시간은 이러한 차이에 따른 영향을 받습니다.

컬렉션 서비스를 사용하여 성능 자료를 수집하는 데 대한 자세한 내용은 컬렉션 서비스를 참조하십시오.

컬렉션 서비스의 사용자 정의 범주

컬렉션 서비스의 사용자 정의 범주 기능을 통해 어플리케이션은 성능 자료 컬렉션을 컬렉션 서비스로 통합할 수 있습니다. 이로써 사용자는 자료 컬렉션 프로그램을 작성하고 이를 컬렉션 서비스에 등록 및 통합하여 어플리케이션의 자료를 수집할 수 있습니다. 컬렉션 서비스는 모든 컬렉션 간격에서 자료 컬렉션 프로그램을 호출하여 자료를 컬렉션 오브젝트에 저장합니다. 컬렉션 오브젝트에 저장된 자료에 액세스하려면 아래 나열된 컬렉션 오브젝트 API를 사용해야 합니다. 자료가 수집되는 중이나 컬렉션 오브젝트가 보유하고 있는 동안 자료에 실시간으로 액세스할 수도 있습니다.

이 기능을 구현하려면 다음이 필요합니다.

1. 컬렉션 서비스에서 새 범주에 대한 성능 자료를 수집하기 위한 프로그램을 개발합니다. 자세한 정보는 컬렉션 프로그램 권장사항 및 요구사항을 참조하십시오.
2. 컬렉션 프로그램에 대한 작업 설명을 작성합니다. QGPL의 작업 설명(QPMUSRCAT)은 예를 제공하기는 하지만 디폴트 값이나 권장값을 표시하지는 않습니다.
3. 새 범주를 등록하고 자료 컬렉션 프로그램을 지정합니다. 자세한 정보는 API 설명을 참조하십시오.
 - QypsRegCollectorDataCategory를 등록합니다.
 - QypsDeregCollectorDataCategory를 등록 해제합니다.범주를 등록하면 컬렉션 서비스는 이 범주를 사용 가능한 컬렉션 범주 리스트에 포함시킵니다.
4. 범주를 컬렉션 서비스 프로파일에 추가한 후 컬렉션 서비스를 순환합니다.
5. 컬렉션 오브젝트를 조회하기 위한 프로그램을 개발합니다. 자세한 정보는 API 설명을 참조하십시오.
 - 활동 관리 컬렉션 오브젝트명을 검색합니다(QpmRtvActiveMgtcolName(컬렉션 오브젝트를 실시간으로 조회하는 경우에만 사용됨)).
 - 관리 컬렉션 오브젝트 속성을 검색합니다(QpmRtvMgtcolAttrs).
 - 관리 컬렉션 오브젝트를 엽니다(QpmOpenMgtcol).
 - 관리 컬렉션 오브젝트를 닫습니다(QpmCloseMgtcol).
 - 관리 컬렉션 오브젝트 저장소를 엽니다(QpmOpenMgtcolRepo).
 - 관리 컬렉션 오브젝트 저장소를 닫습니다(QpmCloseMgtcolRepo).
 - 관리 컬렉션 오브젝트 자료를 읽습니다(QpmReadMgtcolData).

사용자 정의된 컬렉션 프로그램은 이제 각 컬렉션 간격마다 실행되며 수집되는 자료는 컬렉션 오브젝트에 아카이브됩니다.

이들 API의 Java 버전을 구현할 수도 있습니다. 필요한 Java 클래스는 IFS 디렉토리 QIBM/ProdData/OS400/CollectionServices/lib에서 ColSrv.jar에 포함되어 있습니다. Java 어플리케이션은 자신의 클래스 경로에 이 파일을 포함하고 있어야 합니다. Java 구현에 대한 자세한 정보는 javadocs를 참조하십시오.

구현 예를 보려면, 예: 사용자 정의 범주 구현을 참조하십시오.

실시간으로 콜렉션 오브젝트 조회

어플리케이션이 콜렉션 오브젝트를 실시간으로 조회해야 하는 경우 조회를 콜렉션 서비스와 동기화할 필요가 있습니다. 이를 수행하려면 어플리케이션은 자료 대기행렬을 작성하여 이를 콜렉션 서비스에 등록해야 합니다. 등록되면 콜렉터는 각 콜렉션 간격과 콜렉션 주기의 끝에서 통지를 송신합니다. 어플리케이션은 완료 시 자료 대기행렬을 제거하고 비정상 종료를 처리하는 등 자료 대기행렬을 유지보수해야 합니다. 자료 대기행렬을 등록 및 등록 취소하려면 다음의 API 설명을 참조하십시오.

- 콜렉터 통지 추가: QypsAddCollectorNotification
- 콜렉터 통지 제거: QypsRmvCollectorNotification

콜렉션 프로그램 권장사항 및 요구사항: 콜렉션 서비스는 콜렉션 주기의 시작 시, 각 콜렉션 간격마다, 그리고 콜렉션 주기의 종료 시에 한 번씩 자료 콜렉션 프로그램을 호출합니다. 자료 콜렉션 프로그램은 임의의 자료 콜렉션을 수행하여 콜렉션 서비스가 제공하는 자료 버퍼에 해당 자료를 리턴해야 합니다. 자료 버퍼를 제공하는 외에 콜렉션 서비스는 작업 영역도 제공하는데 이를 통해 자료 콜렉션 프로그램은 콜렉션 간격 사이의 일부 상태 정보를 유지보수할 수 있습니다.

자료 콜렉션 프로그램은 가능한 빨리 자료를 수집해야 하며 최소한의 포매팅을 수행해야 합니다. 이 프로그램은 자료 처리 또는 정렬을 수행하면 안됩니다. 사용자 정의 범주의 자료가 데이터베이스 파일로 변환되지 않아도 콜렉션 서비스는 CRTPFRTA 명령을 자동으로 실행하여 각 콜렉션 간격의 종료 시 콜렉션 오브젝트의 자료를 데이터베이스 파일에 추가합니다. 자료 콜렉션 프로그램이 콜렉션 간격 내에 타스크를 완료할 수 없는 경우 CRTPFRTA 명령이 올바르게 실행되지 않습니다.

여러 환경에서 자료 콜렉션 프로그램을 작성할 수 있습니다.

- *PGM for OPM 언어. 이 환경은 콜렉션 오브젝트를 조회하는 데에는 사용할 수 없으며 성능 저하를 초래할 수도 있습니다. 그러나 이전의 프로그래밍 언어에는 사용할 수 있습니다.
- *SRVPGM - 서비스 프로그램에서의 입력점. 이것은 ILE 언어용입니다.
- *JVAPGM - 필요한 Java 클래스는 ColSrv.jar에 포함되어 있습니다. 이 파일은 QIBM/ProdData/OS400/CollectionServices/lib에서 IFS에 들어 있습니다. javadocs .zip 파일을 다운로드하여 index.html을 열고 API의 Java 구현에 대한 정보를 참조하십시오.

콜렉션 서비스는 다음 요구를 자료 콜렉션 프로그램에 송신합니다.

요구	설명
콜렉션 시작	자료 콜렉션 프로그램은 자료 콜렉션 중 사용되는 인터페이스나 자원을 초기화해야 합니다. 선택적으로 콜렉션 간격 사이의 상태 정보를 보관하는 작업 영역(콜렉션 서비스가 제공함)을 초기화하기도 합니다. 수집된 자료 이전에 제어 레코드를 포함시키려는 경우 자료 콜렉션 프로그램이 소량의 자료를 자료 버퍼에 기록할 수도 있습니다. 일반적으로 이 제어 레코드는 자료 해석을 돕기 위해 자료 처리 중에 사용됩니다.

요구	설명
컬렉션 간격	컬렉션 서비스는 각 컬렉션 간격에 대해 간격 요구를 송신합니다. 자료 컬렉션 프로그램은 자료를 수집하여 이를 자료 버퍼에 리턴해야 합니다. 그러면 컬렉션 서비스는 이 자료를 컬렉션 오브젝트의 간격 레코드에 기록합니다. 자료의 양이 자료 버퍼에 비해 너무 많을 경우, 자료 컬렉션 프로그램은 “추가 자료” 플래그를 설정합니다. 이 조치로 컬렉션 서비스는 이 자료가 연속 자료임을 나타내는 수정자와 함께 또다른 간격 요구를 송신하게 됩니다. 컬렉션 서비스는 각 호출 이전에 추가 자료 플래그를 재설정합니다. 모든 자료가 컬렉션 오브젝트로 이동될 때까지 이 프로세스는 반복됩니다.
컬렉션의 끝	자료 컬렉션 프로그램이 들어 있는 범주에 대한 컬렉션이 종료되면 컬렉션 서비스는 이 요구를 송신합니다. 자료 컬렉션 프로그램은 클린업을 수행해야 하고 선택적으로 컬렉션 제어 레코드를 리턴할 수 있습니다. 또한 자료 컬렉션 프로그램은 컬렉션의 결과를 나타내는 리턴 코드도 송신해야 합니다.
클린업 및 종료(시스템 종료)	컬렉션 서비스는 비정상 종료가 필요한 경우 이 요구를 송신합니다. 자료 컬렉션 프로그램 작업이 종료되면 오퍼레이팅 시스템 자원은 자동으로 비워지지만 다른 시스템 종료 조작은 자료 컬렉션 프로그램이 수행해야 합니다. 자료 컬렉션 프로그램은 이 요구를 언제라도 수신할 수 있습니다.

이들 매개변수, 작업 영역, 자료 버퍼 및 리턴 코드에 대한 자세한 설명은 QSYSINC에 들어 있는 헤더 파일 QPMDCPRM을 참조하십시오.

컬렉션 오브젝트의 자료 기억장치

컬렉션 오브젝트는 각 자료 컬렉션 범주에 대한 저장소를 가지고 있습니다. 이 저장소는 해당 범주에 대한 컬렉션이 시작될 때 컬렉션 서비스가 작성합니다. 각 저장소는 다음과 같은 레코드로 구성됩니다.

레코드	설명
제어	이 선택적 레코드는 자료 컬렉션 프로그램을 통해 얻어지는 첫 번째 또는 마지막 레코드가 될 수 있으며 두 위치 모두에서 발생할 수 있습니다. 일반적으로 이 레코드에는 레코드 자료를 해석하는 데 필요한 정보가 들어 있습니다.
간격	각 컬렉션 간격은 비어 있더라도 간격 레코드를 작성합니다. 간격 레코드에는 컬렉션 간격 중 자료 버퍼에 기록된 자료가 들어 있습니다. 레코드의 크기는 4GB를 초과할 수 없습니다.
중단	컬렉션 서비스는 자료 컬렉션 세션의 끝을 나타내기 위해 이 레코드를 자동으로 작성합니다. 컬렉션 서비스를 종료하거나 순환하지 않고 사용자 정의 범주에 대한 컬렉션이 다시 시작된 경우 중단 레코드 이후에 제어 레코드와 추가의 간격 레코드를 차례로 포함시킬 수도 있습니다.

예: 사용자 정의 범주 구현: 다음 샘플 프로그램은 제공되는 API를 사용하여 사용자 정의된 자료 컬렉션을 컬렉션 서비스로 통합하는 방법을 보여줍니다.

- 샘플 자료 컬렉션 프로그램(C++)
- 자료 컬렉션 프로그램을 등록하기 위한 샘플 프로그램(C++)
- 컬렉션 오브젝트를 조회하기 위한 샘플 프로그램(Java)

코드 면책사항 정보

IBM은 귀하에게 유사한 기능을 귀하의 특정 요구에 맞게 조정하여 생성할 수 있도록 모든 프로그래밍 코드 예제를 사용할 수 있는 비독점적인 저작권 사용권을 부여합니다.

모든 샘플 예제는 IBM에 의해 예시 목적으로만 제공됩니다. 이러한 예제는 모든 조건하에서 철저히 테스트된 것은 아닙니다. 따라서 IBM은 이들 프로그램의 신뢰성, 실용성 또는 기능에 대해 보증할 수 없습니다.

여기에 포함된 모든 프로그램은 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 어떠한 종류의 보증 없이 "현상태대로" 제공됩니다.

예: 자료 콜렉션 프로그램: 다음 프로그램은 약간의 테스트 자료를 수집하여 이를 자료 버퍼에 저장하며 콜렉션 서비스는 이를 콜렉션 오브젝트로 복사합니다. 콜렉션 프로그램에 대한 자세한 정보는 콜렉션 프로그램 권장사항 및 요구사항을 참조하십시오.

주: 중요한 법적 고지사항에 관해 언급하는 코드 면책사항 관련 정보를 참조하십시오

C++ 샘플 코드

```
#include "string.h"           // memcpy(), memset(), strlen()
#include "stdio.h"           // printf()
#include "qpmcprm.h"         // data collection program interface
#include "time.h"

extern "C"
void DCPentry( Qpm_DC_Parm_t *request, char *dataBuffer,
               char *workArea, int *returnCode )
{
    static char testData[21] = "Just some test stuff";
    int i;

    /* Print contents of request structure */

    printf( "DCP called with parameters:\n" );
    printf( "  format name: \"%8.8s\"; category name: \"%10.10s\";\n",
            request->formatName, request->categoryName );
    printf( "  rsvd1: %4.4X; req type: %d; req mod: %d; buffer len: %d;\n",
            *(short*)(request->rsvd1), request->requestType,
            request->requestModifier, request->dataBufferLength );
    printf( "  prm offset: %d; prm len: %d; work len: %d; rsvd2: %8.8X;\n",
            request->parmOffset, request->parmLength, request->workAreaLength,
            *(int*)(request->rsvd2) );
    printf( "  rec key: \"%8.8s\"; timestamp: %8.8X %8.8X;\n",
            request->intervalKey,
            *(int*)(request->intervalTimestamp),
            *(int*)(request->intervalTimestamp + 4) );
    printf( "  return len: %d; more data: %d; rsvd3: %8.8X %8.8X;\n",
            request->bytesProvided, request->moreData,
            *(int*)(request->rsvd3),
            *(int*)(request->rsvd3 + 4) );

    switch( request->requestType )
    {
        /* Write control record in the beginning of collection */
        case PM_DOBEGIN:
            printf( "doBegin(%d)\n", request->requestModifier );
            switch( request->requestModifier )
```

```

{
    case PM_CALL_NORMAL:
        memcpy( dataBuffer, testData, 20 );
        *(int *)workArea = 20;
        request->moreData = PM_MORE_DATA;
        request->bytesProvided = 20;
    break;

    case PM_CALL_CONTINUE:
        if( *(int *)workArea < 200 )
        {
            memcpy( dataBuffer, testData, 20 );
            *(int *)workArea += 20;
            request->moreData = PM_MORE_DATA;
            request->bytesProvided = 20;
        }
    else
        {
            *(int *)workArea = 0;
            request->moreData = PM_NO_MORE_DATA;
            request->bytesProvided = 0;
        }
    break;

default:
    *returnCode = -1;
    return;
}
break;

/* Write control record in the end of collection */
case PM_DOEND:
    printf( "doEnd(%d)\n", request->requestModifier );
    switch( request->requestModifier )
    {
        case PM_CALL_NORMAL:
            memcpy( dataBuffer, testData, 20 );
            *(int *)workArea = 20;
            request->moreData = PM_MORE_DATA;
            request->bytesProvided = 20;
        break;

        case PM_CALL_CONTINUE:
            if( *(int *)workArea < 200 )
            {
                memcpy( dataBuffer, testData, 20 );
                *(int *)workArea += 20;
                request->moreData = PM_MORE_DATA;
                request->bytesProvided = 20;
            }
        else
            {
                *(int *)workArea = 0;
                request->moreData = PM_NO_MORE_DATA;
                request->bytesProvided = 0;
            }
        break;

default:

```

```

        *returnCode = -1;
        return;
    }
    break;

/*Write interval record */
case PM_DOCOLLECT:
    printf( "doCollect(%d)\n", request->requestModifier );
    for( i = 0; i < 10000; i++ )
        dataBuffer[i] = i % 256;
    request->bytesProvided = 10000;

    switch( request->requestModifier)
    {
        case PM_CALL_NORMAL:
            *(time_t *)(workArea + 4) = time( NULL );
            *(int *)workArea = 1;
            request->moreData = PM_MORE_DATA;
            break;

        case PM_CALL_CONTINUE:
            *(int *)workArea += 1;
            if( *(int *)workArea < 20 )
                request->moreData = PM_MORE_DATA;
            else
            {
                *(time_t *)(workArea + 8) = time( NULL );
                printf( "doCollect() complete in %d secs(%d bytes transferred)\n",
                    *(time_t *)(workArea + 8) - *(time_t *)(workArea + 4), 10000 * 20 );
                request->moreData = PM_NO_MORE_DATA;
            }
            break;

        default:
            *returnCode = -1;
            return;
    }
    break;

/* Clean-up and terminate */
case PM_DOSHUTDOWN:
    printf( "doShutdown\n" );
    *returnCode = 0;
    return;
    break;

default:
    *returnCode = -1;
    return;
    break;
}

}/* DCPentry() */

```

예: 자료 콜렉션 프로그램을 등록하기 위한 프로그램: 다음 프로그램은 이전 예의 자료 콜렉션 프로그램을 콜렉션 서비스에 등록합니다. 실행 후 콜렉션 서비스는 자료 콜렉션 프로그램을 자료 콜렉션 범주의 리스트에 표시합니다.

C++ 샘플 코드

```
#include "stdlib.h"
#include "stdio.h"
#include "string.h"
#include "qypscoll.cleinc"

int main(int argc, char *argv[])
{
    int    CCSID = 0;
    int    RC = 0;
    Qyps_USER_CAT_PROGRAM_ATTR    *pgmAttr;
    Qyps_USER_CAT_ATTR            catAttr;
    char  collectorName[11] = "*PFR    ";
    char  categoryName[11] = "TESTCAT ";
    char  collectorDefn[11] = "*CUSTOM "; /* Register to *CUSTOM profile only */

    if( argc > 2 )
    {
        int len = strlen( argv[2] );

        if( len > 10 ) len = 10;
        memset( categoryName, ' ', 10 );
        memcpy( categoryName, argv[2], len );
    }

    if( argc < 2 || *argv[1] == 'R' )
    {
        pgmAttr = (Qyps_USER_CAT_PROGRAM_ATTR *)malloc( 4096 );
        memset( pgmAttr, 0x00, sizeof(pgmAttr) );
        pgmAttr->fixedPortionSize = sizeof( Qyps_USER_CAT_PROGRAM_ATTR );
        memcpy( pgmAttr->programType,    "*SRVPGM    ", 10 );
        memcpy( pgmAttr->parameterFormat, "PMDC0100", 8 );
        memcpy( pgmAttr->ownerUserId,    "USERID    ", 10 );
        memcpy( pgmAttr->jobDescription, "QPMUSRCAT QGPL    ", 20 );
        memcpy( pgmAttr->qualPgmSrvpgmName, "DCPTEST  LIBRARY    ", 20 );
        pgmAttr->workAreaSize = 123;
        pgmAttr->srvpgmEntrypointOffset = pgmAttr->fixedPortionSize;
        pgmAttr->srvpgmEntrypointLength = 8;
        pgmAttr->categoryParameterOffset = pgmAttr->srvpgmEntrypointOffset +
                                           pgmAttr->srvpgmEntrypointLength;
        pgmAttr->categoryParameterLength = 10;
        /* Set entry point name */
        memcpy( (char *) (pgmAttr) + pgmAttr->srvpgmEntrypointOffset,
               "DCPentry", pgmAttr->srvpgmEntrypointLength ); /* Set parameter string */
        memcpy( (char *) (pgmAttr) + pgmAttr->categoryParameterOffset,
               "1234567890", pgmAttr->categoryParameterLength );

        memset( &catAttr, 0x00, sizeof(catAttr) );
        catAttr.structureSize = sizeof( Qyps_USER_CAT_ATTR );
        catAttr.minCollectionInterval = 0;
        catAttr.maxCollectionInterval = 0;
        catAttr.defaultCollectionInterval = 30; /* Collect at 30 second interval */
        memset( catAttr.qualifiedMsgId, ' ', sizeof(catAttr.qualifiedMsgId) );
        memcpy( catAttr.categoryDesc,
               "12345678901234567890123456789012345678901234567890", sizeof(catAttr.categoryDesc) );

        QypsRegCollectorDataCategory( collectorName,
                                     categoryName,
                                     collectorDefn,
                                     &CCSID,
                                     (char*)pgmAttr,
                                     (char*)&catAttr,
    }
```



```

                                &RC
                                );
    }
    else
        if( argc >= 2 && *argv[1] == 'D' )
            QypsDeregCollectorDataCategory( collectorName, categoryName, &RC );
    else
        printf("Unrecognized option\n");
}/* main() */

```

예: 컬렉션 오브젝트를 조회하기 위한 프로그램: 다음의 샘플 프로그램은 QIBM/ProdData/OS400/CollectionServices/lib의 ColSrv.jar 파일에 들어 있는 Java 클래스를 사용하여 컬렉션 오브젝트에 저장된 자료를 조회하는 방법을 보여줍니다.

Java 샘플 코드

```

import com.ibm.iseries.collectionservices.*;

class testmco2
{
    public static void main( String argv[] )
    {
        String    objectName = null;
        String    libraryName = null;
        String    repoName = null;
        MgtcolObj mco = null;
        int       repoHandle = 0;
        int       argc = argv.length;
        MgtcolObjAttributes
            attr = null;
        MgtcolObjRepositoryEntry
            repoE = null;
        MgtcolObjCollectionEntry
            collE = null;
        int       i,j;

        if( argc < 3 )
        {
            System.out.println("testmco2 objectName libraryName repoName");
            System.exit(1);
        }

        objectName = argv[0];
        libraryName = argv[1];
        repoName = argv[2];

        if( ! objectName.equals( "*ACTIVE" ) )
            mco = new MgtcolObj( objectName, libraryName );
        else
            try
            {
                mco = MgtcolObj.rtvActive();
            } catch( Exception e)
            {
                System.out.println("rtvActive(): Exception " + e );
                System.exit(1);
            }
        System.out.println("Object name = " + mco.getName() );
        System.out.println("Library name = " + mco.getLibrary() );

        try

```

```

{
    attr = mco.rtvAttributes( "MCOA0100" );
} catch( Exception e)
{
    System.out.println("rtvAttributes(): MCOA0100: Exception " + e );
    System.exit(1);
}

System.out.println("MCOA0100: Object " + mco.getLibrary() + "/" + mco.getName() );
System.out.println("    size = " + attr.size + " retention = " + attr.retentionPeriod +
    " interval = " + attr.dftInterval + " time created = " + attr.timeCreated +
    " time updated = " + attr.timeUpdated );
System.out.println("    serial = " + attr.logicalPSN + " active = " + attr.isActive +
    " repaired = " + attr.isRepaired + " summary = " + attr.sumStatus +
    " repo count = " + attr.repositoryCount );
if( attr.repositoryInfo != null )
    for(i = 0; i < attr.repositoryCount; i++ )
    {
repoE = attr.repositoryInfo[ i ];
System.out.println("        name = " + repoE.name + " category = " + repoE.categoryName +
    " size = " + repoE.size );
for( j = 0; j < repoE.collectionInfo.length; j++ )
{
    collE = repoE.collectionInfo[ j ];
System.out.println("            startTime = " + collE.startTime + " endTime = " + collE.endTime +
    " interval = " + collE.interval );
}
    }

try
{
    attr = mco.rtvAttributes( "MCOA0200" );
} catch( Exception e)
{
    System.out.println("rtvAttributes(): MCOA0200: Exception " + e );
    System.exit(1);
}

System.out.println("MCOA0200: Object " + mco.getLibrary() + "/" + mco.getName() );
System.out.println("    size = " + attr.size + " retention = " + attr.retentionPeriod +
    " interval = " + attr.dftInterval + " time created = " + attr.timeCreated +
    " time updated = " + attr.timeUpdated );
System.out.println("    serial = " + attr.logicalPSN + " active = " + attr.isActive +
    " repaired = " + attr.isRepaired + " summary = " + attr.sumStatus +
    " repo count = " + attr.repositoryCount );
if( attr.repositoryInfo != null )
    for(i = 0; i < attr.repositoryCount; i++ )
    {
repoE = attr.repositoryInfo[ i ];
System.out.println("        name = " + repoE.name + " category = " + repoE.categoryName +
    " size = " + repoE.size );
for( j = 0; j < repoE.collectionInfo.length; j++ )
{
    collE = repoE.collectionInfo[ j ];
System.out.println("            startTime = " + collE.startTime + " endTime = " + collE.endTime +
    " interval = " + collE.interval );
}
    }

    if( repoName.equals("NONE") )
return;

try
{

```

```

        mco.open();
    } catch( Exception e)
    {
        System.out.println("open(): Exception " + e );
        System.exit(1);
    }

    try
    {
        repoHandle = mco.openRepository( repoName, "MCO00100" );
    } catch( Exception e)
    {
        System.out.println("openRepository(): Exception " + e );
        mco.close();
        System.exit(1);
    }
    System.out.println("repoHandle = " + repoHandle );

    MgtcolObjReadOptions readOptions = new MgtcolObjReadOptions();
    MgtcolObjRecInfo recInfo = new MgtcolObjRecInfo();

    readOptions.option = MgtcolObjReadOptions.READ_NEXT;
    readOptions.recKey = null;
    readOptions.offset = 0;
    readOptions.length = 0;

    while( recInfo.recStatus == MgtcolObjRecInfo.RECORD_OK )
    {
        try
        {
            mco.readData( repoHandle, readOptions, recInfo, null );
        } catch( Exception e)
        {
            System.out.println("readData(): Exception " + e );
            mco.close();
            System.exit(1);
        }

        if( recInfo.recStatus == MgtcolObjRecInfo.RECORD_OK )
        {
            System.out.print("Type = " + recInfo.recType );
            System.out.print(" Key = " + recInfo.recKey );
            System.out.println(" Length = " + recInfo.recLength );
        }

    }

    /* while ... */

    mco.closeRepository( repoHandle );
    mco.close();

    /* main() */

}/* class testmco2 */

```

컬렉션 오브젝트 관리

컬렉션 서비스를 사용하여 성능 자료를 수집하는 경우 각 컬렉션은 하나의 오브젝트에 저장됩니다. 다음 단계를 수행하여 관리 컬렉션 오브젝트에 있는 자료의 요약을 볼 수 있습니다.

1. iSeries Navigator에서 중앙 관리 아래의 종료점 시스템을 선택하거나 사용자 연결 (또는 사용자의 활동 환경) 아래에서 직접 연결되어 있는 시스템을 선택하십시오.
2. 구성 및 서비스를 확장하십시오.

3. 컬렉션 서비스를 선택하십시오.

4. 리스트에 있는 임의의 관리 컬렉션 오브젝트를 마우스 오른쪽 버튼으로 클릭하고 등록 정보를 선택하면 해당 컬렉션에 대한 일반 정보와 컬렉션에 들어 있는 자료의 요약을 볼 수 있습니다.

컬렉션 오브젝트를 마우스 오른쪽 버튼으로 클릭하고 데이터베이스 파일 작성을 선택하여 자료 범주, 컬렉션 기간 내의 시간 범위 및 데이터베이스 파일에 포함시킬 샘플링 간격을 지정할 수 있습니다.

컬렉션 오브젝트를 마우스 오른쪽 버튼으로 클릭하고 그래프 이력을 선택하여 관리 컬렉션 오브젝트에 있는 자료를 그래프로 볼 수 있습니다.

오래된 관리 컬렉션 오브젝트 삭제 또는 보존

오브젝트를 마우스 오른쪽 버튼으로 클릭하고 삭제를 선택하여 컬렉션 오브젝트를 시스템에서 삭제할 수 있습니다. 오브젝트를 수동으로 삭제하지 않을 경우 컬렉션 서비스가 만기일 및 시간 이후에 자동으로 오브젝트를 삭제합니다.

컬렉션 서비스는 순환된 관리 컬렉션 서비스만 삭제합니다. 순환된 상태란 컬렉션 서비스가 자료를 수집하고 오브젝트에 자료를 저장하는 조치를 중단했음을 의미합니다. 구성 및 서비스를 확장하고 컬렉션 서비스를 선택하면 각 관리 컬렉션 오브젝트의 상태가 리스트에 표시됩니다.

컬렉션 서비스는 다음 번에 컬렉션을 시작하거나 순환할 때 만기일 및 시간에 도달한 순환된 컬렉션 서비스를 삭제합니다. 만기일은 관리 컬렉션 오브젝트와 연관됩니다. 컬렉션 오브젝트를 다른 라이브러리로 이동한 경우에도 컬렉션 서비스는 오브젝트가 만기된 후에는 이를 삭제합니다.

각 관리 컬렉션 오브젝트의 만기일은 해당 컬렉션 오브젝트의 등록 정보에 표시됩니다. 오브젝트를 시스템에 오랫동안 보유하려면 등록 정보 페이지에서 날짜를 변경하면 됩니다. 리스트에 있는 임의의 관리 컬렉션 오브젝트를 마우스 오른쪽 버튼으로 클릭하고 등록 정보를 선택하면 해당 컬렉션에 대한 정보를 볼 수 있습니다. 컬렉션 서비스가 관리 컬렉션 오브젝트를 삭제하지 않게 하려면 영구를 지정하십시오.

사용자 정의 트랜잭션

컬렉션 서비스 및 Performance Explorer는 이제 사용자가 자신의 어플리케이션에서 정의한 성능 자료를 수집할 수 있습니다. 제공되는 API를 사용하여, 트랜잭션 자료를 컬렉션 서비스를 사용하여 정기적으로 스케줄된 샘플 자료 컬렉션으로 통합하고 Performance Explorer를 실행하여 트랜잭션에 대한 추적 레벨 자료를 얻을 수 있습니다.

자세한 설명 및 사용법에 대해서는 다음의 API를 참조하십시오.

- 트랜잭션 시작: QYPESTRT, qypeStartTransaction
- 트랜잭션 종료: QYPEENDT, qypeEndTransaction
- 트랜잭션 기록: QYPELOGT, qypeLogTransaction(Performance Explorer에서만 사용합니다)
- 추적점 추가: QYPEADDT, qypeAddTracePoint(Performance Explorer에서만 사용합니다)

주: 어플리케이션에 한 번만 기능을 부여하면 됩니다. 컬렉션 서비스와 Performance Explorer는 같은 API 호출을 사용하여 서로 다른 유형의 성능 자료를 수집합니다.

사용자 정의 트랜잭션 자료를 콜렉션 서비스에 통합

콜렉션 서비스 구성에서 사용자 정의 트랜잭션을 콜렉션에 대한 범주로 선택할 수 있습니다. 그러면 콜렉션 서비스는 모든 콜렉션 간격에서 트랜잭션 자료를 수집하여 이 자료를 콜렉션 오브젝트에 저장합니다. CRTPFRTA 명령은 이 자료를 사용자 정의 트랜잭션 성능 데이터베이스 파일 QAPMUSRTNS로 내보냅니다. 콜렉션 서비스는 트랜잭션 유형별로 자료를 분류합니다. 필요한 만큼 트랜잭션 유형을 지정할 수 있습니다. 그러나 콜렉션 서비스는 처음 15개의 트랜잭션 유형만을 보고합니다. 추가 트랜잭션 유형에 대한 자료는 *OTHER 트랜잭션 유형으로 결합되어 저장됩니다. 모든 콜렉션 간격에서 콜렉션 서비스는 각각의 고유한 작업에 대해 각 트랜잭션 유형마다 하나의 레코드를 작성합니다. 자세한 설명은 트랜잭션 시작 API의 사용법을 참조하십시오.

콜렉션 서비스는 트랜잭션 응답 시간과 같은 일반 트랜잭션 자료를 수집합니다. 또한 어플리케이션 특정 자료(예: 트랜잭션이나 기타 증분식 측정에 사용된 SQL문의 수)를 추적할 수 있는 선택적인 어플리케이션 정의 카운터를 최대 16개까지 포함시킬 수 있습니다. 어플리케이션은 트랜잭션 시작 API를 사용하여 새 트랜잭션의 시작을 표시해야 하고 대응하는 트랜잭션 종료 API를 포함시켜 트랜잭션 자료를 콜렉션 서비스에 전달해야 합니다. 자세한 정보는 QAMUSRTNS 파일 설명과 API 설명을 참조하십시오.

샘플 구현에 대해서는 C++ 또는 Java로 된 예를 참조하십시오.

주: 중요한 법적 고지사항에 관해 언급하는 코드 면책사항 관련 정보를 참조하십시오

Performance Explorer를 사용하여 사용자 정의 트랜잭션에 대한 추적 정보 수집

Performance Explorer 세션 중 트랜잭션 시작, 종료 및 기록 API를 사용하여 추적 레코드를 작성할 수 있습니다. Performance Explorer는 이들 추적 레코드의 현재 스레드에 대해 시스템 자원 활용(예: CPU 활용, I/O 및 점유/잠금 활동)을 저장합니다. 또한 어플리케이션 특성의 성능 자료를 포함한 다음 이들 각 API에서 이 자료를 Performance Explorer에 송신하도록 선택할 수도 있습니다. 추적점 추가 API를 사용하여 Performance Explorer가 추적 자료를 수집해야 하는 대상인 어플리케이션 특정 이벤트를 식별할 수도 있습니다.

사용자 트랜잭션에 대해 Performance Explorer 세션을 시작하려면 Performance Explorer 정의의(OSEVT) 매개변수에 *USRTRNS를 지정하십시오. ENDPEX 명령을 입력한 후 Performance Explorer는 어플리케이션이 제공한 자료를 QAYPEMIUSR Performance Explorer 데이터베이스 파일의 QMUDTA 필드에 기록합니다. 시작, 종료 및 임의의 기록부 레코드에 대한 시스템 제공 성능 자료는 QAYPETIDX 데이터베이스 파일의 QAYPEMIUSR에 저장됩니다.

자세한 설명은 API 설명과 트랜잭션 시작 API 설명의 사용법을 참조하십시오.

C++ 예: 사용자 정의 트랜잭션을 콜렉션 서비스에 통합: 다음의 C++ 예제 프로그램은 트랜잭션 시작 API 및 트랜잭션 종료 API를 사용하여 사용자 정의 트랜잭션 성능 자료를 콜렉션 서비스에 통합하는 방법을 보여줍니다.

주: 중요한 법적 고지사항에 관해 언급하는 코드 면책사항 관련 정보를 참조하십시오

```
//*****  
// tnstst.C  
//  
// 이 예는 트랜잭션 시작/종료 API
```

```

//(qypeStartTransaction, qypeEndTransaction)의
// 사용을 보여줍니다.
//
//
// 이 프로그램은 다음과 같이 하여 호출할 수 있습니다.
// CALL lib/TNSTST PARM('threads' 'types' 'transactions' 'delay')
// 여기서
//   threads      = 작성할 스레드 수(최대 10000개)
//   types        = 각 스레드에 대한 트랜잭션 유형의 수
//   transactions = 각 트랜잭션 유형에 대한 트랜잭션의
//                 수
//   delay        = 트랜잭션 시작 및 트랜잭션 종료 사이의
//                 지연 시간(밀리초)
//
// 이 프로그램은 "threads"개의 스레드를 작성합니다. 각 스레드는
// 같은 방식으로 트랜잭션을 생성합니다. 스레드는 각 트랜잭션 유형에 대해
// "transactions"개의 트랜잭션을 수행합니다.
// 여기서 트랜잭션은 차례로 트랜잭션 시작 API에 대한 호출,
// "delay" 밀리초의 지연, 트랜잭션 종료 API에 대한 호출로
// 정의됩니다. 따라서 각 스레드는 "transactions"에 "types"을
// 곱한 수만큼의 트랜잭션을 수행합니다. 각 트랜잭션 유형은
// "TRANSACTION_TYPE_nnn"에 따라 이름이 지어집니다. 여기서 nnn은 001에서 "types"까지의 범위입니다.
// 트랜잭션 유형 n의 경우 n-1(최대 16)의 사용자 제공 카운터가
// 보고되고 카운터 m의 경우에는 각 트랜잭션에 대해 m개의 카운터가
// 보고됩니다.
//
// 이 프로그램은 다중 스레드를 허용하는 작업에서 실행해야 합니다
// (보통 대화식 작업은 다중 스레드를 허용하지 않습니다). 이를 수행하는
// 한 가지 방법은 SBMJOB 명령을 사용하고 ALWMLTTHD(*YES)를 지정하여
// 프로그램을 호출하는 것입니다.
//
//*****

#define _MULTI_THREADED

// Includes
#include "pthread.h"
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include "qusec.h"
#include "lbcpynv.h"
#include "qypesvpg.h"

// Constants
#define maxThreads 10000

// Transaction pgm parm structure
typedef struct
{
    int types;
    int trans;
    int delay;
} tnsPgmParm_t;

// Error code structure
typedef struct
{
    Qus_EC_t error;
    char      Exception_Data[100];
} error_code_t;

//*****

```

```

//
// 각 2차 스레드에서 실행할 트랜잭션 프로그램
//
//*****
void *tnsPgm(void *parm)
{
    tnsPgmParm_t *p =(tnsPgmParm_t *)parm;

    char tnsTyp[] = "TRANSACTION_TYPE_XXX";
    char pexData[] = "PEX";
    unsigned int pexDataL = sizeof(pexData) - 1;
    unsigned long long colSrvData[16] = {1,2,3,4,5,6,7,8,
                                         9,10,11,12,13,14,15,16};

    unsigned int colSrvDataL;
    char tnsStrTim[8];

    struct timespec ts = {0, 0};

    error_code_t errCode;

    _DPA_Template_T target, source; // Used for LBCPYNV MI instr

    unsigned int typCnt;
    unsigned int tnsCnt;
    int rc;

    // Initialize error code
    memset(&errCode, 0, sizeof(errCode));
    errCode.error.Bytes_Provided = sizeof(errCode);

    // Initialize delay time
    ts.tv_sec = p->delay / 1000;
    ts.tv_nsec =(p->delay % 1000) * 1000000;

    // Loop doing transactions
    for(tnsCnt = 1; tnsCnt <= p->trans; tnsCnt++)
    {
        for(typCnt = 1; typCnt <= p->types; typCnt++)
        {
            // Set number field in transaction type
            source.Type = _T_UNSIGNED;
            source.Length = 4;
            source.reserved = 0;
            target.Type = _T_ZONED;
            target.Length = 3;
            target.reserved = 0;
            _LBCPYNV(tnsTyp + 17, &target, &typCnt, &source);

            // Set Coll Svcs data length in bytes
            colSrvDataL =(typCnt <= 16) ?(typCnt - 1) : 16;
            colSrvDataL = colSrvDataL * 8;

            // Call Start Transaction API
            qypeStartTransaction(tnsTyp,
                                (unsigned int *)&tnsCnt,
                                pexData,
                                (unsigned int *)&pexDataL,
                                tnsStrTim,
                                &errCode);

            // Delay specified amount
            rc = pthread_delay_np(&ts);

```

```

        // Call End Transaction API
        qypeEndTransaction(tnsTyp,
                          (unsigned int *)&tnsCnt,
                          pexData,
                          (unsigned int *)&pexDataL,
                          tnsStrTim,
                          (unsigned long long *)&colSrvData[0],
                          (unsigned int *)&colSrvDataL,
                          &errCode);
    }
}

return NULL;
}

//*****
//
// 1차 스레드에서 실행할 기본 프로그램
//
//*****

void main(int argc, char *argv[])
{
    // Integer version of parms
    int threads; // # of threads
    int types;   // # of types
    int trans;   // # of transactions
    int delay;   // Delay in millisecs

    pthread_t threadHandle[maxThreads];
    tnsPgmParm_t tnsPgmParm;
    int rc;
    int i;

    // Verify 4 parms passed
    if(argc != 5)
    {
        printf("Did not pass 4 parms\n");
        return;
    }

    // Copy parms into integer variables
    threads = atoi(argv[1]);
    types   = atoi(argv[2]);
    trans   = atoi(argv[3]);
    delay   = atoi(argv[4]);

    // Verify parms
    if(threads > maxThreads)
    {
        printf("Too many threads requested\n");
        return;
    }

    // Initialize transaction pgm parms(do not modify
    // these while threads are running)
    tnsPgmParm.types = types;
    tnsPgmParm.trans = trans;
    tnsPgmParm.delay = delay;

    // Create threads that will run transaction pgm

```



```

for(i=0; i < threads; i++)
{
    // Clear thread handle
    memset(&threadHandle[i], 0, sizeof(pthread_t));
    // Create thread
    rc = pthread_create(&threadHandle[i],      // Thread handle
                       NULL,                  // Default attributes
                       tnsPgm,                // Start routine
                       (void *)&tnsPgmParm); // Start routine parms

    if(rc != 0)
        printf("pthread_create() failed, rc = %d\n", rc);
}

// Wait for each thread to terminate
for(i=0; i < threads; i++)
{
    rc=pthread_join(threadHandle[i], // Thread handle
                    NULL);          // No exit status
}

} /* end of Main */

```

Java 예: 사용자 정의 트랜잭션을 콜렉션 서비스에 통합: 다음의 Java 예제 프로그램은 트랜잭션 시작 API 및 트랜잭션 종료 API를 사용하여 사용자 정의 트랜잭션 성능 자료를 콜렉션 서비스에 통합하는 방법을 보여줍니다.

주: 중요한 법적 고지사항에 관해 언급하는 코드 면책사항 관련 정보를 참조하십시오

```

import com.ibm.iseries.collectionservices.PerformanceDataReporter;

// parameters:
// number of TXs per thread
// number of threads
// log|nolog
// enable|disable
// transaction seconds

public class TestTXApi
{
    static TestTXApiThread[]  thread;

    static private String[] TxTypeString;
    static private byte[][] TxTypeArray;

    static private String TxEventString;
    static private byte[] TxEventArray;

    static
    {
        int i;

        // initialize transaction type strings and byte arrays

        TxTypeString = new String[20];
        TxTypeString[ 0] = "Transaction type 00";
        TxTypeString[ 1] = "Transaction type 01";
        TxTypeString[ 2] = "Transaction type 02";
        TxTypeString[ 3] = "Transaction type 03";
        TxTypeString[ 4] = "Transaction type 04";
        TxTypeString[ 5] = "Transaction type 05";
        TxTypeString[ 6] = "Transaction type 06";
        TxTypeString[ 7] = "Transaction type 07";
        TxTypeString[ 8] = "Transaction type 08";
        TxTypeString[ 9] = "Transaction type 09";
        TxTypeString[10] = "Transaction type 10";
        TxTypeString[11] = "Transaction type 11";
        TxTypeString[12] = "Transaction type 12";
    }
}

```

```

TxTypeString[13] = "Transaction type 13";
TxTypeString[14] = "Transaction type 14";
TxTypeString[15] = "Transaction type 15";
TxTypeString[16] = "Transaction type 16";
TxTypeString[17] = "Transaction type 17";
TxTypeString[18] = "Transaction type 18";
TxTypeString[19] = "Transaction type 19";

TxTypeArray = new byte[20] [];
for( i = 0; i < 20; i++ )
try
{
    TxTypeArray[i] = TxTypeString[i].getBytes("Cp037");
} catch(Exception e)
{
    System.out.println("Exception \"" + e + "\" when converting");
}

}/* static */

public static void main( String[] args )
{
    int    numberOfTXPerThread;
    int    numberOfThreads;
    boolean log;
    boolean enable;
    int    secsToDelay;

    // process parameters
    if( args.length >= 5 )
try
    {
        numberOfTXPerThread = Integer.parseInt( args[0] );
        numberOfThreads     = Integer.parseInt( args[1] );

        if( args[2].equalsIgnoreCase( "log" ) )
            log = true;
        else
            if( args[2].equalsIgnoreCase( "nolog" ) )
                log = false;
            else
                {
                    System.out.println( "Wrong value for 3rd parameter!" );
                    System.out.println( "\tshould be log|nolog" );
                    return;
                }

        if( args[3].equalsIgnoreCase( "enable" ) )
            enable = true;
        else
            if( args[3].equalsIgnoreCase( "disable" ) )
                enable = false;
            else
                {
                    System.out.println( "Wrong value for 4th parameter!" );
                    System.out.println( "\tshould be enable|disable" );
                    return;
                }

        secsToDelay = Integer.parseInt( args[4] );

    } catch(Exception e)
    {
        System.out.println( "Oops! Cannot process parameters!" );
        return;
    }
else
    {
        System.out.println( "Incorrect Usage." );
        System.out.println( "The correct usage is:" );
        System.out.println( "java TestTXApi numberOfTXPerThread numberOfThreads log|nolog enable|disable secsToDelay");
        System.out.println("\tlog will make the program cut 1 log transaction per start / end pair");
        System.out.println("\tdisable will disable performance collection to minimize overhead");
    }
}

```

```

        System.out.print("\nExample: \"java TestTXApi 10000 100 log enable 3\" will call " );
        System.out.println("cause 10000 transactions for each of 100 threads");
        System.out.println("with 3 seconds between start and end of transaction");
        System.out.println("Plus it will place additional log call and will enable reporting." );
        return;
    }

    System.out.println( "Parameters are processed:" );
    System.out.println( "\tnumberOfTxPerThread = " + numberOfTxPerThread );
    System.out.println( "\tnumberOfThreads = " + numberOfThreads );
    System.out.println( "\tlog = " + log );
    System.out.println( "\tenable = " + enable );
    System.out.println( "\tsecsToDelay = " + secsToDelay );

    // cause initialization of a PerformanceDataReporter class
    {
        PerformanceDataReporter pReporter = new PerformanceDataReporter();
        pReporter.enableReporting();
    }

    TestTXApi t = new TestTXApi( );

    System.out.println( "\nAbout to start ..." );
    t.prepareTests( numberOfTxPerThread, numberOfThreads, log, enable, secsToDelay );

    long startTime = System.currentTimeMillis();

    t.runTests( numberOfThreads );

    // wait for threads to complete
    for( int i = 0; i < numberOfThreads; i++ )
    try
    {
        thread[i].join( );
    } catch(Exception e)
    {
        System.out.println( "***Exception \"" + e + "\" while joining thread " + i );
    }

    long endTime = System.currentTimeMillis();

    System.out.println( "\nTest runtime for " + ( numberOfTxPerThread * numberOfThreads ) +
        " TXs was " + ( endTime - startTime ) + " msec" );

}/* main() */

private void prepareTests( int numberOfTxPerThread,
    int numberOfThreads, boolean log, boolean enable, int secsToDelay )
{
    System.out.println( "Creating " + numberOfThreads + " threads");
    thread = new TestTXApiThread[numberOfThreads];
    for( int i = 0; i < numberOfThreads; i++ )
        thread[i] = new TestTXApiThread( i, numberOfTxPerThread,
            log, enable, secsToDelay );
}/* prepareTests() */

private void runTests( int numberOfThreads )
{
    for( int i = 0; i < numberOfThreads; i++ )
        thread[i].start( );
}/* runTests() */

private class TestTXApiThread extends Thread
{
    private int    ordinal;
    private int    numberOfTxPerThread;
    private boolean log;
    private boolean enable;
    private int    secsToDelay;

    private PerformanceDataReporter    pReporter;

    private long    timeStamp[];

```

```

private long    userCounters[];

public TestTXApiThread( int ordinal, int numberOfTxPerThread,
                       boolean log, boolean enable, int secsToDelay )
{
    super();
    this.ordinal          = ordinal;
    this.numberOfTxPerThread = numberOfTxPerThread;
    this.log              = log;
    this.enable           = enable;
    this.secsToDelay      = secsToDelay;

    pReporter = new PerformanceDataReporter( false );
    if( enable )
        pReporter.enableReporting();
    timeStamp = new long[1];
    userCounters = new long[16];
    for( int i = 0; i < 16; i++ )
        userCounters[i] = i;

    /* constructor */

    public void run()
    {
int        i;

        for( i = 0; i < numberOfTxPerThread; i++ )
        {
            pReporter.startTransaction( TxTypeArray[i%20], i, TxTypeArray[i%20], 20, timeStamp );
//            pReporter.startTransaction( TxTypeArray[i%20], i, TxTypeString[i%20], timeStamp );
            if( log )
                pReporter.logTransaction( TxTypeArray[i%20], i, TxTypeArray[i%20], 20 );
//            pReporter.logTransaction( TxTypeArray[i%20], i, TxTypeString[i%20] );
            if(secsToDelay > 0)
                try
                {
                    Thread.sleep(secsToDelay * 1000);
                } catch(Exception e) { }
            pReporter.endTransaction( TxTypeArray[i%20], i, TxTypeArray[i%20], 20, timeStamp,
                                     userCounters );
//            pReporter.endTransaction( TxTypeArray[i%20], i, TxTypeString[i%20], timeStamp,
//                                     userCounters );
        }

        /* run() */

    } /* class TestTXApiThread */
} /* class TestTXApi */

```

성능 자료 파일

성능 자료는 응답 시간 및 처리량을 이해하는 데 사용될 수 있는 시스템 또는 시스템 네트워크 운영에 대한 정보 세트입니다. 성능 자료를 사용하여 프로그램, 시스템 속성 및 운영을 조정할 수 있습니다. 이렇게 조정함으로써 응답 시간 및 처리량을 향상시킬 수 있습니다. 조정으로 시스템, 운영 또는 프로그램 상의 변화가 미치는 영향을 예측할 수 있습니다.

컬렉션 서비스는 성능 자료를 수집하여 관리 컬렉션 오브젝트(*MGTCOL)를 생성합니다. CRTPFDRDTA(성능 자료 작성) 명령은 컬렉션 오브젝트로부터 자료를 처리하여 그 결과를 성능 자료 파일에 저장합니다. 데이터베이스 파일은 다음과 같은 범주로 나뉩니다.

시간 간격 자료가 들어 있는 성능 자료 파일

이 파일에는 각 간격마다 수집된 성능 자료가 들어 있습니다. 이들 파일의 리스트, 간략한 설명 및 각 파

일의 전체 정보에 대한 링크에 대해서는 시간 간격 자료가 들어 있는 성능 자료 파일을 참조하십시오. 이 파일의 자료가 어디에서 온 것인지를 이해하려면 시스템 범주 및 파일 관계를 참조하십시오. 이들 파일을 볼 때 파일 약어도 유용합니다.

구성 자료 파일

구성 자료는 한 세션에 한 번 수집됩니다. 이 파일의 자료가 어디에서 온 것인지를 이해하려면 시스템 범주 및 파일 관계를 참조하십시오. 구성 자료 파일에서 QAPMCONF, QAPMHDWR 및 QAPMSBSD 파일을 찾을 수 있습니다.

추적 자료 파일

추적 자료는 추적 자료 수집을 선택했을 때에만 수집됩니다. 추적 자료 파일에서 QAPMDMPT 파일을 찾을 수 있습니다.

바이트 수와 버퍼 위치와 같은 추가적인 필드 정보는 시스템에서 사용할 수 있는 DSPFFD(파일 필드 설명 표시) 명령으로 사용할 수 있습니다. 예를 들면 명령행에 다음을 입력하십시오.

```
DSPFFD file(QSYS/QAPMCONF)
```

iSeries 성능에 대한 자세한 정보는 성능을 참조하십시오.

시간 간격 자료가 들어 있는 성능 자료 파일

성능 자료 파일의 전체 내용을 보려면 알파벳 순서로 표시된 아래의 리스트에서 보려는 파일을 선택하십시오.

파일	설명
QAPMAPPN	APPN 자료
QAPMASYN	비동기 통계(링크 당 하나)
QAPMBSC	2진 동시 통계(링크 당 하나)
QAPMBUS	버스 카운터(버스 당 하나)
QAPMCIOP	통신 IOP 자료(IOP 당 하나)
QAPMDDI	분산 디지털 인터페이스(DDI) 자료(링크 당 하나)
QAPMDIOP	기억장치 IOP 자료(IOP 당 하나)
QAPMDISK	디스크 기억장치 자료(읽기/쓰기 헤드 당 하나)
QAPMDOMINO	iSeries용 Domino 자료(domino 서버 당 한 레코드)
QAPMECL	토큰링 파일 항목(링크 당 하나)
QAPMETH	이더넷 통계(링크 당 하나)
QAPMFRLY	프레임 릴레이 자료(링크 당 하나)
QAPMHDLC	HDLC 통계(링크 당 하나)
QAPMHTTPB	IBM HTTP Server용 기본 자료(Apache로 구동)(서버 당 하나)
QAPMHTTPD	IBM HTTP Server 상세 자료(Apache로 구동)(서버 구성요소 당 하나)
QAPMIDLC	종합 정보 통신망(ISDN) 자료 링크 제어 파일 항목(링크 당 하나)
QAPMIOPD	확장 IOP 자료(특수 IOP에 대한 추가 자료(IOP 당 하나))
QAPMJOBMI	MI 작업 자료(작업, 태스크 또는 스레드 하나에 레코드 하나). 이 문서를 사용할 때 태스크 유형 확장자 정보가 유용합니다.
QAPMJOBOS	작업 오퍼레이팅 시스템 자료(작업 당 한 레코드)

파일	설명
QAPMJOBS 및 QAPMJOB	작업 자료(작업, 태스크 또는 스레드 당 한 레코드)
QAPMJOBWT	작업, 태스크 및 스레드 대기 조건
QAPMJOBWTD	QAPMJOBWT 파일에 있는 카운트 세트의 설명.
QAPMJSUM	작업 그룹별 작업 요약 자료(작업 그룹 당 한 레코드)
QAPMLAPD	종합 정보 통신망(ISDN) LAPD 파일 항목(링크 당 하나)
QAPMLIOP	쌍축 워크스테이션 제어기 자료(실제 제어기 당 하나)
QAPMMIOP	다기능 IOP(IOP 당 하나)
QAPMPOOL 및 QAPMPOOLL	주 기억장치 자료(시스템 기억장치 풀 당 하나)
QAPMPOOLB	기억장치 풀 자료(풀 당 하나)
QAPMPOOLT	기억장치 풀 조정 자료(풀 당 하나)
QAPMPPP	지점 간 프로토콜 자료(링크 당 하나)
QAPMRESP	로컬 워크스테이션 응답 시간(워크스테이션 당 하나)
QAPMRWS	리모트 워크스테이션 응답 시간
QAPMSAP	TRLAN, 이더넷, DDI 및 프레임 릴레이 SAP 파일 항목(SAP 항목 당 하나)
QAPMSNA	SNA 자료
QAPMSNADS	SNADS 자료(SNADS 작업 당 하나)
QAPMSTND	DDI 스테이션 자료
QAPMSTNE	이더넷 스테이션 파일 항목
QAPMSTNL	토큰링 스테이션 파일 항목
QAPMSTNY	프레임 릴레이 스테이션 파일 항목
QAPMSYS 및 QAPMSYSL	시스템 성능 자료
QAPMSYSCPU	시스템 CPU 사용 자료
QAPMSYSTEM	시스템 레벨 성능 자료
QAPMTCP	TCP/IP 자료
QAPMTCPIFC	개별 TCP/IP 인터페이스에 대한 TCP/IP 자료
QAPMUSRTNS	사용자 정의 트랜잭션 자료(각 작업은 각 트랜잭션 유형에 대해 하나의 레코드를 갖습니다)
QAPMX25	X.25 통계(링크 당 하나)

성능 자료 파일: 파일 약어

성능 자료 파일은 필드와 바이트 자료 표에 약어를 사용합니다. 이러한 약어에는 다음과 같은 것들이 있습니다.

약어	설명
1차 파일	이 파일들은 해당 범주와 관련되어 있으며 이 범주로부터 생성됩니다.
C	속성 열의 문자
PD	속성 열의 팩 10진수
Z	속성 열의 존 10진수
IOP	입/출력 프로세서, 호스트 시스템과 디스크, 표시장치 및 통신 회선과 같은 다른 장치 간의 활동을 제어하는 프로세서.
DCE	자료 회로 단말장치
MAC	매체 액세스 제어. 통신 IOP의 엔티티.

LLC	논리 링크 제어. 통신 IOP의 엔터티.
Beacon 프레임	링이 작동할 수 없게될 때 송신되는 프레임
유형 II 프레임	시스템 네트워크 구조(SNA)가 사용하는 연결 지향 프레임(정보 프레임)
I-프레임	정보 프레임

성능 자료 파일: 콜렉션 서비스 시스템 범주 및 파일 관계

콜렉션 서비스를 사용하여 성능 자료를 수집할 때 자료는 관리 콜렉션(*MGTCOL) 오브젝트에 저장됩니다. CRTPFRTA 명령은 이 관리 콜렉션 오브젝트에서 자료를 내보내기 한 후 성능 자료 파일에 자료를 기록합니다. 콜렉션 서비스에 의해 개별적으로 제어되고 수집될 수 있는 자료의 각 유형은 자료 범주로 표시할 수 있습니다. 각 자료 범주에는 하나 이상의 성능 자료 파일에 기록된 자료가 들어있거나 이를 제공합니다. 작성될 데이터베이스 파일이나 멤버의 경우 그 파일이나 멤버가 속한 범주 또는 범주 그룹이 존재해야 하고 CRTPFRTA에 의해 처리되어야 합니다. 아래의 표에서는 범주 대 파일의 관계를 보여줍니다. 관계에는 세 가지 유형이 있습니다.

관계	설명
1차 파일	이 파일들은 해당 범주와 관련되어 있으며 이 범주로부터 생성됩니다.
호환성 파일	이 파일들은 성능 데이터베이스 호환성을 제공하기 위해 1차 파일과 이전의 파일 구조를 결합한 논리 파일입니다. 시스템이 모든 참여 파일(1차 범주)을 생성하면 호환성 파일도 역시 생성됩니다.
2차 파일	이 파일들은 해당 범주 또는 1차 파일에 들어 있는 자료로부터 파생된 자료와 관련이 있고 그 자료를 가지고 있습니다. 그러나 이 파일들이 해당 범주에 의해 제어되는 않습니다.

다음 사항을 주의하십시오.

1. CRTPFRTA 명령은 해당 파일이 선택된 범주에 대한 1차 파일인 경우에만 데이터베이스 파일을 생성합니다.
2. 하나의 1차 파일이 여러 범주에 대하여 나열되어 있는 경우, 해당 파일을 생성하려면 이들 범주 각각을 선택해야 합니다.
3. 한 범주에 대한 1차 파일이 다른 범주에 대한 2차 파일로 나열되어 있는 경우 생성된 데이터베이스 파일 정보의 무결성을 보장하기 위해 두 번째 범주를 선택해야 합니다. 예를 들어 아래 표에서와 같이 QAPMECL에 대한 완전한 데이터베이스 파일을 생성하려면 *CMNBASE와 *CMNSTN 모두를 선택해야 합니다.
4. 시스템은 모든 연관된 1차 파일을 생성할 때에만 호환성 파일을 생성합니다.

아래의 표에서는 시스템 범주와 성능 데이터베이스 파일간의 관계를 보여줍니다.

범주	1차 파일	호환성 파일	2차 파일
*SYSBUS	QAPMBUS		
*POOL	QAPMPOOLB	QAPMPOOLL	
*POOLTUNE	QAPMPOOLT	QAPMPOOLL	
*HDWCFG	QAPMHDWR		
*SUBSYSTEM	QAPMSBSD		

*SYSCPU	QAPMSYSCPU	QAPMSYSL	
*SYSLVL	QAPMSYSTEM	QAPMSYSL	
*JOBMI	QAPMJOBMI QAPMJOBWT QAPMJOBWTD QAPMJSUM	QAPMJOBL QAPMSYSL	QAPMSYSTEM
*JOBOS	QAPMJOBOS QAPMJSUM	QAPMJOBL QAPMSYSL	QAPMSYSTEM
*SNADS	QAPMSNADS		
*DISK	QAPMDISK		QAPMSYSTEM
*IOPBASE	QAPMLIOP QAPMDIOP QAPMCIOP QAPMMIOP		
*IPCS	QAPMIOPD QAPMTSK		
*CMNBASE	QAPMASYN QAPMBSC QAPMDDI QAPMECL QAPMETH QAPMFRLY QAPMHDLC QAPMIDLC QAPMLAPD QAPMPPP QAPMX25		
*CMNSTN	QAPMSTND QAPMSTNE QAPMSTNL QAPMSTNY 없음		QAPMDDI QAPMETH QAPMECL QAPMFRLY QAPMX25
*CMNSAP	QAPMSAP		
*LCLRSP	QAPMRESP		
*APPN	QAPMAPPN		
*SNA	QAPMSNA		
*EACACHE	없음		QAPMDISK (주 참조)
*TCPBASE	QAPMTCP		
*TCPIFC	QAPMTCPIFC		
*DOMINO	QAPMDOMINO		
*HTTP	QAPMHTTPB QAPMHTTPD		
*USRTNS	QAPMUSRTNS		

주:

이 범주는 CRTPFRTA 명령으로 선택할 수 없습니다. 그러나 *DISK 범주에 의해 추가적인 자료가 보고되도록 합니다.

성능 자료 파일: 구성 데이터베이스 파일의 필드 자료

구성 자료는 한 세션에 한 번 수집됩니다. 다음의 성능 자료 파일에서는 시스템 구성 자료, 서브시스템 자료 및 하드웨어 구성 자료에 대하여 파일명, 간략한 설명 및 자세한 필드 자료 참조(있는 경우)를 보여줍니다. 콜렉션 서비스가 이 파일을 생성하는 방법과 이 자료의 출처에 대한 정보는 시스템 범주 및 파일 관계를 참조하십시오.

필드명	설명
QAPMCONF	시스템 구성 자료.
QAPMHDWR	시스템 하드웨어 구성.
QAPMSBSD	서브시스템 자료. 필드 및 바이트 자료가 없습니다.

이 주제에 대한 자세한 정보는 성능 데이터베이스 파일 개요를 참조하십시오.

성능 데이터베이스 파일: 추적 데이터베이스 파일의 필드 자료

추적 자료에는 내부 시스템 추적 자료가 포함됩니다. 이것은 특정 작업과 트랜잭션에 대하여 추가 정보를 얻기 위해 수집한 상세 자료입니다. 이 유형의 자료는 분석을 위해 성능 분석 툴 사용권 프로그램을 사용하지 않는 한 수집할 수 없습니다. 다음은 STRPFRTRC(성능 추적 시작) 명령을 사용할 때 시스템이 지원하는 성능 자료 파일입니다.

파일명	설명
QAPMDMPT	시스템 추적 자료(필드 또는 바이트 세부사항이 없음)

이 주제에 대한 자세한 정보는 성능 데이터베이스 파일 개요를 참조하십시오.

iSeries Navigator 모니터

iSeries Navigator에 포함된 모니터는 콜렉션 서비스 자료를 사용하여 관심있는 시스템 성능의 요소를 추적할 수 있습니다. 더욱이 CPU 활용 비율이나 작업 상태와 같은 특정 이벤트가 발생할 경우 지정된 조치를 수행할 수 있습니다. 모니터를 사용하여 여러 개의 시스템과 시스템 그룹에서 발생하는 시스템 성능을 보고 관리할 수 있습니다.

모니터가 있으면 모니터를 시작한 후 사용자 서버, iSeries Navigator 또는 사용자의 PC에 있는 다른 타스크로 전환할 수 있습니다. 사실상 PC를 끌 수도 있습니다. iSeries Navigator는 사용자가 지정한 임계값 명령이나 조치를 지속적으로 모니터링 및 수행합니다. 사용자가 중단할 때까지 모니터는 실행됩니다. 또한 모니터를 사용하면 무선 iSeries Navigator로 모니터에 액세스하여 성능을 원격으로 관리할 수도 있습니다.

iSeries Navigator는 다음 유형의 모니터를 제공합니다.

시스템 모니터

성능 자료를 발생하는 순간 또는 최대 1시간까지 수집하여 표시합니다. 자세한 그래프로 서버에서 발생하는 상황을 시각화할 수 있습니다. 다양한 계측 방법(성능 계측)을 선택하여 특정 부분의 시스템 성능을 정확하게 나타낼 수 있습니다. 예를 들어 사용자 서버에서의 평균 CPU 활용을 모니터링하는 경우 그래프의 아무 콜렉션

점을 클릭하여 CPU 활용이 높은 상위 20개의 작업을 보여주는 상세 도표를 볼 수 있습니다. 그런 후 이들 중 임의의 작업을 마우스 오른쪽 버튼으로 클릭하여 직접 작업에 대해 작업할 수 있습니다.

작업 모니터

작업명, 작업 사용자, 작업 유형, 서브시스템 또는 서버 유형에 기초하여 단일 작업 또는 작업 리스트를 모니터링합니다. 다양한 계측 방법 중 하나를 선택하여 작업에 대한 성능, 상태 또는 오류 메시지를 모니터링합니다. 작업에 대해 직접 작업하려면 작업 모니터 창에 표시되는 리스트에서 작업을 마우스 오른쪽 버튼으로 클릭하기만 하면 됩니다.

메세지 모니터

어플리케이션이 정상적으로 완료되었는지 알아내거나 기업 요구사항에 중요한 특정 메시지를 모니터링합니다. 메시지 모니터 창에서 메시지의 세부사항을 확인하고 메시지에 응답하며 메시지를 송신하고 메시지를 삭제할 수 있습니다.

B2B 활동 모니터

iSeries용 Connect와 같은 어플리케이션이 구성되어 있는 경우 B2B 활동 모니터를 사용하여 B2B 트랜잭션을 모니터링할 수 있습니다. 일정 기간 동안의 활동 트랜잭션 그래프를 보고 임계값이 트리거되면 자동으로 명령을 실행할 수 있습니다. 특정 트랜잭션을 탐색 및 표시하고 이 트랜잭션의 상세 단계에 대한 막대 그래프도 볼 수 있습니다.

파일 모니터

하나 이상의 선택된 파일을 모니터링하여 지정된 텍스트 스트링, 지정된 크기 또는 파일 수정을 찾습니다.

모니터에 대한 자세한 정보는 다음 주제를 참조하십시오.

모니터 개념

모니터는 실시간 성능 자료를 표시할 수 있습니다. 또한 지정된 임계값에 도달하면 선택된 명령을 실행하기 위해 시스템을 지속적으로 모니터링할 수 있습니다. 모니터의 작동 방식, 모니터링할 수 있는 내용, 주어진 성능 상황에 응답할 수 있는 방법에 대해 배우십시오.

모니터 구성

iSeries Navigator에서 모니터를 구성할 수 있습니다. 이 주제에서 모니터를 설정하는 방법과 사용 가능한 옵션을 최대한 활용하도록 모니터를 구성하는 방법에 대해 배우십시오.

시나리오

이 주제에서는 시스템의 특정한 성능 측면을 살펴보기 위해 여러 가지 유형의 모니터를 사용하는 방법을 보여주는 시나리오를 제공합니다.

모니터 개념

시스템 모니터는 콜렉션 서비스에 의해 생성 및 유지보수되는 콜렉션 오브젝트에 저장된 자료를 표시합니다. 시스템 모니터는 최대 한 시간 동안 수집되는 자료를 표시합니다. 더 긴 기간 동안 자료를 보려면 그래프 이력을 사용해야 합니다. 모니터 등록 정보에 있는 자료 콜렉션의 빈도를 변경할 수 있으며 이 값은 콜렉션 서비스의 설정을 대체합니다.

모니터를 사용하여 시스템 성능의 여러 가지 다양한 요소를 추적 및 연구할 수 있고 다수의 서로 다른 모니터가 동시에 실행하게 할 수 있습니다. 여러 개의 모니터를 함께 사용할 경우 시스템 성능을 관찰 및 관리할 수 있는 복잡한 툴을 모니터가 제공합니다. 예를 들어 새로운 대화식 어플리케이션을 구현하는 경우 시스템 모니터를 사용하여 작업의 자원 활용에 대해 우선순위를 정하고 작업 모니터를 사용하여 문제가 있는 작업을 찾아 처리하며 메세지 모니터를 사용하여 지정된 메세지가 시스템에서 발생하는 경우에 사용자에게 경고할 수 있습니다.

임계값 및 조치 설정

새로운 모니터를 작성할 때 시스템 미터법이 지정된 임계 레벨에 도달하거나 이벤트 발생 시 발생시킬 조치를 지정할 수 있습니다. 임계 레벨이나 이벤트 발생 시 송신 또는 작업 대기행렬 보류와 같은 OS/400 명령을 종료점 시스템에서 실행하도록 선택할 수 있습니다. 또한 이벤트 기록부를 갱신하거나, PC에서 경보음을 내거나 모니터를 실행하여 사용자에게 경고하는 등의 몇 가지 사전 정의된 조치를 모니터가 수행하도록 선택할 수도 있습니다. 마지막으로 임계 레벨에 도달했을 때 모니터가 자동으로 정상 활동을 재개하게 하는 두 번째 임계 레벨을 지정하여 모니터를 자동으로 재설정할 수 있습니다.

모니터 구성

시스템 모니터는 종료점 시스템으로부터 실시간 성능 자료를 수집 및 표시할 때 사용할 수 있는 대화식 툴입니다. 새 모니터 작성은 새 모니터 창에서 시작하는 빠르고 쉬운 프로세스입니다.

1. iSeries Navigator에서 중앙 관리를 확장하고 모니터를 선택한 후 시스템을 마우스 오른쪽 버튼으로 클릭한 다음 새 모니터를 선택하십시오.
2. 모니터명을 지정합니다. 새 모니터 - 일반 페이지에서 모니터의 이름을 지정하십시오. 모니터 리스트에서 모니터를 찾을 수 있도록 간략한 설명을 제공하십시오.
3. 미터법을 선택합니다. 새 모니터 - 미터법 페이지를 이용하여 미터법을 선택하십시오. 종료점 시스템 또는 시스템 그룹에 대해 수에 관계 없이 원하는 만큼 미터법을 모니터할 수 있습니다.
4. 미터법 정보를 보고 변경합니다. 새 모니터 - 미터법 페이지를 이용하여 각 미터법에 대한 등록 정보를 편집하십시오. 선택한 각 미터법에 대해 콜렉션 간격, 최대 그래프 값 및 표시 기간을 편집할 수 있습니다.
5. 임계값 명령을 설정합니다. 미터법 페이지의 임계값 탭을 사용하여 임계값을 작동시키고 임계값이 트리거되거나 재설정될 때마다 종료점 시스템에서 실행할 명령을 지정할 수 있습니다.
6. 임계 조치를 설정합니다. 새 모니터 - 조치 페이지를 사용하여 미터법 임계값이 트리거되거나 재설정될 때 발생시킬 조치를 지정할 수 있습니다.
7. 시스템과 그룹을 선택합니다. 새 모니터 - 시스템 및 그룹 페이지를 이용하여 모니터를 시작하려는 종료점 시스템이나 시스템 그룹을 선택할 수 있습니다.

모니터를 작성한 다음 모니터명을 마우스 오른쪽 버튼으로 클릭한 후 시작을 선택하여 모니터를 실행하고 모니터 그래프에 대한 작업을 시작하십시오.

모니터 미터법

시스템 성능을 효율적으로 모니터링하려면 시스템 성능의 어떤 측면을 모니터링할지를 결정해야 합니다. 중앙 관리는 시스템 성능의 서로 다른 측면을 정확히 가려낼 수 있도록 미터법이라고 하는 다양한 성능 측정을 제공합니다.

새 모니터 창의 미터법 페이지를 통해 모니터링하려는 미터법을 보고 변경할 수 있습니다. 이 페이지에 액세스하려면 모니터를 선택하고 마우스 오른쪽 버튼으로 시스템을 클릭한 후 새 모니터를 선택하십시오. 필수 필드에 값을 기입하고 미터법 탭을 클릭하십시오.

모니터를 구성할 때 사용자의 모니터에 포함시킬 임의의 미터법, 미터법 그룹 또는 리스트의 모든 모니터를 사용할 수 있습니다. 사용자의 모니터에 사용할 수 있는 미터법 유형으로는 다음이 있습니다.

미터법 그룹: CPU 활용	미터법 설명: 시스템의 작업이 소비하는 사용 가능한 처리 장치 시간의 백분율. 다음 중에서 사용자의 모니터에서 사용할 CPU 활용 미터법의 유형을 선택하십시오. <ul style="list-style-type: none">• CPU 활용(평균)• CPU 활용(대화식 작업)• CPU 활용(대화식 피쳐)• CPU 활용(데이터베이스 기능)• CPU 활용(2차 작업부하)• CPU 활용 기본(평균) 이들 미터법과 이의 사용 방법에 대해 자세히 알려면 새 모니터 창의 일반 탭이나 iSeries Navigator의 모니터 등록 정보 창에 제공되는 온라인 도움말을 참조하십시오.
대화식 응답 시간 (평균 및 최대)	시스템에서 대화식 작업이 경험하는 응답 시간.
트랜잭션 비율 (평균)	시스템의 모든 작업이 완료한 초당 트랜잭션 수.
트랜잭션 비율 (대화식)	다음 유형의 작업에서 완료한 초당 트랜잭션 수 <ul style="list-style-type: none">• 대화식• MRT(Multiple requester terminal)• System/36 환경 대화식• passthru
일괄처리 논리 데이터베이스 I/O	시스템에서 일괄처리 작업에 의해 현재 수행되는 논리 데이터베이스 입/출력 조작의 평균 수
디스크 암(arm) 활용 (평균 및 최대)	자료 수집 중 시스템에서 사용된 디스크 암(arm) 용량의 백분율
디스크 기억장치 (평균 및 최대)	자료 수집 중 시스템에서 가득 찬 상태의 디스크 암(arm) 기억장치의 백분율
디스크 IOP 활용 (평균 및 최대)	자료 수집 중 시스템에서 디스크 입/출력 프로세서(IOP)의 사용 빈도

통신 IOP 활용 (최대 및 평균)	자료 수집 중 시스템에서 통신 입/출력 프로세서(IOP)의 사용 빈도
통신 회선 활용 (평균 및 최대)	모든 시스템 통신 회선에서 실제 송신 및 수신된 자료의 양.
LAN 활용 (최대 및 평균)	사용자의 근거리 통신망(LAN) 통신 회선에서 실제 송신 및 수신된 자료의 양.
기계 풀(pool) 결합	시스템의 기계 풀(pool)에서 발생한 초당 결합 수.
사용자 풀(pool) 결합 (최대 및 평균)	시스템의 모든 사용자 풀(pool)에서 발생한 초당 결합 수.

도움말이 필요하면 새 모니터-미터법 창의 도움말 버튼을 클릭하십시오. 중앙 관리 미터법에 익숙해지면 컴퓨팅 환경의 정보 요구에 따라 선택하는 미터법이 달라집니다. 찾고자 하는 정보에 목표를 둔 미터법을 선택하였으면 모니터에 대해 선택한 각 미터법에 대한 자세한 정보를 보고 변경할 준비가 된 것입니다.

시나리오: iSeries Navigator 모니터

iSeries Navigator에 포함된 모니터는 시스템 성능을 검토하고 관리하기 위한 강력한 툴 세트를 제공합니다. iSeries Navigator가 제공하는 모니터 유형의 개요를 보려면 iSeries Navigator 모니터를 참조하십시오.

자세한 사용법 예와 샘플 구성에 대해서는 다음 시나리오를 참조하십시오.

시나리오: 시스템 모니터

CPU 활용도가 너무 높아지고, 더 많은 자원을 사용할 수 있게 될 때까지 일시적으로 낮은 우선순위의 작업을 보유하고 있는 경우 사용자에게 경고하는 예제 시스템 모니터를 참조하십시오.

시나리오: 메시지 모니터

iSeries 서버에서 발생하는 메시지 대기행렬에 대한 조회 메시지를 표시하는 예제 메시지 모니터를 참조하십시오. 모니터가 열리고 감지된 메시지가 바로 표시됩니다.

시나리오: 작업 모니터

지정된 작업의 CPU 활용을 추적하여 CPU 활용도가 너무 높을 경우 작업의 소유자에게 경고하는 예제 작업 모니터를 참조하십시오.

시나리오: 시스템 모니터:

상황

시스템 관리자로서 사용자는 사용자의 요구와 비즈니스 요구사항을 충족시킬 수 있을 만큼의 충분한 자원이 iSeries 시스템에 있는지 확인해야 합니다. 사용자 시스템에는 CPU 활용이 특히 중요한 시안입니다. CPU 활용도가 너무 높아지고, 더 많은 자원을 사용할 수 있게 될 때까지 일시적으로 낮은 우선순위의 작업을 보유하고 있는 경우 시스템이 사용자에게 경고하게 할 수 있습니다.

이를 수행하기 위해 CPU 활용이 80%를 초과하면 메시지를 송신하는 시스템 모니터를 설정할 수 있습니다. 더욱이 CPU 활용이 작업 해제점인 60%로 떨어지고 정상 조작을 재개할 때까지 QBATCH 작업 대기행렬의 모든 작업을 보유할 수 있습니다.

구성 예

시스템 모니터를 설정하려면 추적할 미터법과 미터법이 지정된 레벨에 도달하면 모니터가 수행할 조치를 정의해야 합니다. 이 목표를 달성하는 시스템 모니터를 정의하려면 다음 단계를 완료하십시오.

1. iSeries Navigator에서 **중앙 관리 > 모니터**를 확장하고 **시스템 모니터**를 마우스 오른쪽 버튼으로 클릭하고 **새 모니터...**를 선택하십시오.
2. 일반 페이지에서 이 모니터의 이름과 설명을 입력하십시오.
3. 미터법 탭을 클릭하고 다음 정보를 입력하십시오.
 - a. 사용 가능한 미터법 리스트에서 **CPU 활용 기본(평균)**을 선택하고 **추가**를 클릭하십시오. CPU 활용 기본(평균)은 모니터할 미터법 아래에 나열되고 창의 맨 아래 부분에는 이 미터법에 대한 설정이 표시됩니다.
 - b. 컬렉션 간격에 이 자료를 수집하려는 빈도를 지정하십시오. 이 값이 컬렉션 서비스 설정을 대체합니다. 이 예에서는 **30초**를 지정하십시오.
 - c. 이 미터법에 대한 모니터 그래프의 수직 축에 대한 스케일을 변경하려면 **최대 그래프 값**을 변경하십시오. 이 미터법에 대한 그래프의 수평 축에 대한 스케일을 변경하려면 **표시 시간 값**을 변경하십시오.
 - d. 미터법 설정에 대한 **임계값 1** 탭을 클릭하고 CPU 활용이 80%이상이면 **조회** 메시지를 송신하려면 다음 값을 입력하십시오.
 - 1) **임계값 작동 가능**을 선택하십시오.
 - 2) 임계 트리거 값에 **>= 80(80% 이상 사용)**을 지정하십시오.
 - 3) 기간에 **1** 간격을 지정하십시오.
 - 4) **OS/400 명령**에 다음을 지정하십시오.

```
SNDRMSG MSG('Warning,CPU...') TOUSR(*SYSOPR) MSGTYPE(*INQ)
```
 - 5) 임계 재설정 값에 **< 60(60% 미만 사용)**을 지정하십시오. 이 값은 CPU 활용이 60% 아래로 떨어지면 모니터를 재설정합니다.
 - e. **임계값 2** 탭을 클릭하고 다섯 개의 컬렉션 간격 동안 CPU 활용이 80% 이상을 유지하는 경우 QBATCH 작업 대기행렬에 모든 작업을 보유하려면 다음 값을 입력하십시오.
 - 1) **임계값 작동 가능**을 선택하십시오.
 - 2) 임계 트리거 값에 **>= 80(80% 이상 사용)**을 지정하십시오.
 - 3) 기간에 **5** 간격을 지정하십시오.
 - 4) **OS/400 명령**에 다음을 지정하십시오.

```
HLDJOBQ JOBQ(QBATCH)
```
 - 5) 임계 재설정 값에 **< 60(60% 미만 사용)**을 지정하십시오. 이 값은 CPU 활용이 60% 아래로 떨어지면 모니터를 재설정합니다.

- 6) 기간에 5 간격을 지정하십시오.
- 7) **OS/400** 명령에 다음을 지정하십시오.

RLSJOBQ JOBQ(QBATCH)

이 명령은 CPU 활용이 다섯 개의 콜렉션 간격 동안 60% 미만을 유지하는 경우 QBATCH 작업 대기행렬을 해제합니다.

4. 조치 탭에서 트리거와 재설정 열에서 이벤트 기록을 선택하십시오. 이 조치는 임계값이 트리거되어 재설정 될 때 이벤트 기록부에 항목을 작성합니다.
5. 시스템 및 그룹 탭을 클릭하여 모니터링할 시스템과 그룹을 지정하십시오.
6. 확인을 클릭하여 모니터를 저장하십시오.
7. 시스템 모니터의 리스트에서 새 모니터를 마우스 오른쪽 버튼으로 클릭하고 시작을 선택하십시오.

결과

새 모니터는 CPU 활용을 표시하고 지정된 콜렉션 간격에 따라 매 30초마다 새 자료점이 추가됩니다. PC가 꺼져 있는 경우라도 CPU 활용이 80%에 도달하면 모니터는 자동으로 지정된 임계 조치를 수행합니다.

주: 이 모니터는 CPU 활용만 추적합니다. 그러나 원하는 만큼의 미터법을 같은 모니터에 포함시킬 수 있고 각 미터법은 자체의 임계 값과 조치를 가질 수 있습니다. 또한 여러 개의 시스템 모니터를 동시에 실행할 수도 있습니다.

시나리오: 작업 모니터:

상황

현재 iSeries 서버에서 새 어플리케이션을 실행하고 있고 새로운 대화식 작업의 일부가 허용할 수 없을 정도의 자원을 소비하고 있다는 점을 걱정하고 있습니다. 작업이 CPU 용량을 과도하게 소비하는 경우 이러한 작업의 소유자에게 통지를 하고자 할 수 있습니다.

새 어플리케이션의 작업을 감시하여 작업이 CPU 용량의 30% 이상을 소비하는 경우 메시지를 송신하도록 작업 모니터를 설정할 수 있습니다.

구성 예

작업 모니터를 설정하려면 감시할 대상 작업, 감시할 작업 속성 및 지정된 작업 속성이 감지될 경우 모니터가 수행해야 하는 작업을 정의해야 합니다. 이 목표를 달성하는 작업 모니터를 설정하려면 다음 단계를 완료하십시오.

1. iSeries Navigator에서 중앙 관리 > 모니터를 확장하고 작업 모니터를 마우스 오른쪽 버튼으로 클릭하여 새 모니터...를 선택하십시오.
2. 일반 페이지에서 다음 값을 입력하십시오.
 - a. 이 모니터의 이름과 설명을 지정하십시오.

- b. 모니터할 작업 탭에서 다음 값을 입력하십시오.
 - 1) 작업명에 감시하려는 작업의 이름을 지정하십시오(예: MKWIDGET).
 - 2) 서브시스템에 QINTER을 지정하십시오.
 - 3) 추가를 클릭하십시오.
3. 미터법 탭을 클릭하고 다음 정보를 입력하십시오.
 - a. 사용할 수 있는 미터법 리스트에서 요약 숫자값을 확장하고 CPU 활용 비율을 선택한 후 추가를 클릭하십시오.
 - b. 미터법 설정에 대한 임계값 1 탭에서 다음 값을 입력하십시오.
 - 1) 트리거 작동 기능을 선택하십시오.
 - 2) 임계 트리거 값에 ≥ 30 (30% 이상 사용)을 지정하십시오.
 - 3) 기간에 1 간격을 지정하십시오.
 - 4) OS/400 트리거 명령에 다음을 지정하십시오.
 SNDMSG MSG('사용자의 작업이 CPU 용량의 30%를 초과하고 있습니다') TOUSR(&OWNER)
 - 5) 재설정 작동 기능을 클릭하십시오.
 - 6) 임계 재설정 값에 < 20 (20% 미만 사용)을 지정하십시오.
4. 콜렉션 간격 탭을 클릭하고 15초를 선택하십시오. 이 값이 콜렉션 서비스 설정을 대체합니다.
5. 조치 탭에서 트리거와 재설정 열에서 이벤트 기록을 선택하십시오.
6. 서버와 그룹 탭을 클릭하고 이 작업에 대해 모니터하려는 서버와 그룹을 선택하십시오.
7. 확인을 클릭하여 새 모니터를 저장하십시오.
8. 작업 모니터의 리스트에서 새 모니터를 마우스 오른쪽 버튼으로 클릭하고 시작을 선택하십시오.

결과

새 모니터는 매 15초마다 QINTER 서브시스템을 검사하고 MKWIDGET 작업이 CPU의 30% 이상을 소비하는 경우 작업의 소유자에게 메시지를 송신합니다. 작업이 CPU 용량을 20%미만 사용하는 경우 모니터가 재설정됩니다.

시나리오: 메시지 모니터:

상황

회사에는 여러 개의 iSeries 서버가 실행되고 있고 각 시스템에 대해 메시지 대기행렬을 검사하는 것은 시간 낭비입니다. 시스템 관리자로서 사용자는 시스템에서 발생하는 조희 메시지에 대해 알아야 합니다.

iSeries 시스템에서 발생하는 메시지 대기행렬에 대해 조희 메시지를 표시하도록 메시지 모니터를 설정할 수 있습니다. 모니터가 열리고 감지된 메시지가 바로 표시됩니다.

구성 예

메세지 모니터를 설정하려면 감시하려는 메세지의 유형과 이러한 메세지가 발생할 경우 모니터가 수행할 작업을 정의해야 합니다. 이 목표를 달성하는 작업 모니터를 설정하려면 다음 단계를 완료하십시오.

1. iSeries Navigator에서 **중앙 관리 > 모니터**를 확장하고 **메세지 모니터**를 마우스 오른쪽 버튼으로 클릭하고 **새 모니터...**를 선택하십시오.
2. 일반 페이지에서 이 모니터의 이름과 설명을 입력하십시오.
3. 메세지 탭을 클릭하고 다음 값을 입력하십시오.
 - a. 모니터할 메세지 대기행렬에 **QSYSOPR**을 지정하십시오.
 - b. 메세지 세트 **1** 탭에서 유형에 대해 **조회**를 선택하고 **추가**를 클릭하십시오.
 - c. 다음 메세지 계수에서 **트리거**를 선택하고 **1** 메세지를 지정하십시오.
4. 콜렉션 간격 탭에서 **15초**를 선택하십시오.
5. 조치 탭을 클릭하고 **모니터 열기**를 선택하십시오.
6. 시스템 및 그룹 탭을 클릭하고 **조회 메세지**를 모니터할 시스템과 그룹을 선택하십시오.
7. **확인**을 클릭하여 새 모니터를 저장하십시오.
8. 메세지 모니터의 리스트에서 새 모니터를 마우스 오른쪽 버튼으로 클릭하고 **시작**을 선택하십시오.

결과

새 메세지 모니터가 QSYOPR로 송신된 조회 메세지를 모니터되는 임의의 iSeries 서버에 표시합니다.

주: 이 모니터는 QSYSOPR로 송신된 조회 메세지에만 응답합니다. 그러나 두 가지의 서로 다른 메세지 세트를 하나의 모니터에 포함시킬 수 있고 여러 개의 메세지 모니터를 동시에 실행할 수 있습니다. 또한 메세지 모니터는 지정된 메세지가 수신되면 OS/400 명령도 실행할 수 있습니다.

그래프 이력

그래프 이력은 콜렉션 서비스를 사용하여 몇 일, 몇 주, 몇 달 또는 몇 년간 수집된 성능 자료의 그래픽 보기를 제공합니다. 성능 자료를 보기 위해 시스템 모니터를 실행시킬 필요가 없습니다. 콜렉션 서비스를 사용하여 성능 자료를 수집하는 경우에는 그래프 이력 창을 볼 수 있습니다.

• 그래프 이력 개념

그래프 이력에 사용할 수 있는 이력 자료의 양은 콜렉션 서비스에서의 콜렉션 보유 값과, PM/400이 작동 가능한지의 여부에 크게 좌우됩니다. 성능 자료 레코드를 관리 및 표시하기 위해 사용할 수 있는 옵션에 대한 설명은 이 주제를 참조하십시오.

• 그래프 이력 사용

그래프 이력은 iSeries Navigator를 통해 액세스할 수 있습니다. 단계별 지침은 이 주제를 참조하십시오.

시스템 성능을 모니터링하는 데 대한 자세한 정보는 성능 자료 추적 주제를 참조하십시오.

그래프 이력 개념

그래프 이력은 콜렉션 서비스가 작성한 콜렉션 오브젝트에 들어 있는 자료를 표시합니다. 따라서 사용 가능한 자료의 유형과 양은 콜렉션 서비스의 구성에 따라 다릅니다.

사용자가 콜렉션 서비스 등록 정보에서 선택한 설정, 특히 콜렉션 보유 기간으로 그래프로 표시할 자료의 양이 결정됩니다. iSeries Navigator를 사용하여 여러 시스템에서 PM/400을 활성화할 수 있습니다. PM/400을 활성화할 때 그래프 이력 기능을 사용하여 몇 일, 몇 주 또는 몇 달 전에 수집된 자료를 볼 수 있습니다. 실시간 모니터 기능 이상을 수행할 수 있으며 요약 자료 또는 상세 자료에 액세스할 수 있습니다. PM/400이 작동 가능한 상태가 아니면 그래프 자료 필드는 1 - 7일까지만을 지원합니다. PM/400이 작동할 수 있는 상태에서 관리 콜렉션 오브젝트를 시스템에 보유할 기간을 정의합니다.

- 상세 자료

관리 콜렉션 오브젝트가 삭제되기 전에 시스템에 존재하는 시간 길이. 시간이나 일 수로 특정 시간 기간을 선택하거나 영구를 선택합니다. 영구를 선택하면 관리 콜렉션 오브젝트는 자동으로 삭제되지 않습니다.

- 그래프 자료

그래프 이력 창에 표시되는 상세 자료와 등록 정보 자료가 삭제되기 전에 시스템에 남아 있는 시간의 길이. PM/400을 시작하지 않으면 1 - 7일을 지정할 수 있습니다. PM/400을 시작하면 1 - 30일을 지정할 수 있습니다. 디폴트 값은 1시간입니다.

- 요약 자료

그래프의 자료 콜렉션 점이 삭제되기 전에 그래프 이력 창에 표시되거나 시스템에 남아 있는 시간 길이. 상세 자료나 등록 정보 자료는 사용할 수 없습니다. PM/400을 시작하여 요약 자료 필드를 작동 가능케 해야 합니다. 디폴트 값은 1달입니다.

그래프 이력 사용

그래프 이력은 iSeries Navigator에 들어 있습니다. 콜렉션 서비스를 사용하여 모니터링하고 있는 자료의 그래프 이력을 보려면 다음 단계를 수행하십시오.

1. 단일 시스템 또는 시스템 그룹에서 콜렉션 서비스를 시작하려면 iSeries Navigator의 온라인 도움말을 따라 수행하십시오.
2. 콜렉션 서비스 시작 - 일반 페이지에서 필요에 따라 **Performance Management/400** 시작을 선택합니다.
3. 콜렉션 보유 기간 값을 다른 값으로 변경합니다.
4. 확인을 클릭하십시오.
5. 시스템 모니터나 콜렉션 서비스 오브젝트를 마우스 오른쪽 버튼으로 누른 후 그래프 이력을 선택하면 그래프 이력을 볼 수 있습니다.
6. 화면정리를 클릭하여 그래픽 보기를 봅니다.

그래프 이력을 시작하면 그래프로 된 일련의 콜렉션 점이 창에 표시됩니다. 그래프 선의 이러한 콜렉션 점은 사용할 수 있는 세 가지 자료 레벨에 따라 세 개의 서로 다른 그래픽으로 표시됩니다.

- 사각형 콜렉션 점은 상세 정보와 등록 정보가 모두 포함되어 있는 자료를 나타냅니다.
- 삼각형 콜렉션 점은 요약 자료에 상세 정보가 포함됨을 나타냅니다.
- 원형 콜렉션 점은 상세 정보나 등록 정보가 포함되어 있지 않음을 나타냅니다.

Performance Management/400

Performance Management/400(PM/400)은 자동화 되고 자체 관리되어 사용이 용이합니다. PM/400은 콜렉션 서비스를 자동으로 트리거하여 서버로부터 비독점적인 성능 및 용량 자료를 수집하여 이 자료를 IBM으로 송신합니다. 모든 콜렉션 사이트는 네트워크 보안이 되어 있으며 전송 시간은 전적으로 사용자가 제어할 수 있습니다. 자료를 IBM으로 송신할 때 모든 경향 자료를 저장할 필요를 감소시킵니다. IBM은 사용자를 위해 자료를 저장했다가 사용자 서버의 확장 및 성능을 보여주는 일련의 보고서와 그래프를 사용자에게 제공합니다. 종래의 브라우저를 사용하여 보고서에 전자적으로 액세스할 수 있습니다. 이들 보고서는 중요한 성능 인디케이터를 지속적으로 분석함으로써 시스템 자원을 계획하고 관리하는 데 도움이 됩니다.

IBM Operational Support Services for PM/400e에는(성능 경향 분석을 사용하여) 현재의 어플리케이션 및 하드웨어 성능을 최대화하는 데 도움이 되는 보고서, 그래프 및 프로파일도 포함됩니다. 또한 이 제품은(용량 계획을 사용하여) 기업 활동 추이가 CPU나 디스크와 같은 필요한 하드웨어 업그레이드의 시점과 어떻게 연관되는지를 더 잘 이해하는 데에도 도움이 됩니다. 용량 계획 정보는 서버에 대한 조기 경보 시스템으로 생각할 수 있는 시스템 활용 자원 및 처리량 자료의 추이를 분석하여 얻어집니다. PM/400e를 시스템의 “건강” 상태를 알려주는 가상의 자원으로 생각하십시오.

PM/400은 중앙 처리 장치(CPU)의 1퍼센트 미만을 사용합니다. 디스크 공간은 약 58 MB를 사용하는데 이것은 하드웨어 모델과 콜렉션 간격에 따라 달라집니다.

PM/400 개념

PM/400이 제공할 수 있는 기능과 장점 및 중요한 구현 고려사항에 대해 배웁니다.

PM/400 구성

PM/400의 사용을 시작하려면, 이를 활성화하고 자료를 송신하고 보고서를 수신할 수 있는 전송 방법을 설정하며 자료 콜렉션과 기억장치를 사용자 정의해야 합니다.

PM/400

네트워크를 설정했으므로 PM/400을 사용하여 다양한 작업을 수행할 수 있습니다.

PM/400 보고서

PM/400을 사용하여 콜렉션 서비스 자료를 IBM에 바로 송신하도록 iSeries 서버를 구성할 수 있습니다. 자료를 수신한 IBM은 사용자가 웹에서 보거나 사용자에게 바로 송신할 수 있는 여러 가지 보고서를 생성합니다. 보고서를 자동으로 생성하도록 PM/400을 활성화하면 시간과 자원이 절약될 뿐 아니라 향후 성장을 예측하여 미리 계획을 세울 수도 있습니다.

PM/400 개념

PM/400은 콜렉션 서비스를 사용하여 서버로부터 비독점적인 성능 및 용량 자료를 수집하여 이 자료를 IBM으로 송신합니다. 이 정보에는 CPU 활용 및 디스크 용량, 응답 시간, 처리량, 어플리케이션 및 사용자 사용법 등이 포함될 수 있습니다. 자료를 IBM으로 송신할 때 모든 경향 자료를 저장할 필요를 감소시킵니다. IBM은 사용자를 위해 자료를 저장했다가 사용자 서버의 확장 및 성능을 보여주는 일련의 보고서와 그래프를 사용자에게 제공합니다. 종래의 브라우저를 사용하여 보고서에 전자적으로 액세스할 수 있습니다.

PM/400의 장점

PM/400은 시스템 자원 관리와 용량 계획을 아주 쉽게 만들어 줄 수 있습니다. PM/400을 활용하는 특정한 방법을 더 많이 배우십시오.

PM/400 제공 옵션

PM/400은 다양한 옵션을 제공합니다. 이 정보를 가지고 어떤 서비스의 조합이 사용자의 요구에 가장 잘 맞는지 결정하십시오.

자료 콜렉션 고려사항


PM/400은 콜렉션 서비스를 사용하여 성능 자료를 콜렉션합니다. PM/400과 콜렉션 서비스가 어떻게 작동하여 사용자에게 필요한 자료를 제공하는지를 배우십시오.

PM/400의 장점: PM/400 서비스를 사용하면 다음과 같은 이점이 있습니다.

- 예기치 못한 문제를 방지할 수 있습니다.
실망스러운 문제를 미연에 방지할 수 있습니다. 시스템의 확장 및 성능 관리를 제어합니다. 즉 사용자가 시스템을 관리합니다. 시스템이 사용자를 관리하는 것이 아닙니다.
- 시간을 절약합니다.
자동으로 실행함으로써 노동 집약적이고 비용이 많이 드는 성능 자료 수집 및 보고 작업을 감소시킬 수 있습니다. 이러한 장점으로 자원을 시스템 및 어플리케이션 관리에 집중시킬 수 있습니다.
- 최대의 효율을 위해 미래를 계획할 수 있습니다.
최대의 효율로 시스템이 실행될 수 있도록 재정적인 요구사항에 대해 능동적으로 계획할 수 있습니다.
- 이해하기 쉬운 정보를 제공합니다.
정보를 이해하기 쉽기 때문에 “업그레이드가 필요한 이유”에 대한 경영진의 질문에 답을 쉽게 제시할 수 있습니다.
- 미래를 예측할 수 있습니다.
사실적인 경향 정보에 근거하여 자료 처리 확장 예측을 할 수 있습니다.
- 시스템 문제를 식별할 수 있습니다.
PM/400 자료를 사용하여 성능 병목을 식별할 수 있습니다.
- 다음 업그레이드 규모를 예측할 수 있습니다.

PM/400 자료를 Workload Estimator for iSeries  로 업로드하여 다음 업그레이드의 규모를 예측할 수 있습니다.

PM/400을 사용하기 전에 해야 할 작업에 대한 내용은 Performance Management/400을 참조하십시오.

PM/400e에 대한 운영 지원 서비스: 그래프와 보고서를 전자적으로 또는 인쇄된 형태로 받아볼 수 있습니다. 한 달에 한번씩 전자 그래프를 수신할 수 있습니다. 인쇄된 그래프는 월별 또는 분기별로 받아볼 수 있습니다. PM/400e 서비스 요금은 성능 정보를 수신하는 빈도와 전자 형식인지 인쇄 형식인지에 따라 달라집니다. 일부 보고서 옵션은 무료이고 일부는 유료입니다. 해당 국가의 마케팅 및 서비스 조직에서 사용할 수 있는 지원에 대한 자세한 내용을 제공할 것입니다. 무료 옵션 및 유료 옵션에 대한 정보는 PM/400e 웹 사이트  를 참조하십시오.

PM/400을 사용하기 전에 해야 할 작업에 대한 내용은 Performance Management/400을 참조하십시오.

PM/400 자료 수집 고려사항: 시스템 활용, 작업부하 및 성능 측정의 정확한 경향을 설정하는 데 가장 중요한 요구사항은 일관성입니다. 이상적으로 성능 자료는 하루 24시간 수집되어야 합니다. PM/400과 콜렉션 서비스의 관계로 인해 PM/400 사용시 발생할 수 있는 잠재적인 사항을 알고 있어야 합니다.

다음은 PM/400을 사용할 때 콜렉션 정의에 도움이 되는 지침입니다.

- 자료를 저장할 **QMPGDATA** 라이브러리를 선택합니다.
콜렉션 저장 위치 필드는 PM/400이 사용 중인 경우 디폴트 값인 /QSYS.LIB/QMPGDATA.LIB를 사용합니다. QMPGDATA를 다른 값으로 대체하면 PM/400이 매 정시에 콜렉션을 순환하고 이를 다시 QMPGDATA로 되돌려 놓습니다. 자료를 다른 라이브러리로 수집하려면 PM/400이 자료를 탐색할 위치를 변경합니다. 명령행에서 **GO PM400**을 입력하고 옵션 3(사용자 정의에 대한 작업)을 선택한 후 라이브러리명을 변경합니다.
- 콜렉션 서비스를 사용하여 지속적으로 자료를 수집합니다
PM/400은 콜렉션 서비스를 사용하여 하루 24시간 자료를 수집함으로써 이 요구사항을 충족시킵니다. PM/400은 15분 간격으로 성능 자료를 수집합니다. PM/400은 디폴트로 15분 간격을 사용하지만 그 간격 설정값을 변경하지는 않습니다. 15분이 바람직한 간격입니다.
- **Standard** 플러스 프로토콜 프로파일을 선택합니다.
Standard 플러스 프로토콜은 콜렉션 프로파일의 디폴트 값입니다. 콜렉션 프로파일은 어떤 자료가 수집되는지를 나타냅니다. Standard 플러스 프로토콜 프로파일의 자료 범주는 STRPFRMON(성능 모니터 시작) 명령의 DATA 매개변수에 대한 *ALL 값에 대응됩니다. 이 값을 다른 값으로 변경하면 PM/400이 매 정시에 그 값을 다시 되돌려 놓습니다. 이것은 사용자 조정을 선택하고 모든 범주를 포함시켰을 때에도 해당됩니다. 변경된 사항은 즉시 효력을 발휘합니다. 콜렉션은 다른 이유로 반드시 그래야 하는 경우가 아니라면 순환하지 않습니다. 이 조치는 PM/400 보고서에 대한 충분한 정보를 수집하기 위해 수행됩니다.
- **PM/400**이 사용 중일 때 콜렉션 매개변수에 대한 중간 변경은 피합니다.
예를 들어 PM/400을 활성화할 때 콜렉션 중에 데이터베이스 파일 작성 필드는 디폴트 값으로 체크됩니다. 이 값을 변경하면 PM/400이 매 정시에 다시 디폴트 값으로 되돌려 놓습니다. 변경된 사항은 즉시 효력을 발휘합니다. 콜렉션은 다른 이유로 반드시 그래야 하는 경우가 아니라면 순환하지 않습니다.
- 콜렉션 서비스 종료
언제든지 iSeries Navigator에서 콜렉션 서비스를 종료할 수 있습니다. 콜렉션 서비스를 종료하는 경우 PM/400이 실행 중일 때 다음과 같은 고려사항이 적용됩니다.
 - PM/400 스케줄러는 다음에 돌아오는 정시에 콜렉션 서비스를 시작합니다.
 - 자료가 수집되지 않았거나 거의 수집되지 않은 날은 경향 산출에 포함되지 않습니다. 그러므로 콜렉션 서비스를 자주 인터럽트해서는 안됩니다.

콜렉션 서비스를 시작하지 않으려면 PM/400을 일시 중지할 수 있습니다.

PM/400 구성

PM/400은 콜렉션 서비스를 통해 성능 자료 수집을 자동화합니다. 라이브러리가 보조 기억장치 풀(ASP)에 상주하는 경우 자료를 담은 라이브러리를 지정할 수 있습니다. 독립 보조 기억장치 풀은 PM/400 콜렉션 프로세

스를 중단하는 단절변환이 가능하므로 라이브러리를 독립 보조 기억장치 폴로 이동하면 안됩니다. 라이브러리가 아직 없는 경우 PM/400은 활성화 동안 라이브러리를 작성합니다.

PM/400의 사용을 시작하려면 다음 타스크를 수행해야 합니다.

PM/400 활성화

PM/400에는 OS/400이 들어 있습니다. 이의 수집 기능을 사용하려면 PM/400을 활성화해야 합니다.

사용할 전송 방법 결정

자료 송신 방법을 결정합니다. 중앙 관리를 사용하여 자료를 수집하여 전자 서비스 에이전트(극단적 지원)를 사용하여 자료를 송신하거나, PM/400으로 자료를 수집하고 SNA 프로토콜을 통해 송신할 수 있습니다.

PM/400 사용자 정의

네트워크를 설정했으므로 사용자의 요구에 맞게 PM/400을 사용자 정의합니다.

Performance Management/400 활성화: 자료 수집 기능을 활용하려면 먼저 PM/400을 시작합니다. 다음 방법 중 하나를 사용하여 PM/400을 시작할 수 있습니다.

iSeries Navigator 사용

여러 시스템에서 PM/400을 활성화하려면 iSeries Navigator를 사용하십시오. PM/400을 활성화할 때 그래프 이력 기능을 사용하여 몇 일, 몇 주 또는 몇 달전에 수집된 자료를 볼 수 있습니다. 실시간 모니터 기능 이상을 수행할 수 있습니다. 요약 자료나 상세 자료에 액세스할 수 있습니다. PM/400이 작동가능한 상태가 아니면 그래프 자료 필드는 1 - 7일까지만을 지원합니다. PM/400이 작동가능한 상태가 되면 자료를 보유할 수 있는 시간의 길이를 선택할 수 있습니다.

iSeries Navigator에서 PM/400을 시작하려면 다음 단계를 수행하십시오.

1. iSeries Navigator에서 PM/400을 시작하려는 시스템을 확장합니다.
2. 구성 및 서비스를 확장하십시오.
3. 콜렉션 서비스를 마우스 오른쪽 버튼으로 클릭합니다.
4. **Performance Management/400**을 선택합니다.
5. 시작을 선택합니다.
6. PM/400을 시작할 시스템을 선택합니다.
7. 확인을 클릭하십시오.

QSYSOPR 메시지 대기행렬의 메시지 CPAB02A에 응답

QSYSWRK 서브시스템이 시작되면 이 메시지는 PM/400을 활성화할 것인지 묻습니다.

1. 문자 기반의 인터페이스에서 QSYSOPR의 메시지 “PM/400을 활성화하겠습니까?(I G C)”에 G로 응답하십시오. QSYSOPR 메시지 대기행렬은 PM/400이 활성화되었다는 메시지를 수신합니다.
2. 문의처 정보를 갱신합니다. **GO PM400** 명령을 발행하고 옵션 1을 지정하십시오.

CFGPM400(PM/400 구성) 명령 발행

문자 기반 인터페이스에서 CFGPM400(PM/400 구성) 명령을 발행할 수 있습니다.

이제 설정 프로세스에서 IBM으로 자료를 송신하는 데 사용할 전송 방법을 판별하는 다음 단계로 진행할 수 있습니다.

iSeries 성능에 대한 개요는 성능을 참조하십시오.

사용할 PM/400 전송 방법 결정: V5R1부터 PM/400 전송 프로세스는 중앙 관리를 사용하여 수행하는 네트워크 구성의 이점을 활용하여 중앙 시스템과 종료점 시스템을 설정합니다. 그러나 문자 기반 인터페이스를 사용해서 PM/400을 구성할 수도 있습니다. 사용할 전송 방법을 선택하십시오.

- Extreme Support를 통해 Electronic Service Agent를 사용한 자료 송신
이 전송 방법을 선택하면 중앙 관리에 의해 자료가 수집되도록 PM/400을 구성해야 합니다. 서버에 설치되어 있는 오퍼레이팅 시스템이 V4R5이거나 그 이후 릴리스인 경우 이 구성을 수행하십시오(Universal Connection 수정 프로그램은 적용해야 합니다). Extreme Support를 사용하는 경우 이 방법을 선택합니다.
- SNA 프로토콜을 사용하여 자료 송신
이 전송 방법을 선택하면 문자 기반 인터페이스를 사용하여 PM/400을 구성해야 합니다. PM/400은 자료를 수집한 후 SNA를 사용하여 전송합니다. 서버에 OS/400 V4R5 또는 그 이전 릴리스가 설치되어 있다면 PM/400에 대해 이 구성을 수행하십시오.

사용할 전송 방법을 구현하였으면 PM/400 관리를 위한 다른 작업을 수행할 준비가 된 것입니다.

Extreme Support를 통해 Service Agent를 사용한 PM/400 자료 송신 (Universal Connection):

PM/400은 콜렉션 서비스를 사용하여 서버로부터 비독점적인 성능 및 용량 자료를 수집합니다. 자료를 수집한 후 Extreme Support를 통해 Service Agent를 사용하여 이 자료를 IBM으로 송신할 수 있습니다.

이 기능을 활용하려면 V5R1 또는 V5R2가 서버에 설치되어 있거나 V4R5에 Universal Connection 수정 프로그램을 적용해야 합니다. 다음은 PM/400에 대한 구성 단계입니다.

1. PM/400 활성화
자료 수집 기능을 활용하려면 먼저 PM/400을 시작합니다.
2. 중앙 관리 네트워크 설정
어떤 서버가 중앙 시스템이고 어떤 서버가 종료점 시스템인지 정의합니다. 이 네트워크 계층을 사용하여 자료를 IBM으로 송신하기 전에 종료점 시스템에서 중앙 위치로 자료를 송신할 수 있습니다.
3. IBM에 연결하여 Universal Connection으로 자료 송신.
이것은 중앙 관리가 PM/400 자료를 IBM으로 송신하기 위해 사용하는 연결입니다. 이전 릴리스에서는 SNA를 통한 전자 고객 지원(ECS) 연결을 사용했습니다. Universal Connection을 사용할 때 TCP/IP를 통해 자료를 전송할 수 있습니다.
4. PM/400 성능 자료 수집.
중앙 관리의 명세 기능을 사용하여 자료를 수집합니다.

5. IBM에 자료 송신.

중앙 관리 계층에서 Extreme Support 아래에 있는 Electronic Service Agent를 사용하여 자료를 IBM으로 송신합니다. Electronic Service Agent는 Universal Connection을 사용합니다.

또한 SNA 프로토콜을 사용하여 자료를 송신할 수도 있습니다.

PM/400 구성이 완료되었다면 이제 PM/400 관리를 위한 다른 작업을 수행할 준비가 된 것입니다.

PM/400 성능 자료 수집: 다음과 같은 작업을 수행하였다면 중앙 관리를 사용하여 PM/400 성능 자료를 수집할 수 있습니다.

1. PM/400 활성화
2. 범용 연결 구성
3. 중앙 관리 네트워크 설정
4. 전자 서비스 에이전트가 시스템에 설치되어 있는지 확인

종료점 시스템이나 시스템 그룹에서 PM/400 성능 자료를 수집하려면 다음 단계를 수행하십시오.

1. iSeries Navigator에서 중앙 관리를 확장하십시오.
2. 종료점 시스템 또는 시스템 그룹을 확장하십시오.
3. 종료점 시스템이나 시스템 그룹을 마우스 오른쪽 버튼으로 클릭하고 명세를 선택하십시오.
4. 수집을 선택하십시오.
5. 수집한 명세를 하나 이상 선택하십시오. 여기에서는 **PM/400 성능 자료**를 선택합니다.
6. 컬렉션이 완료된 후 중앙 시스템에서 어떤 조치가 실행되도록 하려면 리스트에서 해당 조치를 선택합니다.
7. 확인을 눌러 즉시 자료 수집을 시작하거나 스케줄을 눌러 자료 수집 시점을 지정합니다.

서버 구성을 완료하였으면 PM/400 관리를 위한 다른 작업을 수행할 준비가 된 것입니다.

SNA 프로토콜을 사용하여 자료 송신: Extreme Support를 통해 Electronic Service Agent를 사용한 자료 송신을 활용하지 않는 경우에도 문자 기반의 인터페이스를 사용하여 자료를 전송할 수 있습니다. PM/400은 서버의 구성 및 사용에 대한 몇 가지 질문을 할 것입니다. PM/400 구성 화면에서 서버가 PM/400 성능 자료를 송수신하는 방법에 대하여 사용자에게 질문할 것입니다. 이 프로세스의 처음 부분은 네트워크를 설정하는 것입니다. 두 번째는 자료 전송 방법에 대해 물을 것입니다. 문자 기반의 인터페이스를 사용하는 경우 직접 다이얼 회선을 사용하여 자료를 전송할 수 있습니다.

SNA를 사용하여 자료를 송신하려면 다음 단계를 수행하십시오.

1. PM/400 활성화

자료 수집 기능을 활용하려면 먼저 PM/400을 시작합니다.

2. 어떤 네트워크 구성을 사용할 것인지 선택합니다.

자료를 전송할 때 어떤 네트워크 구성을 사용할 것인지 결정합니다. 직접 다이얼 회선, 기존의 인터넷 서비스 제공자(ISP) 또는 VPN(가상 사설망) 중 IBM에 연결할 때 사용할 방법을 선택합니다. ISP 또는 VPN을 사용하려면 범용 연결을 구성해야 합니다.

직접 다이얼 회선을 사용하여 자료를 IBM으로 보고하려는 경우에는 네트워크 구성 방법으로 몇 가지를 선택할 수 있습니다. 사용자의 네트워크에 적절한 구성을 선택하고 PM/400 구성 화면에서 해당 구성에 대하여 설명되어 있는 단계를 수행합니다.

- 자료를 직접 IBM으로 송신하는 단일 서버.
- 다른 서버(리모트 서버)로부터 성능 자료를 수신한 후 이 자료를 IBM으로 송신하는 호스트 서버. 호스트 서버는 다른 서버보다 이전 릴리스 레벨이어서는 안됩니다. 즉 호스트 서버는 다른 서버와 같은 릴리스 레벨이거나 그 이후의 릴리스 레벨이어야 합니다.
- 성능 자료를 호스트 서버로 송신할 수 있는 리모트 서버. PM/400 구성 화면을 통해 리모트 서버가 필요하다라는 것을 확인한 후 PM/400 메뉴에서 옵션 5(리모트 iSeries 시스템에 대한 작업)를 사용하여 리모트 서버를 정의합니다.

3. 리모트 서버에 대한 작업

호스트 서버용으로 네트워크를 설정하기로 한 경우, 자료를 사용자의 호스트 서버로 송신할 서버를 식별해야 합니다. 단일 서버나 리모트 서버를 사용하고 있다면 이 단계를 무시해도 좋습니다.

4. PM/400 사용자 정의

네트워크를 구성한 후 PM/400 소프트웨어가 수행되기 위해 글로벌 매개변수를 설정해야 합니다. 직접 다이얼 회선을 사용하여 IBM에 연결하려면 PM/400 자료 전화 번호를 정의해야 합니다.

서버 구성이 완료되었다면 이제 PM/400 관리를 위한 다른 작업을 수행할 준비가 된 것입니다.

단일 서버용 PM/400 네트워크: 단일 서버는 자료를 직접 IBM을 송신합니다. 다음에서는 PM/400이 SNA를 통해 자료를 수집하고 전송하는 경우에만 해당되는 단일 서버용 PM/400 구성 단계를 설명합니다. 서버의 PM/400 구성(CFGPM400) 화면에서 다음을 수행하십시오.

1. 명령행에서 **CFGPM400**를 입력합니다.
2. **IBM**으로 성능 자료 송신 필드에 ***YES**를 지정합니다.
3. 성능 자료 수신 필드에 ***NO**를 지정합니다.
4. 디폴트 라이브러리 **QMPGDATA**를 받아들입니다.
5. **IBM**으로 성능 자료 송신 필드에 ***YES**를 지정한 경우 적절한 통신 오브젝트가 존재하는지 여부를 나타내는 추가 정보가 표시됩니다. 오브젝트가 없을 경우 PM/400은 전송을 위한 통신 오브젝트를 작성합니다. 추가 화면에 적절히 응답하십시오.
6. 문의처 정보에 대한 작업 화면에서 회사의 문의처 정보를 입력합니다.

단일 서버 설정을 원하지 않는 경우 다른 SNA 구성 옵션을 선택할 수 있습니다.

서버 구성을 완료하였으면 PM/400 관리를 위한 다른 작업을 수행할 준비가 된 것입니다.

호스트 서버용 PM/400 네트워크: 호스트 서버는 다른 서버로부터 성능 자료를 수신하여 이 자료를 IBM으로 다시 전송합니다. 다음에서는 PM/400이 SNA를 통해 자료를 수집하고 전송하는 경우에만 해당되는 호스트 서버용 PM/400 구성 단계를 설명합니다.

1. **호스트** 서버의 PM/400 구성 화면에서 다음을 수행하십시오.

- 명령행에서 **CFGPM400**를 입력합니다.
- **IBM**으로 성능 자료 송신 필드에 *YES를 지정합니다.
- 성능 자료 수신 필드에 *YES를 지정합니다.
- 디폴트 라이브러리 QMPGDATA를 받아들입니다.

2. 호스트 서버의 리모트 iSeries 시스템에 대한 작업 화면에서 다음을 수행합니다.

- F6(작성)을 눌러 어떤 서버가 호스트 서버로 자료를 송신할 것인지를 식별합니다.
- 필드 입력을 완료하고 Enter를 누릅니다.

주: 다음은 PM/400이 자료를 수집하여 SNA를 통해 전송하는 경우에만 해당됩니다. 시스템의 네트워크가 있는 경우 iSeries Navigator의 범용 연결과 중앙 관리 기능을 사용하여 자료를 수집한 후 이들 시스템에 자료를 전송하는 것이 좋습니다.

PM/400은 리모트 서버로부터 자료를 수신한 다음 날 1차 서버에서 IBM으로의 자료 전송을 자동으로 스케줄합니다. 자동 스케줄링 관리가 사용자의 작업 관리 체계에 적절하지 않다면 1차 서버로부터의 자료 전송을 수작업으로 스케줄할 수 있습니다.

다음은 자료 전송을 스케줄할 때 명심해야 할 추가 정보입니다. 1주일 동안 1차 서버로의 자료 전송을 고르게 스케줄링 하십시오. 이렇게 하면 1차 서버의 성능에 미치는 영향을 최소화할 수 있습니다. 예를 들어, 12개의 서버로 구성된 네트워크에서는 4개의 시스템으로 구성된 3개의 그룹을 구성할 수 있습니다. 각 그룹이 각각의 자료를 월요일, 수요일, 금요일에 송신하도록 스케줄링할 수 있습니다. 이렇게 하면 1차 서버로 송신되는 자료의 양을 분산시킬 수 있습니다.

호스트 서버 설정을 원하지 않는 경우 다른 SNA 구성 옵션을 선택할 수 있습니다.

서버 구성을 완료하였으면 PM/400 관리를 위한 다른 타스크를 수행할 준비가 된 것입니다.

리모트 서버용 PM/400 네트워크: 리모트 서버는 성능 자료를 호스트 서버로 송신합니다. 다음에서는 PM/400이 SNA를 통해 자료를 수집하고 전송하는 경우에만 해당되는 리모트 서버용 PM/400 구성 단계를 설명합니다. 리모트 서버의 PM/400 구성 화면(CFGPM400)에서 다음 단계를 수행하십시오.

1. 명령행에서 **CFGPM400**를 입력합니다.
2. **IBM**으로 성능 자료 송신 필드에 *NO를 지정합니다.
3. 성능 자료 수신 필드에 *NO를 지정합니다.
4. 디폴트 라이브러리 QMPGDATA를 받아들입니다.

주: 시스템의 네트워크가 있는 경우 iSeries Navigator의 명세 기능을 사용하여 자료를 수집한 후 범용 연결을 통해 이들 시스템에 자료를 전송하는 것이 좋습니다.

리모트 서버 설정을 원하지 않는 경우 다른 SNA 구성 옵션을 선택할 수 있습니다.

서버 구성을 완료하였으면 PM/400 관리를 위한 다른 타스크를 수행할 준비가 된 것입니다.

리모트 서버에 대한 작업: 일부 사이트에서는 네트워크의 호스트 서버가 처리를 위해 필요한 성능 자료를 IBM으로 송신합니다. 호스트 서버 네트워크를 사용하는 경우 네트워크의 다른 서버들이 이 호스트 서버로 성능 자료를 송신하여 IBM으로 전송하도록 할 수 있습니다. 호스트 서버를 사용할 수 있도록 네트워크를 설정하려면 다른 리모트 서버를 식별하고 이 리모트 서버들의 자료 전송 스케줄을 설정해야 합니다. 리모트 iSeries 시스템에 대한 작업 화면에서 다른 서버를 작성할 수 있습니다.

주:

1. 네트워크를 리모트 서버나 단일 서버로 설정하는 경우에는 이 화면을 사용할 필요가 없습니다. PM/400이 자료를 수집하여 SNA를 통해 전송하는 경우에만 이 작업을 수행합니다.
2. 시스템의 네트워크가 있는 경우 iSeries Navigator의 명세 기능을 사용하여 자료를 수집한 후 범용 연결을 통해 이들 시스템에 자료를 전송하는 것이 좋습니다.

리모트 서버를 정의하려면 다음 단계를 수행하십시오.

1. 명령행에서 **GO PM400**을 입력합니다.
2. PM/400 메뉴에서 5(리모트 iSeries 시스템에 대한 작업)를 입력하고 Enter를 누릅니다. 처음에는 리모트 서버가 표시되지 않습니다. 새로운 리모트 위치를 작성해야 합니다.
3. F6(작성)을 눌러 새로운 리모트 위치를 작성합니다.
4. 다음 정보에 대한 값을 기록합니다. DSPNETA(네트워크 속성 표시) 명령을 사용하여 리모트 시스템으로부터 이 값들을 표시합니다.
 - 로컬 네트워크 ID
 - 디폴트 로컬 위치

리모트 iSeries 시스템에 대한 작업 화면에는 리모트 서버 리스트가 표시됩니다. 이 리스트에는 서버의 상태(활동 또는 비활동)와 각 서버에 대한 설명도 포함됩니다.

5. PM/400 리모트 사이트 유지보수 화면이나 리모트 사이트 iSeries 변경 화면을 사용하여 리모트 사이트 서버에 대한 설명을 작성 또는 변경할 수 있습니다. 리모트 위치명은 리모트 서버 사이에서 고유해야 합니다.

PM/400은 리모트 서버로부터 자료를 수신한 다음 날 1차 서버에서 IBM으로의 자료 전송을 자동으로 스케줄합니다. 자동 스케줄링 관리가 사용자의 작업 관리 체계에 적절하지 않다면 1차 서버로부터의 자료 전송을 수작업으로 스케줄할 수 있습니다. 자료 전송을 수작업으로 스케줄링하려면 PM/400 스케줄러를 참조하십시오.

PM/400 소프트웨어는 자료를 수신하는 서버(호스트 서버)와 자료를 송신하는 서버(리모트 서버)간에 확장 대 등 시스템간 통신 기능(APPN)이 정의되어 있는 것으로 가정합니다. 시스템의 시스템 값 QCRTAUT(디폴트 공용 권한 작성)가 *EXCLUDE 또는 *USE로 설정된 경우, 제어기 설명을 정의하는 방법에 대한 설명은 리모트 서버에 대한 장치 설명 작성을 참조하십시오. 네트워크가 이 전제를 충족시키지 못하는 경우 각 리모트 서버로의 연결을 지원하기 위한 장치 쌍을 작성하는 데 대한 정보는 비APPN 네트워크 고려사항을 참조하십시오.

리모트 서버를 정의하였으면 특정 회선 연결을 사용하도록 PM/400을 사용자 정의할 준비가 된 것입니다.

비APPN 네트워크에서 리모트 서버에 대한 작업: 1차 서버는 다른 서버로부터 PM/400 자료를 수신하여 이 자료를 IBM으로 송신합니다. 리모트 서버는 PM/400 자료를 1차 서버로 송신합니다. 다음에서는 참조되는 제어기가 이전에 미리 정의되어 있는 것으로 가정합니다.

PM/400이 자료를 수집하여 SNA를 통해 전송하는 경우에만 각 리모트 서버로의 연결을 지원할 장치 쌍을 작성해야 합니다.

1. CRTDEVAPPC(장치 설명 작성(APPC)) 명령을 사용합니다. 리모트 서버에서 CRTDEVAPPC라고 입력합니다. F4를 눌러 매개변수를 프롬프트하고 다음 정보를 사용하여 해당 값을 정의합니다.

리모트 시스템

DEV(D(Q1PLOC)	장치 설명 이름을 정의합니다.
RMTLOCNAME(Q1PLOC)	리모트 위치명을 지정합니다.
ONLINE(*YES)	이 장치가 시스템이 시작 또는 재시작할 때 온라인으로 변환할 것인지를 지정합니다.
LCLLOCNAME(Q1PRMxxx)	로컬 위치명을 지정합니다. Q1PRMxxx는 1차 서버의 RMTLOCNAME과 일치하는데, xxx는 각 리모트 위치마다 고유해야 합니다.
CTL(yyyyyy)	접속된 제어기명을 지정하는데 yyyyyy는 1차 서버에 접속된 제어기입니다.
MODE(Q1PMOD)	모드명을 지정합니다.
APPN(*NO)	장치의 APPN 가능 여부를 지정합니다.

2. 1차 서버에서 다음 정보를 지정합니다. 명령행에서 CRTDEVAPPC를 입력합니다. F4를 눌러 매개변수를 프롬프트하고 다음 정보를 사용하여 해당 값을 정의합니다.

1차 서버

DEV(D(Q1PRMxxx)	장치 설명 이름을 정의합니다. 여기에서 사용된 이름은 리모트 시스템의 장치 설명 이름과 일치해야 합니다.
RMTLOCNAME(Q1PRMxxx)	리모트 위치명을 지정합니다. 여기에서 사용된 이름은 리모트 서버의 LCLLOCNAME 값과 일치해야 하며 xxx는 각 리모트 위치에 고유해야 합니다.
ONLINE(*YES)	이 장치가 시스템이 시작 또는 재시작할 때 온라인으로 변환할 것인지를 지정합니다.
LCLLOCNAME(Q1PLOC)	로컬 위치명을 지정합니다. 이 값은 리모트 서버의 RMTLOCNAME과 일치합니다.
CTL(aaaaaa)	접속된 제어기명을 지정하는데 aaaaaa는 리모트 서버에 접속된 제어기입니다.

MODE(Q1PMOD) 모드명을 지정합니다.

APPN(*NO) 장치의 APPN 기능 여부를 지정합니다.

3. APPC 장치를 정의한 후 장치를 연결변환(VRYCFG(구성 변환) 명령)합니다. 리모트 서버에서 VRYCFG 라고 입력합니다. F4를 눌러 매개변수를 프롬프트합니다.

리모트 시스템 연결변환

CFGOBJ(Q1PLOC) 구성 오브젝트를 지정합니다.

CFGTYPE(*DEV) 구성 오브젝트 유형을 지정합니다.

STATUS(*ON) 상태를 지정합니다.

4. PM/400 메뉴에서 옵션 5를 입력하고 리모트 서버로 Q1PRMxxx를 추가합니다. 리모트 서버를 추가하는 방법은 리모트 서버에 대한 작업을 참조하십시오.

이제 PM/400 구성이 완료되었습니다. PM/400을 사용하여 수행할 수 있는 다른 타스크는 PM/400 관리를 참조하십시오.

PM/400의 장치 설명 작성: QCRTAUT(디폴트 공용 권한 작성) 시스템 값이 *EXCLUDE나 *USE로 설정되어 있는 각 리모트 서버의 경우 다음 단계가 필요합니다. QUSER가 장치 설명 Q1PLOC에 대하여 *CHANGE 권한이 없는 경우 리모트 전송이 실패합니다. 이 단계에서는 이 장치가 작성되지 않거나 자동으로 삭제되도록 합니다.

주: 이 타스크는 PM/400이 자료를 수집하여 SNA를 통해 전송하는 경우에만 필요합니다.

이 장치를 자동으로 작성하려는 경우 QCRTAUT 값에 따라 장치 설명은 PUBLIC *EXCLUDE나 *USE 권한을 갖고 작성됩니다. 장치의 자동 작성 또는 삭제 여부는 제어기에 의해 제어됩니다. 이 매개변수들이 시스템에서 정의되는 방법을 판별하려면 다음 명령을 참조하십시오.

- CRTCTLAPPC(제어기 설명 작성(APPC)) 명령
- CHGCTLAPPC(제어기 설명 변경(APPC)) 명령
- DSPCTLD(제어기 설명 표시) 명령

APPN을 사용하도록 구성되지 않은 시스템의 경우 장치 설명을 작성하는 방법에 대한 설명은 비APPN 환경에서 리모트 서버에 대한 작업을 참조하십시오.

다음에서는 호스트 서버와의 통신에 사용될 제어기가 이전에 리모트 서버에서 정의되었다고 가정합니다.

리모트 서버에서 장치 설명 Q1PLOC를 다시 작성합니다.

```

VRYCFG      CFGOBJ(Q1PLOC)
             CFGTYPE(*DEV)
             STATUS(*OFF)

```

DLTDEV D DEVD(Q1PLOC)

CRTDEVAPPC DEVD(Q1PLOC)
RMTLOCNAME(Q1PLOC)
ONLINE(*NO)
LCLLOCNAME(name of remote system)
RMTNETID(remote netid of primary(or central) system)
CTL(name of controller that the device will be attached to)
AUT(*EXCLUDE)

CRTOBJAUT OBJ(Q1PLOC)
OBJTYPE(*DEV)
USER(QUSER)
AUT(*CHANGE)

VRRCFG CFGOBJ(Q1PLOC)
CFGTYPE(*DEV)
STATUS(*ON)

이제 PM/400 구성이 완료되었습니다. PM/400을 사용하여 수행할 수 있는 다른 타스크는 PM/400 관리를 참조하십시오.

PM/400 사용자 정의: PM/400 사용자 정의에 대한 작업 화면에서는 다음과 같은 기능을 제공합니다.

PM/400 소프트웨어의 조작을 위한 글로벌 매개변수 설정

글로벌 매개변수를 사용하여 다음 항목을 사용자 정의할 수 있습니다. 이 필드들에 대한 설명은 온라인 도움말을 참조합니다.

- 우선순위 제한
- 경향 및 시프트 스케줄
- 성능 자료 라이브러리
- 제거 스펙

PM/400 자료 전화 번호 정의

미국과 캐나다 이외의 지역에서는 사용자의 자료를 수신할 IBM의 전화 번호를 PM/400에 제공해야 합니다. 대부분의 위치에서 PM/400 프로세스 구성을 시작하면 PM/400이 해당 위치에 대한 올바른 자료 전화 번호를 선택해 줍니다.

PM/400을 사용한 연결변환 및 단절변환

PM/400 회선 제어 화면에서 PM/400은 회선을 단절변환하고 PM/400 자료를 전송한 후 회선을 다시 연결 지연 상태에 놓습니다.

글로벌 매개변수를 사용자 정의하려면 다음 단계를 수행하십시오.

1. 명령행에서 **GO PM400**을 입력합니다.

2. PM/400 메뉴에서 3을 눌러 PM/400 사용자 정의에 대한 작업 화면을 표시하고 Enter를 누릅니다.

콜렉션 서비스를 사용하여 PM/400 자료를 수집하는 경우 PM/400에 대한 자료 콜렉션 고려사항을 고려해야 합니다.

PM/400을 사용하여 수행할 수 있는 다른 task에 대한 내용은 PM/400 관리를 참조하십시오.

PM/400 자료 번호 확인: 서버가 IBM과 직접 다이얼 연결을 사용하고 있는 경우 PM/400 전화 번호가 올바른지 확인해야 합니다. 전화 번호에는 사용자의 회선에 맞는 올바른 접두부가 포함되어야 합니다.

주: 이것은 SNA 전송에 한합니다.

전자 고객 지원 회선의 전화 번호 형식을 확인하려면 다음 단계를 수행하십시오.

1.

DSPDTAARA DTAARA(QUSRSYS/QESTELE)

를 입력하고 Enter를 누릅니다.

2. 오프셋 0에 있는 연결 번호 접두부를 판별합니다. 예를 들어 오프셋 0이 'T9:1800xxxxxxx'이면 접두부는 T9:입니다.

3.

DSPDTAARA DTAARA(QUSRSYS/Q1PGTELE)

를 입력하고 Enter를 누릅니다.

4. 오프셋 0(zero)은 사용될 다이얼링 스트링입니다. (다른 번호는 사용되지 않습니다.)

5. ECS 회선을 PTF 주문에 사용하는 경우 오프셋 0의 형식을 ECS 회선 CALL QESPHONE에 사용된 형식과 비교하고 사용되고 있는 문자열을 기록한 후 이를 단계 2에서 확인한 값과 비교합니다.

전화 번호는 다르지만 접두부는 같아야 합니다(즉 SST9:1800..., SST:1800... 등).

사용자의 전화 번호를 변경해야 하는 경우 CHGDTAARA(자료 영역 변경) 명령을 사용하십시오.

CHGDTAARA를 입력하십시오. 여기서 DTAARA는 Q1PGTELE이고 LIB는 QUSRSYS이며 서브스트링 시작 위치는 *ALL이며 새 값은 'SST:18005475497'입니다.

주: 새로운 값은 사용자의 다이얼링 접두부에 18005475497(미국과 캐나다의 경우)가 후속되는 값이어야 합니다.

이제 PM/400 구성을 완료하였습니다. 수행할 수 있는 task에 대한 내용은 PM/400 관리를 참조하십시오.

PM/400용 직접 다이얼 회선 설정: 대부분의 위치에서 PM/400이 해당 위치에 대한 올바른 자료 전화 번호를 선택해 줍니다. 항상 PM/400 자료 전화 번호가 올바른지 확인해야 합니다. PM/400 자료 전화 번호와 PM/400 지원 번호가 들어 있는 정보가 없는 경우에는 해당 지역의 IBM 지원 담당자에게 문의하십시오. 적절한 전화 번호를 받을 수 있습니다.

주: 이 전화 번호는 범용 연결을 통해 자료를 전송할 때는 필요하지 않습니다. 직접 다이얼 회선을 사용하는 경우에만 이 번호가 필요합니다.

PM/400 자료 전화 번호를 정의하거나 변경하려면 다음 단계를 수행하십시오.

1. 명령행에서 **GO PM400**을 입력합니다.
2. PM/400 메뉴에서 3을 눌러 PM/400 사용자 정의에 대한 작업 화면을 표시하고 Enter를 누릅니다.
3. 이 화면에서 전화번호 필드가 있는 부분이 나타날 때까지 화면을 이동하십시오.
4. **IBM PM/400** 전화 번호 필드에 올바른 다이얼링 순서를 입력합니다. 많은 IBM 모뎀에서 다이얼 톤에 콜론(:)을 사용해야 합니다.

PM/400을 사용한 회선 연결변환 및 단절변환: 때로 PM/400이 사용하는 회선이 연결 지연 상태에 있는 경우가 있습니다. 이 상태에서는 PM/400이 자료 전송을 위해 회선에 액세스할 수 없습니다. PM/400 회선 제어 화면에서 PM/400은 회선을 단절변환하고, 자료를 전송한 후 회선을 다시 연결 지연 상태에 놓습니다. 이 화면을 사용할 때 PM/400 전송 태스크(Q1PCM1)를 사용하여 회선 상태를 확인하고 적절한 회선을 단절변환할 수 있습니다. 전송이 일단 완료되면 같은 회선이 연결 지연 상태에 놓이게 됩니다.

주: 이 태스크는 PM/400이 자료를 수집하여 SNA를 통해 전송하는 경우에만 필요합니다.

회선을 연결변환 및 단절변환하려는 경우 다음 단계를 수행하십시오.

1. 명령행에서 **PMLINMON**을 입력하여 PM/400 회선 모니터링 기능을 시작합니다. PM/400 회선 제어 화면을 보아야 합니다.
2. 첫 번째 화면에 표시된 경고를 읽고 Enter를 누릅니다.
3. PM/400이 단절변환할 회선, 제어기 및 장치의 조합을 정의합니다.
4. 프롬프트 **PM/400** 자동 회선 제어를 활성화하시겠습니까?를 기능에 대한 마스터 제어 스위치로 사용합니다. 예를 지정하면 PM/400 기능이 사용 중이 됩니다. 아니오를 지정하면 기능이 작동 불가능이 됩니다. 아니오를 지정하면 예를 지정할 때 회선 제어 리스트를 다시 정의할 필요가 없습니다. 회선만 지정하여 회선을 연결변환 및 단절변환할 수 있습니다. 세가지 설명을 모두 지정하여 회선, 제어기 및 장치를 연결변환 및 단절변환할 수 있습니다.
5. 정의한 회선, 제어기 및 장치를 검증합니다. Enter를 눌러 선택사항 요약을 검토합니다.
6. Enter를 눌러 선택한 내용을 확정하거나 F12를 눌러 이전 화면으로 돌아가서 항목을 변경합니다.

CFGPM400(PM/400 구성) 명령을 사용하여 PM/400 회선 제어를 설정할 수도 있습니다.

PM/400을 사용하여 수행할 수 있는 다른 태스크에 대한 내용은 PM/400 관리를 참조하십시오.

PM/400 관리

PM/400을 사용할 수 있도록 네트워크를 설정한 후 다음 태스크를 수행할 수 있습니다.

PM/400 비활성화

PM/400을 중단하는 방법을 배웁니다.

문의처 정보 변경

원래 설정에서 문의처 정보를 변경하는 방법을 배웁니다.

PM/400을 사용한 작업 스케줄링

PM/400을 사용하여 작업을 스케줄하는 방법을 배웁니다.

PM/400 분석에서 항목 생략

PM/400을 사용하여 분석을 수행할 때 작업, 사용자 및 통신 회선을 생략하는 방법을 배웁니다.

PM/400 일시 정지

PM/400을 일시적으로 중단하는 방법을 배웁니다.

PM/400 상태 표시

iSeries Navigator나 PM/400 메뉴를 사용하여 PM/400 상태를 표시하는 방법을 배웁니다.

PM/400 보고서 보기

PM/400 보고서의 예와 이 보고서를 해석하는 방법에 대한 설명을 참조하십시오.

자료의 그래프 이력 보기

그래프 이력은 지정된 기간 동안 수집된 성능 자료를 그래픽 화면으로 제공합니다. 이 자료를 보는 방법을 찾아 보십시오.

PM/400 비활성화: PM/400의 실행을 중단하려면 다음 방법 중 하나를 사용하십시오.

iSeries Navigator 사용

다음 단계를 수행하십시오.

1. iSeries Navigator에서 PM/400이 실행되는 시스템을 확장합니다.
2. 구성 및 서비스를 확장하십시오.
3. 콜렉션 서비스를 마우스 오른쪽 버튼으로 클릭합니다.
4. **Performance Management/400**을 선택합니다.
5. 중단을 선택합니다.
6. PM/400을 중단할 시스템을 선택합니다.
7. 확인을 클릭하십시오.

API 사용

PM/400 종료(Q1PENDPM) API를 사용하여 PM/400을 비활성화합니다.

수행할 수 있는 다른 태스크에 대한 내용은 PM/400 관리를 참조하십시오.

PM/400 문의처 정보 변경: PM/400 소프트웨어를 구성하는 중에 연락 책임자 및 메일 정보를 식별하게 됩니다. 나중에 이 정보를 갱신해야 하는 경우 문의처 정보에 대한 작업 옵션을 사용하여 변경하십시오. 문의처 정보를 변경하려면 다음 단계를 수행합니다.

1. 명령행에서 **GO PM400**을 입력합니다.
2. PM/400 메뉴에서 1을 입력하고 Enter를 누릅니다. 문의처 정보에 대한 작업 화면이 나타납니다.
3. 문의처 정보를 알맞게 변경한 후 Enter를 누릅니다.

수행할 수 있는 다른 task에 대한 내용은 PM/400 관리를 참조하십시오.

PM/400을 사용한 작업 스케줄링: PM/400 소프트웨어에는 PM/400 성능 자료 콜렉션 및 분석을 지원하는 데 필요한 작업을 자동으로 시작하는 스케줄러가 있습니다.

PM/400 소프트웨어 활성화 프로세스의 일부에는 Q1PSCH라고 하는 작업을 시작하는 것이 포함됩니다. 이 작업은 다시 다음 표에 보인 것과 같은 다른 작업을 시작합니다.

PM/400의 스케줄된 작업에 액세스하려면 다음을 수행하십시오.

1. 명령행에서 **GO PM400**을 입력합니다.
2. PM/400 메뉴에서 2를 입력하고 Enter를 누릅니다. 자동으로 스케줄된 작업에 대한 작업 화면이 표시됩니다.
3. 각 작업에 대한 상태를 활동에서 비활동에서 변경할 수 있습니다. 변경하려는 작업 옆에 2(변경)를 입력하고 Enter를 누릅니다. 자동으로 스케줄된 작업 변경 화면이 표시됩니다.

다음 표에 가능한 PM/400 작업이 나와 있습니다.

PM/400 스케줄된 작업

작업	스케줄	기능
Q1PTEST	활성화 중에	PM/400이 활성화되어 있는지 확인한 후 비 활동 상태로 들어갑니다.
Q1PCM1	매주	요약된 성능 자료를 IBM으로 전송합니다. 이 작업은 직접 다이얼 회선을 사용하는 경우에만 활성화됩니다.
Q1PCM2	매일	통신을 오프라인으로 변환합니다.
Q1PPMSUB	매 시간	콜렉션 서비스가 자료를 수집하는지 확인합니다.
Q1PDR	매일	자료 축약을 수행하고 성능 자료를 제거합니다.
Q1PPG	매월	축약된 성능 자료를 제거합니다.
Q1PCM3	필요시	직접 다이얼 전송이 회선을 단절변환한 후 통신을 오프라인으로 변환합니다.
Q1PCM4	필요시	리모트 서버에서 PM/400 자료에 액세스합니다. 이 작업은 PM/400 메뉴에서 옵션 5를 사용하여 리모트 시스템을 추가한 경우에만 시작됩니다.

Q1PPMCHK	4시간마다	자료 수집이 활동 중인지 확인합니다.
Q1PMONTH	매월	해당월 중에 추가 전송이 필요한 경우 월간 전송을 허용합니다. 디폴트 값은 비활동으로 설정됩니다. 이 작업은 직접 다이얼 회선을 사용하는 경우에만 사용할 수 있습니다.

수행할 수 있는 다른 태스크에 대한 내용은 PM/400 관리를 참조하십시오.

PM/400 분석에서 항목 생략: PM/400 소프트웨어 요약에는 일괄처리 작업, 사용자 및 통신 회선에 대한 상위 10개의 항목 분석이 포함됩니다. 그러나 일부 작업, 사용자 또는 통신 회선은 그러한 분석에 적합하지 않습니다. 예를 들어 런타임 범주에서 자동시작 작업과 같은 정상적인 실행 시간보다 긴 작업은 제외시키고자 할 수 있습니다.

총괄 생략 기능을 사용하여 상위 10개의 분석으로부터 일괄처리 작업 및 사용자 그룹을 생략할 수 있습니다. 예를 들어 MYAPP로 시작하는 작업을 모두 생략하려면 MYAPP*를 지정합니다.

생략 작업을 하려면 다음 단계를 수행하십시오.

1. 명령행에서 **GO PM400**을 입력합니다.
2. PM/400 메뉴에서 4를 입력하고 Enter를 누릅니다. 상위 10개 항목 생략에 대한 작업 화면이 표시됩니다.
3. 어떤 항목을 생략하고자 하는가에 따라 적절한 옵션 번호를 입력합니다.
 - 작업에 대한 작업은 1을 입력합니다.
 - 사용자에 대한 작업은 2를 입력합니다.
 - 통신 회선에 대한 작업은 3을 입력합니다.
4. 특정 범주에서 사용자나 작업을 생략하려면 적절한 필드에 1을 입력합니다. 통신 회선의 경우, 회선명을 입력하고 적절한 필드에 1을 입력합니다.

수행할 수 있는 다른 태스크에 대한 내용은 PM/400 관리를 참조하십시오.

PM/400 일시 정지: 콜렉션 서비스의 자료 수집을 PM/400이 확인하는 것을 중단해야 하는 경우 스케줄러 작업을 사용하여 Q1PPMSUB 작업에 대한 날짜를 미래의 날짜로 변경합니다.

1. 명령행에서 **GO PM400**을 입력합니다.
2. 2(자동 스케줄된 작업에 대한 작업)를 입력합니다.
3. Q1PPMSUB 옆에 2(변경)를 입력합니다.
4. 날짜나 시간을 미래의 날짜와 시간으로 변경합니다.
5. Enter를 누릅니다. 그러면 일시적으로 PM/400의 콜렉션 서비스 자료 수집 확인을 중단하게 됩니다. 현재 수집되고 있는 것을 종료해야 합니다.

주: Q1PPMSUB 작업에 설정한 날짜와 시간에 도달할 때까지 PM/400은 콜렉션 서비스를 시작, 순환 또는 변경하지 않습니다.

스케줄러를 사용하여 수행할 수 있는 다른 작업에 대한 내용은 PM/400을 사용한 작업 스케줄링을 참조하십시오.

수행할 수 있는 다른 task에 대한 내용은 PM/400 관리를 참조하십시오.

PM/400 상태 표시: 서버에서 iSeries Navigator나 PM/400 메뉴를 사용하여 PM/400의 상태를 표시할 수 있습니다. Performance Management/400 상태 대화 상자를 사용하여 하나 이상의 서버나 그룹에서 PM/400의 전반적인 상태를 볼 수 있습니다. 예를 들어 PM/400이 사용 중인지 여부에 대한 세부사항을 볼 수 있습니다. PM/400 메뉴를 사용하여 콜렉션 서비스 상태, PM/400 스케줄러 상태, 성능 자료 릴리스, 최종 전송 시도, 성능 자료 멤버 및 성능 자료 크기를 볼 수 있습니다.

iSeries Navigator에서 PM/400의 전반적인 상태를 보려면 다음 단계를 수행하십시오.

1. iSeries Navigator에서 종료점 시스템이나 시스템 그룹을 확장합니다.
2. 구성 및 서비스를 확장하십시오.
3. 콜렉션 서비스를 마우스 오른쪽 버튼으로 클릭합니다.
4. **Performance Management/400**을 선택합니다.
5. 상태를 선택합니다.

PM/400 메뉴에서 PM/400의 자세한 상태를 보려면 다음 단계를 수행하십시오.

1. 명령행에서 **GO PM400**을 입력합니다.
2. 명령행에서 6을 입력하고 Enter를 누릅니다. 각 필드에 대한 설명은 온라인 도움말을 참조하십시오.

수행할 수 있는 다른 task에 대한 내용은 PM/400 관리를 참조하십시오.

PM/400 보고서 보기: PM/400e 서비스의 출력은 월별 또는 분기별로 작성된 관리 보고서와 그래프입니다. PM/400e에는 보고서에 대한 두 가지 옵션이 있습니다.

보고서와 그래프의 목적은 관리자가 현재 서버의 성능 및 정확한 확장 추세를 명확하게 이해할 수 있도록 하는 것입니다. 각 보고서와 그래프를 자세히 보고 장점 및 사용법에 대한 내용을 배우려면

PM/400 웹사이트  를 참조하십시오.

수행할 수 있는 다른 task에 대한 내용은 PM/400 관리를 참조하십시오.

성능 분석 툴

iSeries용 성능 분석 툴 사용권 프로그램을 이용하면 다양한 방법으로 성능 자료를 분석할 수 있습니다. 성능 분석 툴은 자료를 보고, 보고하고 그래프화하기 위한 툴과 명령의 집합입니다. iSeries용 성능 분석 툴을 사용하여 콜렉션 서비스를 사용하여 수집된 성능 자료를 보거나 STRPFTRC(성능 추적 시작) 명령을 사용하여 수집된 추적 자료를 볼 수 있습니다. 그런 후 시스템의 성능 문제를 파악하기 위해 자료를 보고서로 요약할 수 있습니다. 또한 일정 기간 동안의 자원 활용을 보기 위해 성능 자료의 그래프를 작성할 수도 있습니다.

iSeries용 성능 분석 툴에는 기본 제품과 두 개의 피처(관리자 및 에이전트)가 들어 있습니다. 기본 제품과 둘 중 한 피처는 필수입니다. 성능 분석 툴의 관리자 및 에이전트 피처에 대한 자세한 정보는 관리자 및 에이전트 피처 비교 주제를 참조하십시오.

성능 분석 툴 개념


성능 정보를 수집 및 분석하는 데 도움이 되는 다양한 툴에 대해 설명합니다. 어떤 툴이 어떤 기능을 수행하는지와 이들의 작동 방식에 대한 상세 정보를 찾아 보십시오.

성능 분석 툴 설치 및 구성

설치 및 설정 지침에 대해서는 이 주제를 참조하십시오.

성능 분석 툴 보고서

성능 분석 툴 보고서는 일정 기간 동안 수집된 자료에 대한 정보를 제공합니다. 성능과 시스템 자원의 사용에 대한 추가 정보를 얻으려면 이들 보고서를 이용하십시오.

성능 분석 툴을 사용하여 시스템, 작업 또는 프로그램의 성능에 대한 자료를 수집하는 방법에 대한 자세한 내용은 성능 분석 툴 책  을 참조하십시오. 이 책에서는 문제를 식별하고 정정하기 위해 자료를 분석하고 인쇄하는 방법에 대해서도 설명합니다. 성능 분석 툴이 이 TASK에서 소비하는 CPU 시간을 표시하는 방법을 알려면 HVLPTASK TASK를 참조하십시오.

성능 분석 툴 개념

iSeries용 성능 분석 툴 사용권 프로그램은 두 개의 고유한 성능 자료 유형 즉, 샘플 자료와 추적 자료를 분석합니다. 콜렉션 서비스는 정규 간격으로 캡처되는 요약 자료인 샘플 자료를 수집합니다. 사용자는 경향 분석 및 성능 분석을 위해 샘플 자료를 수집합니다. 자료는 기억장치 풀 및 응답 시간과 같은 것과 관련됩니다. 그러나 콜렉션 서비스는 추적 자료의 콜렉션을 지원하지 않습니다. 추적 자료는 특정 작업과 트랜잭션에 대한 추가 정보를 얻기 위해 수집하는 상세 자료입니다. 추적 자료를 수집하려면 STRPFRTRC(성능 추적 시작) 명령이나 Performance Explorer를 이용하십시오.

성능 분석 툴에 포함된 기능

성능 분석 툴에는 성능 자료를 수집, 분석 및 보고하기 위한 다양한 어플리케이션이 들어 있습니다. 사용할 수 있는 기능에 대해 알고 주어진 TASK에 가능 적합한 것이 어떤 것인지를 아는 것은 복잡할 수 있습니다. 이 사용권 프로그램에 들어 있는 기능에 대한 설명은 이 주제를 참조하십시오.




관리자 및 에이전트 피처 비교

관리자 및 에이전트 피처를 사용하여 분산 환경에서 성능 분석 툴의 필수 기능을 효율적으로 나눌 수 있습니다. 이 주제에는 이들 두 피처에 대한 설명과 이들 각각에 들어 있는 기능, 이들을 가장 효과적으로 사용하는 방법에 대한 정보가 들어 있습니다.

성능 정보 표시

iSeries Navigator에서 시스템 자원 활용 자료를 볼 수 있습니다. 자료를 보고 이를 그래프화하며 보고서로 요약할 수 있습니다. 여기에서 이 정보에 액세스하는 방법에 대한 정보를 찾아 보십시오.

성능 분석 툴에 제공되는 기능: 성능 분석 툴에는 보고서, 대화식 명령 및 기타 기능이 포함되어 있습니다. 예를 들어 성능 분석 툴에는 다음이 포함됩니다.

툴	설명
WRKSYSACT(시스템 활동에 대한 작업) 명령	WRKSYSACT(시스템 활동에 대한 작업) 명령을 사용하여 시스템에서 현재 실행 중인 작업, 스레드 및 태스크에 대해 대화식으로 작업할 수 있습니다. WRKSYSACT 명령은 시스템 자원 활용(공유 처리 풀을 사용하는 파티션에 대한 태스크 기반의 CPU 활용 등)을 보고합니다. 이 명령이 이 태스크에서 소비하는 CPU 시간을 표시하는 방법을 알려면 HVLPTASK 태스크를 참조하십시오.
성능 자료 표시	성능 자료 표시 그래픽 인터페이스를 통해 성능 자료를 보고, 자료를 요약하여 보고서를 만들며 그래프를 작성하여 경향을 파악하고 iSeries Navigator 내에서 자세한 시스템 성능을 분석할 수 있습니다.
보고서	보고서는 콜렉션 서비스 성능 자료를 논리적이고 유용한 형식으로 조직화합니다. 이 툴에 대해서는 성능 분석 툴 책  에서 자세히 설명됩니다.
그래픽 기능	성능 분석 툴 그래픽 기능을 사용하여 성능 자료에 대하여 그래픽 형식으로 작업할 수 있습니다. 대화식으로 그래프를 표시하고, 프린터나 플로터로 인쇄하고 또는 다른 유틸리티에서 사용할 목적으로 그래픽 자료 형식(GDF) 파일에 저장할 수 있습니다. 이 툴에 대해서는 성능 분석 툴 책  에서 자세히 설명됩니다.
Performance Explorer	Performance Explorer는 콜렉션 서비스가 수집한 샘플 자료나 일반적인 경향 분석으로는 식별할 수 없는 성능 문제점의 원인을 파악하는 데 도움이 되는 자료 콜렉션 툴입니다. Performance Explorer는 프로그램, 프로시저, 모듈 또는 메소드 레벨에서 자세한 어플리케이션 분석 용도로 사용됩니다. 예를 들어 개별 프로그램이나 프로시저, CPU 및 I/O 통계 또는 개별 오브젝트 I/O 특성에 대한 추적 자료를 수집할 수 있습니다. 이 툴에 대해서는 성능 분석 툴 책  에서 자세히 설명됩니다.

관리자 및 에이전트 피처 비교: 성능 분석 툴은 두 개의 별도 설치가능 피처와 함께 사용할 수 있습니다. 이 주제에서는 사용자 어플리케이션에 더 적합한 피처를 판별할 수 있도록 두 피처 사이의 차이점에 대해 설명합니다.

관리자 피처

성능 분석 툴 관리자 피처는 완전한 기능을 갖춘 패키지로, 분산 환경의 중앙 시스템이나 단일 시스템에서 사용하도록 고안되었습니다. 추적 자료 분석, 자료의 그래픽 보기, 시스템 활동을 실시간으로 보기 또는 시스템 성장을 관리 및 추적해야 하는 경우, 성능 분석 툴 사용권 프로그램의 관리자 피처가 더 유용합니다.

에이전트 피처

관리자 기능의 서브세트를 갖춘 성능 분석 툴 에이전트 피처는 기본 기능을 갖는 저가형 패키지입니다. 분산 환경에서는, 상세 분석이 필요한 경우 자료를 관리자에 송신할 수 있으므로 에이전트 피처는 네트워크의 관리 시스템에서 잘 작동합니다. 또한 적절한 레벨의 자체 숙련을 필요로 하지만 전문 기술이 없는 사이트에 대한 효과적인 툴입니다.

성능 분석 툴의 에이전트 피처는 성능 자료의 수집, 관리, 온라인 표시, 자료 축소 및 분석을 간소화하기 위한 기능을 제공합니다. Performance Explorer 보고서 기능 및 이와 연관된 명령은 iSeries용 성능 분석 툴 사용권 프로그램의 기본 옵션에 들어 있으므로 관리자 피처 또는 에이전트 피처에서 사용할 수 있습니다. 에이전트

피처에 들어 있지 않은 주요 성능 분석 툴 기능은 성능 및 추적 보고서 성능 유틸리티(작업 추적 및 파일과 액세스 그룹 선택), 시스템 활동 모니터링 및 성능 그래프입니다.

성능 정보 표시: 성능 분석 툴은 iSeries Navigator의 성능 자료를 표시할 수 있습니다. 이 그래픽 인터페이스를 통해, 성능 자료를 보고 자료를 요약하여 보고서를 만들며 그래프를 작성하여 경향을 파악하고 자세한 시스템 성능을 분석할 수 있습니다.

미터법

iSeries Navigator는 선택된 시간 간격 동안 성능 미터법을 표시합니다. 성능 자료 표시 창의 그래프 분할 창에서 볼 수 있는 성능 미터법에는 다음이 포함됩니다.

- 트랜잭션 계수
- 트랜잭션 응답 시간
- 전체 CPU 활용
- 대화식 CPU 활용
- 일괄처리 CPU 활용
- 높은 디스크 활용
- 기계 풀(pool) 페이지 결함/초
- 사용자 풀(pool) 페이지 결함/초
- 예외

세부사항 분할 창에서는 선택된 시간 간격에 대한 상세한 성능 자료를 다양한 방법으로 볼 수 있습니다. 시스템 성능을 분석하기 위해 작업 자료, 서브시스템 자료, 풀 자료 또는 디스크 장치 자료를 볼 수 있습니다.

보고서

그래프와 상세 자료를 보는 외에, 성능 자료 표시 창에서 보고서를 인쇄할 수도 있습니다. 성능 보고서는 성능 문제를 유발하는 시스템 영역을 연구할 수 있는 한 방법이 됩니다. 여러 보고서를 실행하여 시스템 자원이 사용되고 있는 위치를 확인할 수 있습니다. 성능 분석 툴에서 보고서 인쇄는 iSeries용 성능 분석 툴(5722-PT1)의 옵션 1(관리자 피처)이 중앙 시스템에 설치된 경우에만 사용할 수 있습니다. 관리자 피처에 대한 자세한 정보는 관리자 및 에이전트 피처 비교 주제를 참조하십시오.

성능 자료 표시 창에서 인쇄할 수 있는 보고서는 다음과 같습니다.

- 시스템
- 구성요소
- 작업
- 풀
- 자원

iSeries Navigator를 통해 액세스

성능 자료 표시 창에는 다음 단계를 수행하여 iSeries Navigator에서 액세스할 수 있습니다.

1. iSeries Navigator에서 사용자 연결(또는 활성 환경)을 확장하십시오.
2. 보려는 성능 자료가 들어 있는 서버를 확장하십시오.
3. 구성 및 서비스를 확장하십시오.
4. 컬렉션 서비스를 마우스 오른쪽 버튼으로 클릭하고 성능 분석 툴을 선택한 후 성능 자료를 선택하십시오.
5. 표시하려는 성능 자료 파일을 선택하십시오.
6. 표시를 클릭하십시오.

iSeries Navigator에서 성능 자료 표시 창을 사용하는 방법에 대한 자세한 정보는 iSeries Navigator 온라인 도움말을 참조하십시오.

성능 분석 툴 설치 및 구성

성능 분석 툴을 설치하려면 시스템 저장(*SAVSYS) 권한이 있는 사용자 프로파일이 필요합니다. 시스템 오피레이터 프로파일을 사용하여 이 권한을 획득할 수 있습니다.

성능 분석 툴은 라이브러리 QPFR에서 실행해야 합니다. 시스템에 이 이름의 라이브러리가 있는 경우 성능 분석 툴을 설치하기 전에 RNMOBJ(오브젝트 이름 변경) 명령을 사용하여 이의 이름을 변경하십시오. 이 단계를 수행함으로써 성능 분석 툴의 적절한 운영이 보장됩니다.

다음 명령을 사용하여 성능 분석 툴을 라이브러리 QPFR에 배치하십시오.

```
RSTLICPGM LICPGM(5722PT1) DEV(NAME) OPTION(*BASE)
```

그런 후 다음 중 하나를 수행해야 합니다.

- 관리자 피처를 구입한 경우 다음 명령을 사용하십시오.

```
RSTLICPGM LICPGM(5722PT1) DEV(tape-device-name) OPTION(1)
```

- 에이전트 피처를 구입한 경우 다음 명령을 사용하십시오.


```
RSTLICPGM LICPGM(5722PT1) DEV(NAME) OPTION(2)
```

설치할 CD-ROM이 여러 개인 경우 다음 상황이 발생할 수 있습니다. 첫 번째 CD-ROM을 설치한 후 사용권 프로그램이 복원되었지만 언어 오브젝트가 복원되지 않았음을 나타내는 메시지가 수신됩니다. 이 경우 다음 번 CD-ROM을 넣고 다음을 입력하십시오.

```
RSTLICPGM LICPGM(5722PT1) DEV(NAME) RSTOBJ(*LNG) OPTION(*BASE)
```


성능 분석 툴 프로그램을 설치하는 또다른 방법은 GO LICPGM을 입력하고 메뉴 옵션을 사용하는 것입니다.

성능 분석 툴은 프로세서 기반 프로그램입니다. 사용 유형은 동시적이며 프로그램은 사용 한계를 *NOMAX로 하여 설치됩니다.

이 프로그램은 성능 분석 툴 책  에서 자세히 설명됩니다.

성능 분석 툴 보고서

성능 분석 툴은 수집된 자료를 검토할 수 있는 쉬운 방법을 제공합니다. 콜렉션 서비스는 트랜잭션, 잠금 및 추적 보고서를 제외한 대부분의 성능 분석 툴 보고서 자료를 제공합니다. 이들 세 가지 보고서에 대한 추적 정보를 수집하려면 STRPFTRC 및 ENDPFTRC 명령을 사용해야 합니다. 성능 분석 툴 보고서를 참조하여 성능 문제를 가려낼 수 있습니다. 일정 기간 동안 성능 자료를 수집한 후 시스템 자원이 사용된 방식 및 사용된 곳을 보여주는 보고서를 인쇄할 수 있습니다. 보고서는 전반적인 응답 시간을 지연시키는 특정 어플리케이션 프로그램, 사용자 또는 비효율적인 작업부하 등을 지적합니다.

다음은 각 보고서에 대한 내용 및 특정 보고서를 사용해야 하는 이유를 간략히 설명하는 리스트입니다. 각 보고서는 성능 분석 툴 책  에서 자세히 설명됩니다.

주: STRPFMON(성능 모니터 시작) 명령을 사용하여 이전 릴리스에서 수집되었던 자료에 대한 보고서를 인쇄할 수 있습니다.

성능 분석 툴 보고서 개요

보고서	설명	내용	정보 사용 방법
시스템	콜렉션 서비스 자료를 사용하여 시스템의 작동 방식에 대한 개요를 제공합니다. 보고서에는 작업부하, 자원 사용, 기억장치 풀 활용, 디스크 활용 및 통신에 대한 요약 정보가 들어 있습니다. 이 보고서를 자주 실행 및 인쇄하여 시스템 사용에 대해 파악하도록 하십시오.	시스템 작업부하. 보고서에는 데이터베이스 기능 자료가 포함됩니다.	예상 작업부하
구성요소	콜렉션 서비스 자료를 사용하여 동일한 시스템 성능 구성요소에 대한 정보를 상세한 시스템 보고서로 제공합니다. 이 보고서를 통해 시스템 자원(CPU, 디스크 등)을 다량으로 소비하는 작업을 찾을 수 있습니다.	자원 사용, 통신, 시스템 및 사용자 작업. 보고서에는 데이터베이스 기능 자료와 대화식 피처 활용이 포함됩니다.	하드웨어 확장 및 구성 처리 경향
트랜잭션	추적 자료를 사용하여 성능 자료 수집 중 발생한 트랜잭션에 대한 자세한 정보를 제공합니다.	작업부하 및 CPU 활용, 디스크, 주 기억장치, 트랜잭션 작업부하, 오브젝트 경합	예상 작업부하, 풀 구성, 어플리케이션 설계, 파일 경합 및 프로그램 사용

잠금	추적 자료를 사용하여 시스템 작동 중의 잠금 및 점유 충돌에 대한 정보를 제공합니다. 이 정보를 토대로, 충족되지 않은 잠금 요구 또는 내부 기계 점유 충돌로 인해 처리 중 작업이 지연되는지를 판별할 수 있습니다. 이러한 조건을 대기라고도 합니다. 대기 조건이 발생하는 경우 작업이 대기 중인 오브젝트와 대기 시간을 판별할 수 있습니다.	시간별 파일 레코드 또는 오브젝트 경합, 보류 중인 작업 또는 오브젝트명, 작업 요구 또는 오브젝트명	문제점 분석. 오브젝트 경합의 감소 또는 제거.
추적	추적 자료를 사용하여 추적된 여러 가지 작업 유형(예: 일괄처리 작업)의 진행을 보여줍니다. 활용된 자원, 예외 및 상태 전이가 보고됩니다.	작업 클래스 시간 분할 종료 및 추적 자료	문제점 분석 및 일괄처리 작업 진행
작업	컬렉션 서비스 자료를 사용하여 전부 또는 선택된 간격 및 작업에 대한 정보(대화식 작업과 비대화식 작업에 대한 상세 및 요약 정보 등)를 보여줍니다. 보고서가 길어질 수 있으므로 포함시키려는 간격과 작업을 선택하여 출력을 제한할 수도 있습니다.	간격을 둔 작업	작업 자료
풀	컬렉션 서비스 자료를 사용하여 서브시스템 활동에 대한 섹션과 풀(pool) 활동에 대한 섹션을 제공합니다. 각 샘플 간격에 대해 자료가 표시됩니다. 보고서가 길어질 수 있으므로 포함시키려는 간격과 작업을 선택하여 출력을 제한할 수도 있습니다.	간격을 둔 풀	풀 자료
자원 간격	컬렉션 서비스 자료를 사용하여 전부 또는 선택된 간격에 대한 자원 정보를 제공합니다. 보고서가 길어질 수 있으므로 포함시키려는 간격을 선택하여 출력을 제한할 수도 있습니다.	간격을 둔 자원	시스템 자원 사용

Performance Explorer 및 컬렉션 서비스는 별개의 컬렉션 에이전트입니다. 각각은 수집된 자료 세트가 포함된 개별적인 데이터베이스 파일을 생성합니다. 동시에 두 컬렉션을 실행시킬 수 있습니다.

다른 톨에 대한 보고서 리스트는 다음을 참조하십시오.


- Performance Explorer 보고서

- Performance Management/400 보고서

Performance Explorer

Performance Explorer는 콜렉션 서비스를 사용한 자료 수집이나 일반적인 경향 분석으로는 식별할 수 없는 성능 문제점의 원인을 파악하는 데 도움이 되는 자료 콜렉션 툴입니다. 이 Performance Explorer를 사용하는 두 가지 이유로 다음을 들 수 있습니다.

- 문제를 초래하는 시스템 자원, 어플리케이션, 프로그램, 프로시저어 또는 메소드에 따라 성능 문제를 분리
- 어플리케이션의 성능 분석

Performance Explorer의 콜렉션 기능 및 관련 명령은 OS/400 사용권 프로그램의 일부입니다. 보고 기능과 이와 연관된 명령은 iSeries용 성능 분석 툴 사용권 프로그램의 기본 옵션에 들어 있으므로 관리자 피쳐 또는 에이전트 피쳐에서 사용할 수 있습니다. AS/400 Performance Explorer Tips and Techniques 책  에서는 Performance Explorer 기능의 추가 예와 향상된 Performance Explorer 추적 지원의 예를 제공합니다.

Performance Explorer는 일반적인 성능 모니터링을 수행하는 툴로는 식별할 수 없는 성능 문제점의 원인을 파악하는 데 도움이 되는 툴입니다. 컴퓨터 환경은 크기와 복잡성이 점차 커지므로 성능 분석에서도 복잡성이 증대되는 것은 당연합니다. Performance Explorer는 복잡한 성능 문제에 대한 자료를 수집하여 복잡성 증대에 따른 문제를 해결합니다.

주: Performance Explorer는 다른 툴을 사용해본 후 사용해야 하는 툴입니다. 이는 성능 문제에 관련된 요소를 좀 더 쉽게 분리시킬 수 있는 특정한 형태의 자료를 수집합니다. 그러나 이 자료를 수집할 때 시스템의 성능에 커다란 영향을 줄 수 있습니다.

이 툴은 프로그램의 성능에 대한 이해와 개선에 관심이 있는 어플리케이션 개발자를 위해 고안되었습니다. 또한 복잡한 성능 문제를 식별하여 분리해낼 수 있는 성능 관리에 대한 지식이 있는 사용자에게도 도움이 됩니다.

Performance Explorer에 대해 더 자세히 알려면, 다음의 Performance Explorer 주제를 참조하십시오.

Performance Explorer 개념


Performance Explorer는 지정된 시스템 프로세스나 자원에 대한 상세 정보를 수집하여 작동합니다. 이 주제에서는 Performance Explorer의 작동 방식과 Performance Explorer를 가장 잘 사용할 수 있는 방법에 대해 설명합니다.

Performance Explorer 구성


자세한 추적 정보를 수집하려면 추적이 이루어지는 어플리케이션 프로세스에서 가장 잘 작동하도록 Performance Explorer를 조정해야 합니다.

Performance Explorer 보고서

Performance Explorer 세션에서 성능 자료를 수집한 후 포함된 보고서를 실행하거나 데이터베이스 파일을 직접 조회하여 이 자료를 볼 수 있습니다.

자세한 내용은 성능 분석 툴 책  을 참조하십시오.

Performance Explorer 개념

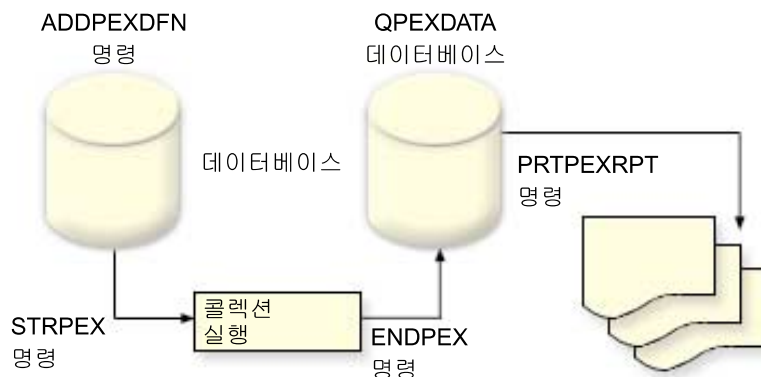
컬렉션 서비스와 마찬가지로 Performance Explorer는 차후 분석을 위해 자료를 수집합니다. 그러나 이들은 서로 다른 유형의 자료를 수집합니다. 컬렉션 서비스는 정기적인 스케줄 간격으로 광범위한 시스템 자료를 수집하며 시스템 자원은 최소한으로 소비합니다. 이와 반대로 Performance Explorer는 추적 레벨 자료를 수집하는 세션을 시작합니다. 이 추적은 어플리케이션, 작업 또는 스레드에서 소비하는 자원에 대해 다량의 상세 정보를 생성합니다. 특히 Performance Explorer에서는 시스템 생성 디스크 I/O, 프로시저 호출, Java 메소드 호출, 페이지 결함 및 기타 trace events  와 같은 영역에 대한 특정 질문에 응답할 수 있습니다. 이것은 성능 문제를 가려내는 데 있어서 Performance Explorer를 효과적으로 만들어주는 아주 특정하고 상세한 정보를 수집하는 능력입니다. 예를 들어 컬렉션 서비스는 디스크 기억장치 공간이 급속하게 소비되고 있음을 알려줄 수 있습니다. Performance Explorer를 사용하여 많은 디스크 공간을 소비하고 있는 프로그램 및 오브젝트를 식별하고 그 이유도 알아낼 수 있습니다.

주: Performance Explorer 자료와 컬렉션 서비스 자료를 동시에 수집할 수 있습니다.

Performance Explorer의 작동 방식

다음 그림은 Performance Explorer의 정상 경로에 익숙해지는 데 도움이 될 것입니다. 이들 각 단계에 대한 세부사항은 Performance Explorer 구성을 참조하십시오. 이 그림은 다음 단계로 이루어진 기본 작업 주기를 보여줍니다.

1. Performance Explorer 자료 컬렉션을 정의합니다. 또한 특정 이벤트에 대한 비교 값을 지정하여 수집된 자료의 양을 제한하기 위한 필터를 추가할 수도 있습니다.
2. Performance Explorer를 시작하여 사용자의 정의에 기반한 자료를 수집합니다.
3. 프로그램, 명령 또는 작업부하를 실행합니다.
4. 컬렉션을 종료합니다. 그러면 수집된 자료가 데이터베이스 파일 세트에 저장됩니다.
5. 데이터베이스 파일을 가지고 보고서를 작성하여 인쇄합니다.



Performance Explorer에 대해 더 자세히 알려면, 다음의 Performance Explorer 주제를 참조하십시오.

Performance Explorer 정의

Performance Explorer가 수집하는 자료와 이의 자료 수집 방법을 결정하는 매개변수와 조건은 Performance Explorer 정의를 사용하여 구성 및 저장됩니다. 이 주제에서는 이들 정의를 사용하는 방법에 대해 설명하고 간단한 정의를 보여주는 샘플을 제공합니다.

Performance Explorer 데이터베이스 파일

Performance Explorer가 수집하는 자료는 Performance Explorer 데이터베이스 파일에 저장됩니다.

Performance Explorer의 장점

Performance Explorer에는 자세한 성능 정보를 수집 및 분석할 수 있는 여러 가지 기능이 들어 있습니다. 이 주제에서는 이러한 여러 가지 기능에 대한 개요를 제공합니다.

Performance Explorer 정의: Performance Explorer 자료를 수집하려면 수집할 자료를 Performance Explorer에 알려줘야 합니다. 이 작업은 Performance Explorer 정의를 작성하기 위한 ADDPEXDFN(Performance Explorer 정의 추가) 명령을 사용하여 수행하십시오. 정의가 완료되어 저장되면 작업 주기의 다음 task로 진행할 준비가 된 것입니다.

새로운 정의를 작성하기 전에 필요로 하는 정보의 유형과 세부사항의 양을 고려하십시오. Performance Explorer는 다음 유형의 자료 컬렉션을 제공합니다.

통계 유형 정의

CPU를 과도하게 소비하거나 디스크 I/O 조사를 많이 수행하는 어플리케이션 및 IBM 프로그램 또는 모듈을 식별합니다. 보통 잠재적 성능 병목으로 추가 조사가 필요한 프로그램을 식별하려면 통계 유형을 사용하십시오.

- OS/400 프로그램, 프로시저 및 MI 복합 명령어의 최초 분석에 좋습니다.
 - 호출 수를 제공합니다.
 - 인라인 및 누적 CPU 사용을 밀리초로 제공합니다.
 - 동시 및 비동기 I/O의 인라인 및 누적 수를 제공합니다.
 - 작성된 호출 수를 제공합니다.
- 짧거나 긴 실행에서 잘 작동합니다.
- 수집된 자료의 크기가 상당히 작으며 모든 실행에 대해 일관됩니다.
- 호출 빈도 때문에 ILE 프로시저의 런타임 컬렉션 오버헤드가 문제가 될 수 있습니다. 런타임이 저하되어도 수집되는 통계는 정확한데 Performance Explorer가 대부분의 컬렉션 오버헤드를 자료에서 제거하기 때문입니다.
- 결합 또는 분리된 자료 영역을 사용합니다. ADDPEXDFN 명령의 MRGJOB 매개변수는 모든 프로그램 통계가 하나의 자료 영역에 누적되는지 또는 별도로 보관(예: 각 작업에 대해 하나의 작업 영역)되는지를 지정합니다.

통계는 계층식 또는 편형 구조를 가질 수 있습니다.

- 계층 구조는 통계를 호출 트리 형태로 구성하는데, 여기서는 트리의 각 노드가 작업이나 타스크에 의해 실행되는 프로그램 프로시듀어를 나타냅니다.
- 편형 구조는 통계를 프로그램이나 프로시듀어의 단순 리스트로 구성하는데 이들 프로그램이나 프로시듀어는 자체의 통계 세트를 갖습니다.

다음은 프로그램이나 프로시듀어 레벨에서 CPU 및 디스크 자원 사용을 보여주는 MYSTATS라는 Performance Explorer 통계 정의의 예입니다.

```
ADDPEXDFN DFN(MYSTATS) /* 정의명. */
TYPE(*STATS) /* 정의 유형 */
JOB(*ALL) /*모든 작업 */
TASKS(*ALL) /*모든 타스크 */
MRGJOB(*YES) /* 콜렉션 오버헤드를 줄이기 위해 레코드를 병합 */
DTAORG(*FLAT) /* 호출 대상에 대한 추적을 보관하지 않음 */
```

프로파일 유형 정의

소스 프로그램 명령문 번호에 기초하여 CPU를 과도하게 소비하는 고급 언어(HLL) 프로그램을 식별합니다. 또한 프로그램 종료 시 프로그램과 서브루틴의 시작 사이에서 계속 분기를 발생시키는 프로그램을 식별할 수도 있습니다. 프로그램이 아주 클 경우 이러한 연속적인 앞뒤 이동이 주 기억장치가 제한된 시스템에서는 과도한 페이지 결함을 초래할 수 있습니다.

- 프로그램 프로파일(ADDPEXDFN 명령에서 TYPE(*PROFILE) 및 PRFTYPE(*PGM) 지정)
 - 특정 작업 내에서 사용자가 시간을 소비하는 프로그램 내의 위치에 대한 상세 내역을 제공합니다.
 - 프로그램, 모듈, 프로시듀어, 명령문 또는 명령어별로 자료를 요약할 수 있습니다.
 - 콜렉션의 크기가 비교적 작고 실행 시간에 관계 없이 일정합니다.
 - 16MI 프로그램의 제한은 이를 두 번째 주문 분석 툴로 사용해야 함을 의미합니다.
 - 샘플 간격을 변경하여 오버헤드를 다양하게 할 수 있습니다. 2밀리초의 간격은 벤치마크하기에 좋은 첫 번째 선택사항으로 보입니다.
 - 지정된 프로그램의 수 때문에 분할 창 크기 또는 지정된 프로그램의 크기에 제한이 없습니다.

다음은 특정 프로시듀어에 대한 사용법을 보여주는 PGMPROF라고 하는 Performance Explorer 프로그램 프로파일 정의의 예입니다.

```
ADDPEXDFN DFN(PGMPROF) /* 정의명. */
TYPE(*PROFILE) /* 정의 유형 */
JOB(*ALL) /*모든 작업 */
PGM((MYLIB/MYPGM MYMODULE MYPROCEDURE)) /* 모니터할 프로그램의 이름. */
INTERVAL(1) /* 1밀리초 샘플이 채택됩니다. */
```

- 작업 프로파일(ADDPEXDFN 명령에 TYPE(*PROFILE) 및 PRFTYPE(*JOB) 지정)
 - 작업 세트나 콜렉션 작업에서 사용자가 시간을 소비하는 위치에 대한 상세 내역을 제공합니다.
 - 콜렉션의 크기는 비교적 작지만 일정하지 않습니다. 실행 길이가 증가할수록 크기가 커집니다.
 - 시스템상의 모든 작업과 타스크를 프로파일화하거나 수집되는 자료의 범위를 몇몇 작업이나 관심있는 타스크만으로 좁힐 수 있습니다.

- 샘플 간격을 변경하여 오버헤드를 다양하게 할 수 있습니다. 2밀리초의 간격은 벤치마크하기에 좋은 첫 번째 선택사항으로 보입니다.

다음은 사용자의 작업에 대한 사용법을 보여주는 ALLJOBPROF라고 하는 Performance Explorer 작업 프로파일 정의의 예입니다.

```
ADDPEXDFN DFN(ALLJOBPROF) /* 정의명. */
TYPE(*PROFILE) /* 정의 유형 */
PRFTYPE(*JOB) /* 작업 프로파일 유형이 모니터됨. */
JOB(*ALL) /*모든 작업 */
TASKS(*ALL) /*모든 TASK */
INTERVAL(1) /* 1밀리초 샘플이 채택됩니다. */
```

추적 정의

시스템에 있는 하나 이상의 작업이 생성한 성능 활동의 이력 추적을 수집합니다. 추적 유형은 이벤트 발생 시기와 순서에 대한 특정 정보를 수집합니다. 추적 유형은 프로그램, 사용권 내부 코드(LIC) TASK, OS/400 작업 및 오브젝트 참조 정보에 대한 자세한 참조 정보를 수집합니다.

- 공통 추적 이벤트로는 다음이 있습니다.
 - 프로그램 및 프로시저 호출과 리턴
 - 기억장치(예: 할당 및 할당해제)
 - 디스크 I/O(예: 읽기 조작 및 쓰기 조작)
 - Java 메소드(예: 진입 및 나감)
 - Java(예: 오브젝트 작성 및 가비지 콜렉션)
 - 저널(예: 시작 확약 및 종료 확약)
 - 동기화(예: Mutex 잠금 및 잠금 해제 또는 세마포어 대기).
 - 통신(예: TCP, IP 또는 UDP)
- 실행이 길수록 더 많은 자료를 수집합니다.

다음은 모든 디스크 이벤트의 사용법을 보여주는 DISKTRACE라고 하는 Performance Explorer 추적 정의의 예입니다.

```
ADDPEXDFN DFN(DISKTRACE) /* 정의명. */
TYPE(*TRACE) /* 정의 유형 */
JOB(*ALL) /*모든 작업 */
TASKS(*ALL) /*모든 TASK */
TRCTYPE(*SLTEVT) /* 개별 이벤트와 기계 명령어가
추적 정의에 포함된 경우에만 선택됨 */
SLTEVT(*YES) /* *SLTEVT는 개별 기계 명령어와 이벤트,
TRCTYPE 매개변수에서 사용할 수 있는 이벤트의 범주를
지정할 수 있도록 합니다. */
DSKEVT((*ALL)) /* 모든 디스크 이벤트가 추적됩니다. */
```


Performance Explorer 데이터베이스 파일: 다음 표는 자료 수집 명령을 사용할 때 시스템이 수집한 Performance Explorer 자료 파일을 보여줍니다. 단일 파일의 내용을 보려면 다음과 같이 DSPFFD(파일 필드 설명 표시) 명령을 입력하십시오.

```
DSPFFD FILE(xxxxxxxx)
```

여기서 xxxxxxxx는 표시하려는 파일의 이름입니다.

파일에 포함된 정보의 유형	파일명
참조 정보	QAYPEREF
일반 정보	QAYPERUNI
PMC 선택	QAYPEFQCFG
기본 구성 정보	QAYPECFG
수집된 기계 인터페이스(MI) 복합 명령어	QAYPELCPLX
수집된 작업	QAYPELJOB
자료 수집을 위한 미터법	QAYPELNET
수집된 기계 인터페이스(MI) 프로그램, 모듈 또는 프로시저어	QAYPELMI
자료 수집을 위한 사용권 내부 코드(LIC) 모듈	QAYPELLIC
자료 수집을 위한 타스크명	QAYPELNAMT
자료 수집을 위한 타스크 번호	QAYPELNMT
기계 인터페이스(MI) 복합 명령어 맵핑	QAYPEMICPX
이벤트 유형 및 서브유형 맵핑	QAYPEEVENT
하드웨어 맵핑 자료	QAYPEHWMAP
사용권 내부 코드(LIC) 주소 분석 맵핑	QAYPEPROCI
세그먼트 주소 분석 맵핑	QAYPESEGI
프로세스 및 타스크 분석 맵핑	QAYPETASKI
모든 이벤트에 대한 공통 추적 자료	QAYPETIDX
보조 기억장치 관리 이벤트 자료	QAYPEASM
기본 이벤트 자료	QAYPEBASE
디스크 이벤트 자료	QAYPEDASD
디스크 서버 이벤트 자료	QAYPEDSRV
페이지 결함 이벤트 자료	QAYPEPGFLT
자원 관리 프로세스 이벤트 자료	QAYPERMPM
자원 관리 점유 잠금 이벤트 자료	QAYPERMSL
고급 36 이벤트 자료	QAYPES36
세그먼트 주소 범위(SAR) 자료	QAYPESAR
알 수 없는 이벤트 자료	QAYPEUNKWN
기본 통계 자료	QAYPESTATS
통계 프로파일 요약 자료	QAYPEPSUM
사용권 내부 코드(LIC) 브라켓 자료	QAYPELBRKT
기계 인터페이스(MI) 사용자 인터페이스 자료	QAYPEMIUSR
기계 인터페이스(MI) 프로그램 브라켓 자료	QAYPEMBRKT
기계 인터페이스(MI) 포인터의 주소	QAYPEMIPTR
사용자 정의 브라켓 후크 자료	QAYPEUSRDF
하드웨어 모니터 자료	QAYPEHMON
하드웨어 모니터 전체 자료	QAYPEHTOT
릴리스, 버전, 수정 레벨	QRLVRM
Performance Explorer 레벨 인디케이터	QRLVL

파일에 포함된 정보의 유형	파일명
Performance Explorer Java 이벤트 자료	QAYPEJVA
Performance Explorer Java 클래스 정보 자료	QAYPEJVICI
Performance Explorer Java 메소드 정보 자료	QAYPEJVM I
Performance Explorer Java 이름 정보 자료	QAYPEJVNI
동기화 이벤트 자료	QAYPESYNC
통신 이벤트 자료	QAYPECMN
파일 제공 이벤트 자료	QAYPEFILSV
힙(heap) 이벤트 자료	QAYPEHEAP
PASE 이벤트 자료	QAYEPASE
추적 작업 상응 이벤트 자료	QAYPETBRKT
타스크 교환 이벤트 자료	QAYPETSWSW
동기화 이벤트 자료	QAYPESYNC
프로그램 프로파일 자료	QAYPEPPANE

Performance Explorer 보고서: Performance Explorer는 프로그램이나 작업의 작동 및 성능에 대한 정보를 수집하여 이 정보를 Performance Explorer 데이터베이스 파일에 저장합니다. SQL이나 여러 보고서 중 하나를 실행하여 이들 파일을 조회할 수 있습니다. Performance Explorer를 사용하여 통계, 프로파일, 추적 및 기본 보고서의 네 가지 서로 다른 보고서를 생성할 수 있습니다. 이들 보고서를 생성하기 위해 특정 정의를 사용해야 하는 이유에 대해서는 Performance Explorer 정의를 참조하십시오. 각 보고서는 성능 분석 툴  책에서 자세히 설명됩니다.

PRTPEXRPT(Performance Explorer 보고서 인쇄) 명령을 사용하여 Performance Explorer 보고서를 작성 및 인쇄할 수 있습니다. 추적 보고서를 사용자 정의하려면 OUTFILE 매개변수를 사용하십시오. 다음 명령은 각 유형의 Performance Explorer 자료에 대한 보고서를 인쇄하기 위한 예입니다.

- *STATS 보고서를 사용된 CPU 시간별로 정렬하여 인쇄
PRTPEXRPT MBR(MYSTATS) LIB(MYLIB) TYPE(*STATS) STATSOPT(*CPU)
- 프로파일 보고서를 프로시저별로 요약하여 인쇄
PRTPEXRPT MBR(MYPROFILE) LIB(MYLIB) TYPE(*PROFILE) PROFILEOPT(*SAMPLECOUNT *PROCEDURE)
- 추적을 타스크 ID별로 정렬하여 인쇄
PRTPEXRPT MBR(MYTRACE) LIB(MYLIB) TYPE(*TRACE) TRACEOPT(*TASK)

Performance Explorer는 수집된 자료를 QPFR 라이브러리의 QAVPETRCI 파일에 저장합니다. 단일 레코드의 내용을 보려면 다음 명령을 입력하십시오.

```
DSPFFD FILE(QPFR/QAVPETRCI)
```

Performance Explorer의 장점: Performance Explorer는 iSeries 서버에 대한 자세한 성능 분석을 필요로 하는 사람들에게 도움이 됩니다. Performance Explorer를 사용하여 다음을 수행할 수 있습니다.

- 시스템에서 성능 문제를 초래하는 사안을 사용자, 작업, 파일, 오브젝트, 스레드, 태스크, 프로그램, 모듈, 프로시저, 명령문 또는 명령어 주소 레벨에서 판별합니다.
- 사용자 개발 소프트웨어 및 시스템 소프트웨어에 대한 성능 정보를 수집합니다.
- 시스템 상의 다른 조작 성능에는 영향을 주지 않고 한 작업에 대해 상세 분석을 수행합니다.
- 자료를 수집한 시스템 이외의 시스템에 대한 자료를 분석합니다. 예를 들어 네트워크의 관리 시스템에 대한 자료를 수집하는 경우 분석을 위해 이 자료를 중앙 시스템으로 송신할 수 있습니다.

Performance Explorer 구성

Performance Explorer를 구성하려면 다음 단계를 수행하십시오.

1. 수집하려는 성능 자료를 iSeries 서버에 알려주는 세션 정의를 작성하십시오. Performance Explorer 정의 추가(ADDPEXDFN) 표시 화면에서, 정의에 대한 이름과 콜렉션 유형을 지정하십시오. 이 정의는 라이브러리 QUSRSYS에서 QAPEXDFN 파일에 그 이름으로 데이터베이스 멤버로 저장됩니다. 사용자가 지정하는 이름은 STRPEX(Performance Explorer 시작) 명령에서 사용됩니다.
2. 필터를 추가하십시오(ADDPEXFTR 명령). Performance Explorer 필터는 Performance Explorer 세션 중 수집될 성능 자료를 식별하며 특정 이벤트에 대한 비교 값을 지정하여 수집되는 자료의 양을 제한합니다.
3. 자료 수집을 시작하십시오(STRPEX 명령). *PMCO 이벤트가 수집되지 않는 경우 하나 이상의 Performance Explorer 콜렉션에 작업이 있을 수 있습니다. *PMCO 이벤트가 수집되는 경우에는 모든 콜렉션이 동일한 간격 스펙(ADDPEXDFN INTERVAL() 매개변수)을 갖는 경우에만 작업이 한 콜렉션 이상에 있을 수 있습니다.
4. 분석하려는 자료에 대한 명령, 프로그램 또는 작업부하를 실행하십시오.
5. 자료 수집을 중단하고 분석을 위해 자료를 데이터베이스 파일에 저장하십시오. ENDPEX(Performance Explorer 종료) 명령을 사용하여 콜렉션을 중단하십시오.
6. 성능 자료를 분석하십시오. 성능 분석 툴 사용권 프로그램에 들어 있는 PRTPEXRPT(Performance Explorer 보고서 인쇄) 명령은 각 자료 유형(통계, 프로파일, 추적 프로파일 또는 추적)에 대해 고유한 보고서를 제공합니다. 다른 분석 옵션은 사용자 자신의 조회를 데이터베이스 파일 세트에 기록하는 것입니다.

다음 방법 중 하나를 사용하여 모든 Performance Explorer 명령에 액세스할 수 있습니다.

- 명령 인터페이스. 명령행에 명령을 입력하십시오. PRTPEXRPT(Performance Explorer 보고서) 명령을 제외한 모든 명령은 OS/400 오퍼레이팅 시스템의 부분입니다.
- 성능 분석 툴 메뉴 옵션.

Performance Explorer의 작업 주기를 보려면 Performance Explorer 개념을 참조하십시오.

Performance Explorer 종료

Performance Explorer 세션을 종료하려면 ENDPEX(Performance Explorer 종료) 명령을 사용하십시오. ENDPEX 명령은 수집된 자료에 대해 다음 조치를 수행합니다.

- 수집된 자료를 사용자가 지정한 라이브러리의 QAYPExxx 파일에 배치합니다.
이를 수행하려면 OPTION(*END) 및 DTAOPT(*LIB)를 사용하십시오. 모든 QAYPExxx 파일의 데이터베이스 멤버명은 DTAMBR 매개변수에 대한 이름을 지정하지 않는 경우 세션명을 디폴트로 사용합니다.

RPLDTA(*NO)를 지정하여 이 세션명을 사용하여 수집된 자료를 삭제하거나 RPLDTA(*YES)를 사용하여 수집된 자료를 기존 자료에 추가할 수 있습니다. 기만한 사용자가 아니라면 RPLDTA(*NO)를 사용하십시오.

- 수집된 자료를 하나의 IBM 정의 파일에 배치합니다.

이를 수행하려면 OPTION(*END) 및 DTAOPT(*MGTCOL)를 사용하십시오. 보통 IBM 서비스 담당자의 지시에 따라 *MGTCOL만을 사용하게 됩니다. DTAOPT 매개변수에 *MGTCOL 값을 지정하면 컬렉션 정보가 관리 컬렉션 오브젝트에 저장됩니다. 관리 컬렉션 오브젝트 옵션은 자료가 IBM에 제공되는 경우에만 사용해야 합니다. 성능 분석 틀은 데이터베이스 파일만을 분석할 수 있습니다.

- 수집된 자료를 삭제합니다.

자료를 저장하려면 OPTION(*END)을 사용하고 수집된 자료를 삭제하려면 DTAOPT(*DLT)를 사용하십시오. 수집된 자료가 사용할 수 없는 것으로 판별되면 이를 수행하십시오. 예를 들어 의심스런 작업 중 하나가 예상대로 시작되지 않았습니까. *DLT 옵션을 선택하면 세션에 대해 수집된 성능 자료는 절대 저장되지 않습니다.

- 컬렉션 세션을 일시중단하지만 종료하지는 않습니다.

이를 수행하려면 OPTION(*SUSPEND)을 사용하십시오. 차후에 특정 세션 ID에 대해 STRPEX 명령을 OPTION(*RESUME)과 함께 사용하여 자료 컬렉션을 다시 시작할 수 있습니다.

주: 활동 컬렉션 세션명을 잊은 경우 ENDPEX SSNID(*SELECT) 명령을 사용하십시오.

iDoctor for iSeries

iDoctor for iSeries는 작업 감시자, 오브젝트 탐색기 및 성능 탐색 분석기의 세 가지 구성요소 구성된 툴입니다.

Job Watcher 및 성능 탐색 분석기는 성능 분석용입니다. 이들 구성요소의 서버측 부분은 성능 자료를 좀 더 유용한 형식으로 통합하기 위해 고안된 다양한 분석 프로그램과 자료 컬렉션으로 구성됩니다. Job Watcher와 Analyzer의 클라이언트측 구성요소는 융통성 있는 그래프와 표로 서버 자료를 표시하기 위한 그래픽 인터페이스로 구성됩니다.

Object Explorer는 iSeries 자료가 그래픽 인터페이스에 좀 더 쉽게 액세스할 수 있도록 하기 위해 고안된 툴입니다. 시스템의 임의의 오브젝트 유형을 나열, 보기 및 설명할 수 있습니다. Object Explorer를 이용하면 시스템의 모든 실제 파일(PF)이나 논리 파일에 액세스할 수 있으며 사용이 쉬운 조회 정의와 그래프 정의 인터페이스를 사용하여 자료에 대한 사용자 자신의 조회와 그래프를 작성할 수 있습니다.

Job Watcher

Job Watcher는 작업이 수행 중인 내용 및 실행되지 않는 이유를 매우 상세하게 나타내는 실시간 표와 그래프 자료를 표시합니다. Job Watcher는 간격별로 상세 작업 통계를 제공하는 여러 가지 서로 다른 보고서를 제공합니다. 이들 통계를 이용하여 CPU 활용, DASD 카운터, 대기, 결합, 호출 스택 정보 및 충돌 정보 등과 같은 것을 판별할 수 있습니다.

Object Explorer

Object Explorer를 이용하여 iSeries 서버 상의 오브젝트를 찾아보고 이에 대해 작업할 수 있습니다. Object

Explorer의 1차 피쳐는 자료 표시기(Data Viewer)로 시스템 상의 실제 파일의 내용을 표시하고 자료를 가지고 사용자 자신의 조회와 그래프를 작성할 수 있도록 합니다. 오브젝트 탐색기에는 다음과 같은 피쳐도 들어 있습니다.

- 라이브러리 및 오브젝트 리스트의 쉽고 빠른 필터링
- 빠르고 사용이 쉬운 사용자 인터페이스를 통해 오브젝트를 복사(CRTDUPOBJ), 자르기(MOVOBJ), 이름 변경(RENOBJ) 및 삭제(DLT*)하는 능력
- 논리 파일이나 실제 파일 멤버 찾아보기
- 사용자 자신의 조회를 정의하여 나중에 조회 정의 인터페이스에서 사용하기 위해 저장
- 사용자 자신의 그래프를 정의하여 나중에 그래프 정의 인터페이스에서 사용하기 위해 저장


Performance explorer Analyzer

Performance explorer Analyzer는 시스템의 전반적인 성능을 평가하고 성능 분석 툴 사용권 프로그램을 사용하여 수행한 내용을 빌드합니다. Analyzer는 성능 문제를 분리해내기 위해 그래프로 표시하거나 볼 수 있는 보고서로 추적 자료의 볼륨을 압축합니다. Analyzer는 CPU 활용, 실제 디스크 조작, 논리 디스크 입/출력, 자료 영역 및 자료 대기행렬을 분석하기 위해 사용이 쉬운 그래픽 인터페이스를 제공합니다. Analyzer를 이용하여 어플리케이션 속도 저하의 원인을 알아낼 수 도 있습니다.

자세한 정보는 iDoctor for iSeries  웹 사이트를 참조하십시오.

PTDV(Performance Trace Data Visualizer)

PTDV(Performance Trace Data Visualizer)는 iSeries 서버에서 실행 중인 어플리케이션의 성능 분석에 사용할 수 있는 Java 어플리케이션입니다. PTDV는 OS/400 기본 오퍼레이팅 시스템의 Performance Explorer 기능과 함께 작동하여 분석자가 프로그램 흐름을 보고 추적, 작업, 스레드 및 프로시저어별로 요약된 세부사항(예: CPU 시간, 현재 시스템 시간, 순환 수, 명령어 수)을 얻을 수 있도록 합니다. Java 어플리케이션 추적을 가시화할 때 추가 세부사항(작성된 오브젝트의 수 및 유형)과 Java 잠금 작동에 대한 정보를 표시할 수 있습니다. 또한 WebSphere Application Server가 생성하는 Performance Explorer 이벤트에 대한 지원도 들어 있습니다. PTDV를 이용하면 열의 정렬, 자료 내보내기 및 다양한 레벨의 자료 요약도 가능합니다.

추가 정보는 Performance Trace Data Visualizer  웹 사이트를 참조하십시오.

성능 관리 API

성능 관리 API를 이용하면 콜렉션 서비스, 성능 콜렉터, Performance Explorer 및 Performance Management/400(PM/400)을 사용하여 성능 자료를 수집하고 관리할 수 있습니다.

성능 관리 API에는 다음이 있습니다.

- 콜렉션 서비스 API
- 성능 콜렉터 API
- Performance Explorer(PEX) API
- Performance Management/400(PM/400) API

OS/400 성능 명령에 대한 “작업”

OS/400에는 문자 기반의 인터페이스에서 성능 자료의 실시간 모니터링을 수행할 수 있는 명령이 많이 포함되어 있습니다. 이들 명령을 사용하여 시스템 성능에 대한 특정 질문에 응답하고 시스템의 성능을 조정할 수 있습니다. iSeries Navigator에서의 실시간 모니터링에 대한 정보는 iSeries Navigator 모니터를 참조하십시오.

명령	기능
WRKACTJOB(활동 작업에 대한 작업)	사용자 시스템에서 실행 중인 작업의 속성과 자원 활용을 검토하고 변경할 수 있습니다.
WRKDSKSTS(디스크 상태에 대한 작업)	시스템 디스크 장치의 성능 정보와 속성을 표시합니다.
WRKSYSSTS(시스템 상태에 대한 작업)	현재 시스템 활동의 개요를 제공합니다. 특히 시스템의 작업 수와 기억장치 풀 활용 정보를 표시합니다.
WRKSYSACT(시스템 활동에 대한 작업)	시스템 상의 작업과 태스크에 대해 작업합니다. 이 명령은 성능 분석 툴 사용권 프로그램(PT1)의 일부입니다.
WRKOBJLCK(오브젝트 잠금에 대한 작업)	적용 대기 중인 잠금 등 지정된 오브젝트에서의 잠금에 대해 작업하고 표시합니다.
WRKSHRPOOL(공유 기억장치 풀에 대한 작업)	활용 정보를 표시하고 기계와 기본 풀 등 공유 기억장치 풀의 속성을 변경합니다.

Extended Adaptive Cache

주: 피쳐 코드 #4331 및 #6831(CCIN #6731)는 시장에서 철수되었습니다. 여기에 제공되는 정보는 기존 사용자를 위한 참조입니다. iSeries 기억장치 I/O의 캐시 피쳐에 대한 향후 확장 기능은 이들 페이지를 통해 제공됩니다.

Extended Adaptive Cache를 사용하여 iSeries 시스템 성능을 개선하십시오! Extended Adaptive Cache는 디스크에서 읽혀지는 실제 I/O 요구의 수를 줄임으로써 I/O 서브시스템과 시스템 응답 시간을 개선하는 고급 캐시 읽기 기술입니다. Extended Adaptive Cache는 자료에 대한 통계 정보를 생성한 후 관리 전략을 혼합 사용하여 캐시할 자료를 판별합니다. Extended Adaptive Cache는 여러 가지 유형의 작업부하에서도 고효율이 입증되었습니다.

IBM은 Extended Adaptive Cache가 iSeries 컴퓨팅 환경에서 제공할 수 있는 이점을 판별할 수 있는 혁신적 툴을 제공합니다. **Extended Adaptive Cache Simulator**는 iSeries Navigator 내의 콜렉션 서비스 기능을 통해 활성화됩니다. Simulator는 일정 시간 동안 실제 작업부하에 대해 에뮬레이트된 성능 결과를 디스크 단위로 보여줍니다. Extended Adaptive Cache Simulator는 I/O 기억장치 어댑터 레벨에서 수행되며 Extended Adaptive Cache를 관리하는 것과 같은 알고리즘을 사용합니다.

더 자세히 배우려면 다음을 읽으십시오.

- **Extended Adaptive Cache 개념**

Extended Adaptive Cache를 탐색하십시오. 이 툴을 사용하기 전에 계획, 제한사항 및 중요한 고려사항에 대한 정보를 찾아 보십시오.

- **Extended Adaptive Cache Simulator**

계산 환경에서 Extended Adaptive Cache가 제공할 수 있는 응답 시간 개선점을 판별하려면 Extended Adaptive Cache 사용법에 대해 배우십시오.

- **Extended Adaptive Cache 확보**

Extended Adaptive Cache Simulator를 사용하여 이 툴이 사용자 환경에 제공할 수 있는 이점을 확인한 후 Extended Adaptive Cache의 입수 방법을 알아보십시오.

Extended Adaptive Cache 개념

Extended Adaptive Cache로 시스템 성능 개선 - 디스크에서 읽혀지는 실제 I/O 요구의 수를 줄임으로써 I/O 서브시스템과 시스템 응답 시간을 개선하는 고급 캐시 읽기 기술. Extended Adaptive Cache는 데이터베이스 읽기 조치뿐 아니라 모든 읽기 조치의 성능을 향상시킵니다. 여기에는 다른 시스템 구성요소(예: Integrated xSeries Server)가 생성하는 읽기 조치가 포함됩니다. 또한 장치 패리티 보호 또는 이중 복사 보호를 갖는 기억장치 서브시스템에서도 효과적으로 작용합니다. Extended Adaptive Cache는 여러 가지 유형의 작업부하에서도 고효율이 입증되었습니다.

Extended Adaptive Cache의 작동 방식

Extended Adaptive Cache는 iSeries I/O 서브시스템에 통합됩니다. 이는 디스크 서브시스템 제어기 레벨에서 작동하며 iSeries 시스템 프로세서에는 영향을 주지 않습니다. 기억장치 I/O 어댑터는 솔리드 상태 디스크와 같은 RCD(Read Cache Device)를 사용하여 Extended Adaptive Cache를 관리하여 캐시 메모리를 제공합니다.

Extended Adaptive Cache는 자료에 대한 통계 정보를 생성한 후 관리 전략을 혼합 사용하여 캐시할 자료를 판별합니다. 캐시 관리는 I/O 어댑터 내에서 자동으로 수행되며 예측 알고리즘을 사용하여 자료를 캐시하기 위해 고안되었습니다. 알고리즘에서는 호스트가 미리 정해진 자료 범위에 액세스하는 빈도를 고려합니다.

Extended Adaptive Cache의 설계는 iSeries 서버의 특정한 자료 관리 전략에 기반합니다. 디스크가 장치 패리티 보호이든, 이중복사되든 또는 보호되지 않든지에 관계 없이 디스크에 저장된 자료는 대역에서 발생합니다. 이는 자료가 활발하게 읽히는 실질적인 연속 디스크 기억장치 영역, 빈번하게 기록되는 실질적인 연속 영역, 읽고 쓰기 활동이 활발하게 일어나는 실질적인 연속 영역 또는 빈번하게 액세스되지 않는 실질적인 연속 기억장치 영역이 있음을 의미합니다.

Extended Adaptive Cache 설계에서 이러한 자료의 “대역”이 고려됩니다. 목표는 읽기/쓰기 및 읽기 전용으로 특화된 대역을 캐시하는 것입니다. 쓰기 전용으로 특화된 대역은 기억장치 서브시스템 쓰기 캐시에 캐시되며 Extended Adaptive Cache의 영향을 크게 받지 않습니다. 또한 Extended Adaptive Cache는 순차적으로 쓰여지거나 순차적으로 읽히는 대형 자료 블록의 성능에 영향을 주지 않도록 설계되었습니다. 이 예에서 디스크의 프리페치 기능과 시스템의 기타 캐시는 빠른 응답 시간을 보장합니다.

더 자세히 배우려면 다음을 읽으십시오.

- **Extended Adaptive Cache의 제한사항 및 고려사항**

Extended Adaptive Cache가 필요로 하는 구성요소를 알아 보고 어떤 것을 예상할 수 있는지 익히십시오.

- **Extended Adaptive Cache Simulator**

Extended Adaptive Cache Simulator의 사용 방법을 익혀 사용자의 컴퓨팅 환경에서 Extended Adaptive Cache가 제공할 수 있는 응답 시간 개선 정도를 판별하십시오.

- **Extended Adaptive Cache 시작**

Extended Adaptive Cache를 활성화하는 방법을 익히십시오.

제한사항 및 고려사항 - Extended Adaptive Cache: Extended Adaptive Cache를 사용하기 전에 사용자의 컴퓨팅 환경과 관련될 수 있는 고려사항이나 제한사항에 대한 초기 계획을 수립해야 합니다.

제한사항

Extended Adaptive Cache를 사용하려면 다음이 시스템에 있어야 합니다.

- Extended Adaptive Cache를 지원하는 하나 이상의 기억장치 I/O 어댑터 (V4R4 이상을 실행하는 시스템의 경우 CCIN 2748, V5R1 이상을 실행하는 시스템의 경우 CCIN 2778, 최신 릴리스 V5R2를 실행하는 시스템의 경우 CCIN 2757(정보 APAR III3365 참조).
- Extended Adaptive Cache를 활성화할 각 기억장치 I/O 어댑터에 대한 RCD(Read Cache Device)(V4R4 이상을 실행하는 시스템의 경우 CCIN 6731)
- Performance Tools for iSeries 사용권 프로그램

Extended Adaptive Cache는 RCD를 통해 자동으로 작동 가능하게 됩니다. 제어식의 On 또는 Off 스위치는 없습니다. RCD는 동시 유지보수를 통해 시스템 중단 없이 추가됩니다. RCD는 내부 디스크 슬롯에 상주하며 다른 모든 디스크 유형과 용량에서 작동합니다. Extended Adaptive Cache의 모든 자료는 디스크에 저장됩니다. RCD 실패와 같은 있을 것 같지 않은 이벤트에도 자료 유실은 없습니다.

I/O 어댑터 아래의 다른 디스크에 대한 장치 패리티 보호 및 이중복사 보호와 관련하여 Extended Adaptive Cache를 사용하는 데에는 제한이 없습니다. 그러나 Extended Adaptive Cache를 동일한 I/O 어댑터에서 Integrated Hardware Disk Compression과 함께 사용할 수는 없습니다. 마지막으로 Extended Adaptive Cache는 iSeries Expert Cache를 보충하기 위해 특별히 고안된 것으로 iSeries Expert Cache와 함께 또는 이것 없이 사용할 수 있습니다.

고려사항

Extended Adaptive Cache를 사용하면 대다수 환경에서 I/O 응답 시간은 크게 저하하고 시스템 I/O 처리량은 증가합니다. 대부분의 캐시에서처럼 시스템 구성과 작업부하는 Extended Adaptive Cache의 효율성에 영향을 줍니다. Extended Adaptive Cache는 기억장치 서브시스템 레벨에서 수행됩니다. 이는 해당 특정 서브시스템 내에 있는 디스크 세트에 대한 자료를 캐시합니다. 따라서 시스템 내의 가장 활동적이고 성능이 중요한 기억장치 서브시스템에 Extended Adaptive Cache를 추가하는 것이 합당합니다. Extended Adaptive Cache는 프리 페치 유형 캐시로 간주되지 않으므로 디스크의 사전 읽기 기능을 방해하지 않습니다.

활발하게 I/O 요구를 수신하고 있는 디스크 기억장치의 영역이 클수록 더 많은 선택적 Extended Adaptive Cache가 새 자료를 캐시로 가져갈 시기를 결정하게 됩니다. 이와 같은 적응 기능이 있어서 Extended Adaptive

Cache는 많은 작업부하 유형과 크기에서 효율적이 될 수 있습니다. 전체적인 캐시 효율은 Extended Adaptive Cache Simulator의 사용을 통한 이러한 관점에서 가장 잘 이해할 수 있습니다.

또한 Extended Cache Simulator와 Extended Adaptive Cache는 같은 기억장치 I/O 어댑터에서 동시에 활성화될 수 없습니다.

이러한 제한사항과 고려사항에 대해 알았으면 Extended Adaptive Cache를 시작할 준비가 되었습니다.

Extended Adaptive Cache 시작: Extended Adaptive Cache를 시작하여 시스템의 성능을 향상시키려면 RCD(Read Cache Device)를 구입하십시오. RCD를 서버시스템의 디스크 슬롯에 삽입하면 Extended Adaptive Cache가 활성화됩니다. 사용자 제어의 on 또는 off 스위치는 없습니다. Extended Adaptive Cache가 자료 흐름을 모니터링하여 RCD를 이식하는 데에는 약 한 시간이 걸립니다. Extended Adaptive Cache를 한 시간동안 실행하고 나면 시스템은 개선된 성능(현재의 작업부하에 따름)과 증가된 I/O 처리량을 보여줄 것입니다.

사용자의 iSeries 시스템이 Extended Adaptive Cache를 사용할 수 있는지 알아보려면 Extended Adaptive Cache에 대한 제한사항 및 고려사항을 참조하십시오.

Extended Adaptive Cache Simulator

주: 피쳐 코드 #4331 및 #6831(CCIN #6731)는 시장에서 철수되었습니다. 여기에 제공되는 정보는 기존 사용자를 위한 참조입니다. iSeries 기억장치 I/O의 캐시 피쳐에 대한 향후 확장 기능은 이들 페이지를 통해 제공됩니다.

Extended Adaptive Cache Simulator를 사용하여 Extended Adaptive Cache를 통한 시스템 성능 개선을 평가할 수 있습니다. Extended Adaptive Cache Simulator는 Extended Adaptive Cache가 사용자 시스템에서 제공할 수 있는 응답 시간 개선 정도를 판별할 수 있는 성능 분석 툴입니다. 이러한 판별은 시스템 구성과 자료 작업부하에 기반할 수 있으며 RCD(Read Cache Device) 구입 이전에 수행됩니다.

Extended Adaptive Cache Simulator는 콜렉션 서비스 내에서 제어되며 CCIN 2757 I/O 어댑터가 있는 최신 릴리스 V5R2를 실행하는 시스템에서 사용할 수 있습니다. (CCIN 2748, CCIN 2778 및 CCIN 2757 I/O 어댑터는 Extended Adaptive Cache 자체를 지원하는 동일한 기억장치 I/O 어댑터입니다.) Simulator는 유연성이 있어서 사용자는 자신의 특정 시스템 및 작업부하 요구에 가장 잘 맞는 용량을 더욱 잘 판별하기 위해 여러 가지 캐시 용량을 에뮬레이트할 수 있습니다. 실제 RCD(Read Cache Device) 용량은 1600MB입니다.

Extended Adaptive Cache Simulator의 활성화를 통해 수집된 성능 정보는 Extended Adaptive Cache의 사용으로 절약할 수 있는 디스크 읽기 조작의 수를 나타내줍니다. 성능 자료는 디스크 액세스 시간에서의 잠재적 개선을 반영합니다.

Extended Adaptive Cache 확보

Extended Adaptive Cache Simulator의 성능 자료를 입수한 후 Extended Adaptive Cache를 사용하여 시스템의 성능을 개선하기로 결정하면 RCD(Read Cache Device)를 구입해야 합니다. Extended Adaptive Cache는 RCD를 통해 자동으로 작동 가능하게 됩니다.


Extended Adaptive Cache를 사용하려면 다음이 있어야 합니다.

- Extended Adaptive Cache를 지원하는 하나 이상의 기억장치 I/O 어댑터 (V4R4 이상을 실행하는 시스템의 경우 CCIN 2748, V5R1 이상을 실행하는 시스템의 경우 CCIN 2778, 최신 릴리스 V5R2를 실행하는 시스템의 경우 CCIN 2757).
- Extended Adaptive Cache를 활성화할 각 기억장치 I/O 어댑터에 대한 RCD (V4R4 이상을 실행하는 시스템의 경우 CCIN 6731).

Extended Adaptive Cache는 RCD를 통해 자동으로 작동 가능하게 되므로 제어식 on 또는 off 스위치가 없습니다. RCD는 동시 유지보수를 통해 시스템 중단 없이 추가됩니다. RCD는 내부 디스크 슬롯에 상주하며 다른 모든 디스크 유형과 용량에서 작동합니다. Extended Adaptive Cache의 모든 자료는 디스크에 저장됩니다. RCD 실패와 같은 있을 것 같지 않은 이벤트에도 자료 유실은 없습니다.

RCD는 iSeries 하드웨어를 판매하는 곳에서 구입할 수 있습니다. 그렇지 않으면 현재의 IBM 영업대표에게 문의하십시오.

iSeries용 작업부하 예측기

Workload Estimator  는 특정 작업부하 유형에 대해 측정된 작업부하에 근거하여 시스템 요구를 조정할 수 있도록 도와줍니다. PM/400은 프로세서 보증 또는 IBM 유지보수 계약을 체결한 사용자가 추가 비용 없이 활성화할 수 있는 통합된 OS/400 기능입니다. 사용자는 시스템 확장과 성능을 계획 및 관리할 때 유용한 용량 및 성능 분석 그래프를 제공합니다.


작업부하 예측기 및 PM/400은 서로에 대해 작동하도록 개선되었습니다. 웹 기반 어플리케이션을 통해 PM/400이 보고한 기존 시스템의 활용, 성능 및 성장을 수용할 수 있는 iSeries 시스템에 맞게 업그레이드를 조정할 수 있습니다. 추가 옵션으로, 크기 조정에는 특정 어플리케이션(예: Domino, Java, Websphere)을 추가하기 위한 용량과 여러 AS/400 또는 iSeries OS/400 워크로드를 한 시스템에 통합하는 것도 포함할 수 있습니다. 이 기능을 이용하여 사용자 시스템에서 발생하는 기존 활용 자료에 근거한 향후의 시스템 요구사항을 계획할 수 있습니다.

무선 iSeries Navigator

무선 iSeries Navigator를 이용하면 인터넷폰, 무선 모뎀이 장착된 PDA 또는 종래의 웹 브라우저를 사용하여 시스템 성과 상태를 원격으로 모니터링할 수 있습니다. 무선 장치가 있으면 다음을 수행할 수 있습니다.

- 여러 시스템에 걸쳐 명령을 실행합니다.
- 시스템, 작업 및 메시지 모니터를 시작하고 봅니다.
- 모니터에서 작업과 메시지에 대해 작업합니다(보류, 해제, 종료, 응답, 세부사항 확보).
- 통합 xSeries 서버를 관리합니다.


무선 iSeries Navigator를 이용하여 무선 모니터링을 시작하는 방법에 대해서는 무선 iSeries Navigator 주제를 참조하십시오.

리모트 모니터링에 대한 완벽한 최신 정보를 얻으려면 무선 iSeries Navigator  홈 페이지를 참조하십시오.

PATROL for iSeries(AS/400) - Predict

PATROL for iSeries(AS/400) - Predict 제품은 다수의 성능 관리 작업을 자동화하여 iSeries 성능 관리를 도와줍니다. 이 제품은 iSeries 서버 그룹에 대해 현재와 이력 시스템 자료의 상세 보기를 제공하여 CPU, 파일 활용 및 상태와 같은 특정 세부사항을 파악할 수 있도록 합니다. 또한 Patrol은 iSeries 서버에서 성능 및 가용성 문제가 발생하기 이전에 이들을 해결할 수 있도록 여러 가지 자동화 옵션을 제공합니다.

이 제품은 자세한 용량 계획을 제공하여 향후의 업그레이드를 계획하고 iSeries 환경의 성장을 관리할 수 있도록 돕습니다.

자세한 정보는 BMC 제품 웹 사이트  를 참조하십시오.

시나리오: 성능

성능 관리에 대해 배울 수 있는 가장 좋은 방법 중 하나는 기업 환경에서 이들 어플리케이션이나 기능이 어떻게 사용되는지를 보여주는 예를 참조하는 것입니다. 다음 시나리오와 구성 예를 보고 성능 관리에 대해 더 자세히 배우십시오.

시나리오: 업그레이드 또는 마이그레이션 후 시스템 성능 개선

이 시나리오에서는 시스템을 막 업그레이드 또는 마이그레이트하였고 이전 보다 실행 속도가 느려진 것처럼 보입니다. 이 시나리오는 성능 문제를 식별하고 수정하도록 돕습니다.

시나리오: 시스템 모니터

CPU 활용도가 너무 높아지고, 더 많은 자원을 사용할 수 있게 될 때까지 일시적으로 낮은 우선순위의 작업을 보유하고 있는 경우 사용자에게 경고하는 예제 시스템 모니터를 참조하십시오.

시나리오: 메시지 모니터

iSeries 서버에서 발생하는 메시지 대기행렬에 대한 조회 메시지를 표시하는 예제 메시지 모니터를 참조하십시오. 모니터가 열리고 감지된 메시지가 바로 표시됩니다.

시나리오: 작업 모니터

지정된 작업의 CPU 활용을 추적하여 CPU 활용도가 너무 높을 경우 작업의 소유자에게 경고하는 예제 작업 모니터를 참조하십시오.

관련 정보

아래에는 성능 주제와 관련된 PDF 형식의 iSeries 매뉴얼(“백서”라고도 함) 및 IBM 레드북^(TM)이 나열되어 있습니다. 또한 다음 PDF도 무엇이든 열람하거나 인쇄할 수 있습니다.

- 매뉴얼

Performance Tools for iSeries

이 책에서는 시스템, 작업 또는 프로그램 성능에 대한 자료를 수집하는 데 필요한 정보를 프로그래머에게

제공합니다. 또한 이 책에는 있을 수 있는 비효율성을 식별하고 수정하기 위해 성능 자료를 인쇄하고 분석하기 위한 추가 정보와 관리자 및 에이전트 피처에 대한 정보도 들어 있습니다.

- 웹 사이트

iSeries Performance Capabilities Reference

이 참조서는 성능 벤치마킹, 용량 계획 및 서버 성능 계획에 유용한 기술적인 정보를 제공합니다.

- 레드북:

- **IBM eserver iSeries Universal Connection for Electronic Support and Services** 

이 책에서는 범용 연결에 대해 소개합니다. 또한 사용자의 시스템 자료에 기초하여 개별화된 전자 지원을 얻을 수 있도록 사용자 시스템의 소프트웨어 및 하드웨어 고안물을 IBM에 보고하는 다양한 지원 툴의 사용 방법에 대해서도 설명합니다.

- **Lotus Domino for AS/400: Performance, Tuning, and Capacity Planning** 

이 문서에서는 성능 관리 방법에 대해 설명합니다. 여기에는 성능 목표 설정, 성능 자료 수집 및 검토 자원 조정 그리고 용량 계획 등이 포함됩니다. 성능 지침 및 어플리케이션 설계에 대한 추가 정보도 제공됩니다.

- **AS/400 Performance Management** 


이 문서에서는 성능 관리 방법에 대해 설명합니다. 여기에는 성능 목표 설정, 성능 자료 수집 및 검토 자원 조정 그리고 용량 계획 등이 포함됩니다. 성능 지침 및 어플리케이션 설계에 대한 추가 정보도 제공됩니다.

- **AS/400 HTTP Server Performance and Capacity Planning** 


인터넷 및 웹 브라우저 기반 어플리케이션은 조직이 정보를 분배하고 기업 운영 프로세스를 수행하고 고객에게 서비스하고 새로운 시장을 개척하는 방법에 지대한 영향을 미칩니다. 이 책은 iSeries 프로그래머, 네트워크 및 시스템 관리 전문가 그리고 웹 기반 어플리케이션 및 정보 시스템을 설계, 개발 및 전개하는 책임을 진 기타 전보 기술자를 위한 책입니다.

- **Java and WebSphere Performance on IBM eserver iSeries Servers** 

이 책은 iSeries 서버에 초점을 두고 Java 및 WebSphere Application Server 성능 관련 사안에 대해 작업하기 위한 추가 정보, 기술 및 방법을 제공합니다.

- **Management Central: A Smart Way to Manage AS/400 Systems** 

중앙 관리의 장점을 경험해 보십시오. 이 책에서는 성능 모니터를 대체할 수 있는 콜렉션 서비스에 대해 설명합니다. 또한 오퍼레이터와 관리자는 중앙 관리를 사용하여 이벤트 통지 및 이벤트에 대한 자동 응답과 같은 실시간 성능 모니터 기능을 제공함으로써 네트워크의 서버를 모니터할 수 있습니다.

– **Managing AS/400 V4R4 with Operations Navigator** 

이 책에서는 Operations Navigator(지금의 iSeries Navigator)를 통해 사용할 수 있는 폭넓은 iSeries 기능을 살펴봅니다. 이들 기능에 대한 자세한 설명과 함께 특히 중앙 관리 및 콜렉션 서비스를 사용하는 시스템 성능 모니터에 대해 설명합니다.

– **AS/400 Performance Explorer Tips and Techniques** 

이 책에서는 V3R6에서 사용할 수 있었던 Performance Explorer 기능에 대한 설명과 자세한 예를 제공합니다. 특정한 어플리케이션 예와 보고서가 제공됩니다.

iSeries 성능에 대한 모든 정보는 성능 주제를 참조하십시오.



Printed in U.S.A.