IBM SPSS Analytic Server
버전 3.2.1

관리자 안내서

IBM

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 27 페이지의 『주의사항』에 있는 정보를 확인하십시오.

제품 정보

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM SPSS Analytic Server의 버전 3, 릴리스 2, 수정사항 1 및 모든 후속 릴리스와 수정에 적용됩니다.

목차

제 1 장 테넌트 관리 1
이름 지정 규칙 3

제 2 장 사용자 시작 5

제 3 장 Analytic Server 작업 이름 7

제 4 장 IBM SPSS Analytic Server 우수 사례 및 권장사항 9

제 5 장 문제 해결 19
로그 기록. 19
버전 정보. 19
로그 콜렉터 20
일반 문제. 20
성능 튜닝. 23

주의사항 27
상표. 29

제 1 장 테넌트 관리

테넌트는 테넌트 간에 오브젝트를 공유할 수 없도록 사용자, 프로젝트 및 데이터 소스의 상위 레벨 구획을 제공합니다. 각 사용자는 지정된 테넌트의 컨텍스트에서 시스템에 액세스합니다.

Analytic Server 콘솔에서 테넌트를 관리하고 테넌트에 사용자를 지정합니다. 테넌트의 보기 페이지는 콘솔에 로그인한 사용자의 역할에 따라 다르게 표시됩니다.

- 설치 중 설정된 "수퍼유저" 관리자가 테넌트 관리자입니다. 이 사용자만 새 테넌트를 작성하고 테넌트 특성을 편집할 수 있습니다.
- 관리자 역할의 사용자는 로그인한 테넌트의 특성을 편집할 수 있습니다.
- 사용자 역할의 사용자는 테넌트 특성을 편집할 수 없습니다. 이 사용자에게는 테넌트 페이지가 표시되지 않습니다.
- 독자 역할이 있는 사용자는 데이터 소스를 편집하거나 Analytic Server 콘솔에 로그인할 수 없습니다.

관리자는 프로젝트 및 데이터 소스 페이지에 액세스하고 정리 및 관리를 위해 프로젝트 또는 데이터 소스를 관리할 수 있습니다. 자세한 정보는 *IBM® SPSS® Analytic Server* 사용자 안내서를 참조하십시오.

테넌트 목록

기본 테넌트 페이지에는 기존 테넌트가 테이블에 표시됩니다. "수퍼유저" 관리자만 이 페이지를 편집할 수 있습니다.

- 세부사항을 표시하고 특성을 편집하려면 테넌트의 이름을 클릭하십시오.
- 해당 테넌트의 컨텍스트에서 콘솔을 열려면 테넌트의 URL을 클릭하십시오.

  참고: 콘솔에서 로그아웃되면 테넌트에 유효한 신임 정보를 사용하여 로그인해야 합니다.
- 검색 문자열이 포함된 테넌트만 표시하려면 검색 영역을 입력하여 목록을 필터링하십시오.
- **새 테넌트 추가** 대화 상자에 지정한 이름으로 새 테넌트를 작성하려면 **새로 만들기**를 클릭하십시오. 테넌트에 지정할 수 있는 이름에 대한 제한사항은 3 페이지의 『이름 지정 규칙』의 내용을 참조하십시오.
- 선택한 테넌트를 제거하려면 **삭제**를 클릭하십시오.
- 목록을 업데이트하려면 **새로 고치기**를 클릭하십시오.

개별 테넌트 세부사항

컨텐츠 영역은 접을 수 있는 여러 섹션으로 나뉩니다.

세부사항

이름 테넌트의 이름을 표시하는 편집 가능한 텍스트 필드입니다.

설명 테넌트에 대한 설명 텍스트를 제공할 수 있도록 하는 편집 가능한 텍스트 필드입니다.

URL Analytic Server 콘솔을 통해 테넌트에 로그인하고 SPSS Modeler 서버를 구성하는 데 사용할 수 있도록 사용자에게 제공하기 위한 URL입니다. SPSS Modeler 구성에 대한 세부사항은 *IBM SPSS Analytic Server* 설치 및 구성 안내서를 참조하십시오.

상태 **활성** 테넌트는 현재 사용 중입니다. 테넌트를 **비활성**으로 지정하면 해당 테넌트에 로그인할 수 없으며 기본 정보를 삭제할 수 없습니다.

프린시펄

프린시펄은 설치 중에 설정한 보안 제공자에게서 얻은 사용자와 그룹입니다. 프린시펄을 관리자, 사용자 또는 독자로 테넌트에 추가할 수 있습니다.

- 텍스트 상자에 입력하면 이름에 검색 문자열이 포함되어 있는 사용자 및 그룹이 필터링됩니다. 드롭 다운 목록에서 **관리자**, **사용자** 또는 **독자**를 선택하여 테넌트 내의 역할을 지정하십시오. 작성자 목록에 참가자를 추가하려면 **참가자 추가** 단추를 클릭하십시오.
- 참가자를 제거하려면 멤버 목록에서 사용자 또는 그룹을 선택하고 **참가자 제거**를 클릭하십시오.

메트릭

테넌트의 자원 한계를 구성할 수 있습니다. 현재 테넌트에서 사용하는 디스크 공간을 보고합니다.

- 테넌트의 최대 디스크 공간 할당량을 설정할 수 있습니다. 이 한계에 도달하면 디스크 공간을 비워서 테넌트 디스크 공간 사용량이 할당량 미만으로 내려가야 이 테넌트의 디스크에 데이터를 작성할 수 있습니다.
- 테넌트의 디스크 공간 경고 레벨을 설정할 수 있습니다. 할당량을 초과하면 디스크 공간을 비워서 테넌트 디스크 공간 사용량이 할당량 미만으로 내려가야만 이 테넌트의 프린시펄에서 분석 작업을 제출할 수 있습니다.
- 이 테넌트에서 한 번에 실행할 수 있는 병렬 작업의 최대 수를 설정할 수 있습니다. 할당량을 초과하면 현재 실행 중인 작업이 완료되어야만 이 테넌트의 프린시펄에서 분석 작업을 제출할 수 있습니다.
- 데이터 소스에서 포함할 수 있는 최대 필드 수를 설정할 수 있습니다. 데이터 소스를 작성하거나 업데이트할 때마다 한계가 선택됩니다.
- 최대 파일 크기(MB)를 설정할 수 있습니다. 파일 업로드 시 한계가 선택됩니다.

보안 제공자 구성

사용자 인증 제공자를 지정할 수 있습니다. **기본**은 설치 및 구성 중에 설정된 기본 테넌트 제공자를 사용합니다. **LDAP**을 사용하면 Active Directory 또는 OpenLDAP과 같은 기존 LDAP 서버에서 사용자를 인증할 수 있습니다. 제공자에 대한 설정을 지정하고 필터 설정을 선택적으로 지정하여 프린시펄 섹션에서 사용 가능한 사용자와 그룹을 제어합니다.

이름 지정 규칙

Analytic Server에서 고유한 이름을 지정할 수 있는 항목(예: 데이터 소스 및 프로젝트)의 경우 다음 규칙이 해당 이름에 적용됩니다.

- 단일 테넌트 내에서 같은 유형의 오브젝트는 서로 이름이 고유해야 합니다. 예를 들어, 두 개의 데이터 소스에 모두 insuranceClaims라는 이름을 지정할 수 없지만 데이터 소스와 프로젝트에 각각 insuranceClaims라는 이름을 지정할 수는 있습니다.
- 이름은 대소문자를 구분합니다. 예를 들어, insuranceClaims와 InsuranceClaims는 각각 고유한 이름으로 취급됩니다.
- 이름의 선행 및 후행 공백은 무시됩니다.
- 다음 문자는 이름에 사용할 수 없습니다.

  ~, #, %, &, *, {, }, \\, :, <, >, ?, /, |, ", \t, \r, \n

제 2 장 사용자 시작

사용자가 http://<host>:<port>/<context-root>/admin/<tenant>로 이동한 후 자신의 사용자 이름과 비밀번호를 입력하여 Analytic Server 콘솔에 로그온하도록 지시하십시오.

참고: Analytic Server 콘솔 로그인 프롬프트에 입력한 사용자 이름이 영역 이름 접미부 없이 입력됩니다. 따라서 다중 영역이 정의된 경우 **영역** 드롭 다운 목록이 사용자에게 표시되며 사용자는 이 목록에서 적합한 영역을 선택할 수 있습니다. 영역이 한 개만 정의된 경우 Analytic Server에 로그인할 때 **영역** 드롭 다운 목록이 사용자에게 표시되지 않습니다.

<host>
Analytic Server 호스트의 주소입니다.

<port>
Analytic Server가 청취하는 포트입니다. 기본적으로 9080입니다.

<context-root>
Analytic Server의 컨텍스트 루트입니다. 기본적으로 analyticserver입니다.

<tenant>
다중 테넌트 환경에서 사용자가 속한 테넌트입니다. 단일 테넌트 환경에서는 기본 테넌트가 **ibm** 입니다.

예를 들어, 호스트 시스템의 IP 주소가 9.86.44.232이며 "mycompany" 테넌트를 작성하여 이 테넌트에 사용자를 추가하고 기타 설정은 기본값으로 남겨 둔 경우 사용자는 http://9.86.44.232:9080/analyticserver/admin/mycompany로 이동하여 Analytic Server 콘솔에 액세스해야 합니다.

제 3 장 Analytic Server 작업 이름

Analytic Server에서는 Hadoop 클러스터의 자원 관리자 사용자 인터페이스를 통해 모니터할 수 있는 맵리듀스 및 Spark 작업을 생성합니다.

맵리듀스 작업 이름은 다음 구조를 가지고 있습니다.

```
AS/{tenant name}/{user name}/{algorithm name}
```

{tenant name}
작업이 실행되는 테넌트 이름입니다.

{user name}
작업을 요청한 사용자입니다.

{algorithm name}
작업의 기본 알고리즘입니다. 단일 스트림에서 여러 맵리듀스 작업을 생성할 수 있습니다. 따라서 단일 맵리듀스 작업에 스트림의 여러 조작이 포함될 수 있습니다.

모든 맵리듀스 작업은 자원 관리자 사용자 인터페이스에 표시됩니다. 단일 Spark 애플리케이션은 각 Analytic Server에 대해 시작됩니다. Spark 애플리케이션의 사용자 인터페이스를 열고 Spark 작업을 모니터하십시오. 작업 이름은 **설명** 열에 표시됩니다.

제 4 장 IBM SPSS Analytic Server 우수 사례 및 권장사항

다음 절에서는 데이터 소스, 클러스터 구성 및 IBM SPSS Modeler 스트림에 대한 Analytic Server 우수 사례 및 권장사항을 제공합니다.

데이터 소스

Analytic Server는 다음과 같은 데이터 소스 유형을 지원합니다.

- 구분자에 의한 배열, 고정된 텍스트 및 Microsoft Excel 파일 등 파일 기반의 데이터 소스
- Db2, Oracle, Microsoft SQL Server, Teradata, Postgres, Netezza, MySQL 및 Amazon Redshift 등의 관계형 데이터베이스
- 모든 기본 제공 데이터 유형(예: ORC 및 Parquet) 및 적절한 Hive 시리얼라이저-디시리얼라이저 구현이 사용 가능한 모든 사용자 정의 데이터 유형을 포함하는 Hive/HCatalog 데이터 소스. 또한 Analytic Server는 HBase, MongoDB, Accumulo, Cassandra, Oracle NoSQL 등의 NoSQL 데이터베이스 및 적절한 Hive 저장 공간 핸들러 구현이 사용 가능한 기타 데이터베이스에 액세스하도록 구성될 수 있습니다.
- 지리적 공간 유형의 데이터 소스(shape 파일 기반 및 맵 서비스 기반).

Hive/HCatalog 데이터 소스에서의 Analytic Server 제한사항

- SPSS Modeler 선택 노드에 Hive 푸시백이 필요한 경우, 필터링 표현식이 STRING 유형의 파티션된 열만 참조할 수 있습니다. Analytic Server 3.0부터는 파티션된 TINYINT, SMALLINT, INT, BIGINT 열에 대해 데이터 유형 지원이 추가되었습니다. Hive 데이터 소스에 대해 지정된 정적 필터링 표현식이 모든 데이터 유형의 파티션된 열에 대해 필터링 표현식을 가질 수 있습니다.
- Analytic Server는 Hive 보기를 지원하지 않습니다.

클러스터 구성 - 보안

Kerberos 위장

버전 3.0.1 미만에서는 Kerberos 보안이 사용 가능한 경우, HDFS 작업을 인증하기 위해 Analytic Server 인스턴스가 Analytic Server 키탭의 사용자 프린시펄 이름을 사용했습니다. 버전 3.0.1부터는 Analytic Server가 Kerberos 위장을 사용하는 HDFS 작업을 인증하기 위해 REST 요청을 작성하는 사용자의 사용자 이름을 요청하면서 Analytic Server 키탭의 서비스 프린시펄 이름을 사용합니다. Analytic Server 3.0.1 이상은 Kerberos 사용 가능 클러스터를 실행할 때 위장 구성 속성을 HDFS 또는 Hive 서비스 구성에 추가해야 합니다. HDFS의 경우, 다음 특성이 HDFS `core-site.xml` 파일에 추가되어야 합니다.

```
hadoop.proxyuser.<analytic_server_service_principal_name> .hosts = *
hadoop.proxyuser.<analytic_server_service_principal_name> .groups = *
```

여기서, <analytic_server_service_principal_name>은 Analytic Server 구성의 Analytic_Server_User 필드에서 지정되는 기본 as_user 값입니다.

데이터가 Hive/HCatalog를 통해 HDFS에서 액세스되는 경우, 다음 특성도 HDFS core-site.xml 파일에 추가되어야 합니다.

```
hadoop.proxyuser.hive.hosts = *
hadoop.proxyuser.hive.groups = *
```

Kerberos 교차 영역 인증

Analytic Server는 Kerberos 교차 영역 인증을 지원합니다. 이 기능을 사용하려면 먼저 KDC 교차 영역 인증을 사용할 수 있는지 확인하고 다음 설정을 Analytic Server Ambari 구성의 **사용자 정의 analytics.cfg** 섹션에 추가해야 합니다.

```
kerberos.user.realm.trim = true
```

클러스터 구성 - 성능 튜닝 설정 및 결과

Spark 구성

Analytic Server는 Hadoop 클러스터에서 YARN과 상호 작용하고 Spark 작업을 실행하기 위해 yarn-client 모드를 사용합니다.

Analytic Server 사용자 정의 설정:

- Ambari 설정은 Analytic Server Ambari 구성의 **사용자 정의 analytics.cfg** 섹션에서 정의됩니다.
- Cloudera 설정은 Cloudera Manager의 **analyticserver-conf/config.properties에 대한 Analytic Server 고급 구성 스니펫(안전 밸브)** 섹션에 있습니다.

1. Analytic Server 사용자 정의 설정에서 구성 항목을 추가하여 **spark.driver.memory** 구성 설정의 값을 늘리는 것을 고려해 보십시오(명시적으로 설정되지 않으면 기본값은 1g임).예를 들어, 다음과 같습니다.

   ```
   spark.driver.memory=2g
   ```

2. 다음의 Spark를 사용하는 Analytic Server 자원 사용 옵션 증 하나에서 선택하십시오.
   - **옵션 A: 정적 자원 할당 구성**

     Analytic Server 사용자 정의 설정에서 다음과 같은 세 개의 매개변수가 구성되어야 합니다.

     ```
     spark.executor.instances
     spark.executor.cores
     spark.executor.memory
     ```

     다음 단계에서는 매개변수 값을 결정하는 방법에 대해 설명합니다.

     a. Analytic Server가 Spark에 대해 영구적으로 할당할 수 있는 CPU 및 메모리 백분율을 설정하십시오. 그러면 특정 코어 수(C) 및 각 시스템에서 사용될 수 있는 고정된 메모리 양(M)을 얻을 수 있습니다.

b. 각 시스템이 실행할 수 있는 실행기의 수(E)를 설정하십시오. 이러한 실행기는 각 클러스터 노드에서 별도의 Hadoop 컨테이너(프로세스)로 실행됩니다. 일반적으로 2보다 큰 값이 적절하나 총 코어 수보다 작은 수여야 합니다. Spark에 대해 할당되는 메모리가 이러한 실행기 사이에 분할되므로 이 매개변수에 대해 높은 값을 선택하면 각 컨테이너에 대해 할당되는 메모리의 양이 줄어듭니다.

c. 각 실행기에 대해 사용되는 코어의 수(CE)를 설정하십시오. 일반적으로 이 값은 C/E(Spark 애플리케이션에 대해 할당되는 각 시스템의 코어 수를 실행기의 총 수로 나눈 값)입니다.

d. 각 실행기에 대해 사용되는 메모리의 양(ME)을 설정하십시오. 이 값은 일반적으로 M/E입니다.

참고: 사용되는 실행기 및 코어의 수는 각 실행기 메모리의 양이 3G * CE보다 커야 하는 식으로 균형을 맞춰야 합니다. 각 실행기의 각 코어에는 저장 공간 또는 계산 메모리로 사용되는 3G 이상의 메모리가 할당되어야 합니다.

```
spark.executor.instances = <E>*N /<E> // value established in step b where N is the number of compute nodes
spark.executor.cores = <CE> // value established in step c
spark.executor.memory = <ME> // value established in step d
```

그림 *1.* 사용자 정의 *analytics.cfg Spark* 설정

- **옵션 B: 동적 자원 할당 구성**

  이 옵션을 사용하면 전체 클러스터의 실제 사용 가능한 자원에 따라 YARN에 의해 할당된 모든 실행기가 동적으로 증가하거나 감소합니다.

  최소 구성은 다음과 같습니다.

  ```
  spark.dynamicAllocation.enabled = true
  spark.shuffle.service.enabled = true
  ```

  일반적인 구성은 다음과 같습니다.

  ```
  spark.default.emitter.class = com.spss.ae.spark.ListEmitter
  spark.default.emitter.compressed = false
  spark.dynamicAllocation.enabled = true
  spark.executor.cores = 4
  spark.executor.memory = 16g
  spark.io.compression.codec = snappy
  spark.rdd.compress = true
  spark.shuffle.service.enabled = true
  ```

  참고:

- spark.executor.instances = <E>는 사용해서는 안되며 사용할 경우 정적 자원 할당이 이용됩니다.
- 실행기 코어 및 메모리 값에 대한 고려사항은 옵션 A와 동일합니다.

3. 다음 설정을 사용하여 Analytic Server 사용자 정의 설정의 Spark 캐시를 사용 안함으로 설정할 수 있습니다.

```
spark.cache=false
spark.storage.memoryFraction = 0.3
```

그림 2. 사용자 정의 *analytics.cfg Spark* 캐시 설정

대형 IBM SPSS Modeler 스트림이 사용되는 경우 Spark 캐시를 사용하지 않도록 설정해야 합니다. 이 경우에 Spark 캐시를 사용하지 않도록 설정하면 실행 중인 스트림이 느려지나 실행기당 지정된 메모리 양이 작은 경우에 발생할 수 있는 메모리 부족 상황을 방지할 수 있습니다.

JVM 구성

Ambari 설정:

1. Analytic Server Ambari 구성에서 서버가 로컬 처리에 사용할 수 있는 메모리의 양을 설정하십시오. 소형에서 중형 스트림에는 기본값(2GB)을 사용해도 안전하나 대형 스트림에는 더 높은 값의 힙 크기(예: 10GB)를 사용해야 합니다.

   Analytic Server > **구성** > **고급 analytic-jvm-options**

2. `-Xmx2048M`을 `-Xmx10G`로 바꾸고 구성을 저장한 다음 Analytic Server를 다시 시작하십시오.

그림 3. 고급 *analytic-jvm-options* 설정

Cloudera 설정:

1. Cloudera Manager에서 Analytic Server 서비스의 **구성** 탭으로 이동한 다음, 서버가 로컬 처리에 사용할 수 있는 메모리의 양을 설정하도록 `jvm-options` 제어를 업데이트하십시오. 소형에서 중형 스트림에는 기본값(2GB)을 사용해도 안전하나 대형 스트림에는 더 높은 값의 힙 크기(예: 10GB)를 사용해야 합니다.

   Analytic Server 서비스 > **구성** > **jvm-options**

2. `-Xmx2048M`을 `-Xmx10G`로 바꾸고 구성을 저장한 다음 Analytic Server를 다시 시작하십시오.

Yarn MapReduce2 구성:

- Analytic Server 실행에 대해 Spark 작업과 병렬로 MapReduce 작업을 실행해야 하는 경우, Yarn 클러스터가 각 Yarn 컨테이너에 대해 4GB 이상의 메모리를 갖도록 구성되어야 합니다.

Zookeeper 구성:

- Cloudera에서는 Zookeeper 구성을 수동으로 업데이트해야 합니다. 추가 정보는 https://www.cloudera.com/documentation/enterprise/5-4-x/topics/cdh_ig_zookeeper_server_maintain.html의 내용을 참조하십시오.
- 복합 SPSS Modeler 스트림 또는 가로형 데이터(많은 수의 필드)를 사용하는 경우, 차단된 Analytic Server–Zookeeper 연결로 인해 작업이 실패하여 문제가 발생할 수 있습니다. 문제는 SPSS Modeler 서버가 Analytic Server로 전송하는 대형 프로그램 크기의 결과입니다. 이 문제가 Analytic Server 3.0 이상에서 발생할 확률은 낮습니다. 다음 단계를 사용하여 이 문제를 해결하십시오.

  1. Ambari 콘솔에서 Zookeeper 서비스 **구성** 탭으로 이동하여 다음 행을 **고급 zookeeper-env** 아래의 `zookeeper-env` 템플리트에 추가하고 Zookeeper 서비스를 다시 시작하십시오.

     ```
     export JVMFLAGS="-Xmx2048m -Djute.maxbuffer=2097152"
     ```

그림 *4. zookeeper-env* 템플리트 설정

  2. Ambari 콘솔에서 Analytic Server 서비스의 **구성** 탭으로 이동하여 다음을 **고급 analytics-jvm-options**에 추가한 다음 Analytic Server 서비스를 다시 시작하십시오.

     ```
     -Djute.maxbuffer=2097152
     ```

그림 *5.* 고급 *analytics-jvm-options* 설정

참고: 문제가 지속되면 두 위치 모두에서 `-Djute.maxbuffer` 값을 `2097152`에서 `4194304`로 늘리십시오.

IBM SPSS Modeler 스트림 권장사항

참고: 다음 권장사항 중 대부분은 소형 데이터에도 적용됩니다.

소형 데이터에 대한 프로토타입

스트림으로 실험하는 경우, 적은 수의 노드를 추가한 상태로 스트림을 테스트하고 노드를 추가하여 일부 테이블 또는 그래픽 출력을 체크아웃한 다음 스트림을 계속 작성하는 경우가 종종 있습니다. 일반적으로 스트림을 테스트할 때마다 대형 데이터를 전달할 수는 없습니다.

대형 데이터의 적절한 데이터 표본을 작성하면 완전한 데이터 전달을 수행할 때 필요한 시간 페널티를 발생시키지 않고 실제 데이터에 대해 스트림을 테스트할 수 있습니다. 데이터 표본은 스트림을 실행하기에 충분한 데이터를 포함해야 합니다. 예를 들어, 미네소타에 있는 저장소의 트랜잭션을 분석할 계획이라면 데이터 표본이 미네소타에 있는 저장소의 트랜잭션을 포함해야 합니다.

표본추출 후 다음 중 하나를 수행할 수 있습니다.

- 대형 데이터가 상주하는 클러스터에 데이터 표본의 캐시 작성

  장점 - 단순하며 소스 노드 전환이 필요하지 않음

  단점 - 세션이 종료될 때 캐시가 사라짐
- 데이터 표본을 포함하는 새 Analytic Server 데이터 소스 작성

  장점 - 영구적인 데이터 소스

  단점 - 소스 노드 종료/전환이 필요함
- 데이터 표본을 로컬 시스템에 다운로드하고 로컬 데이터 소스 작성

  장점 - 프로토타입 생성 시 클러스터 자원을 이용하지 않으며 소형 데이터에 대해 작업할 때 SPSS Modeler 클라이언트가 Analytic Server 보다 효율적임

  단점 - 소스 노드 전환이 필요함

소스 노드에서 별도의 유형 및 필터 노드 작성

모든 SPSS Modeler 소스 노드 또한 필터 및 유형 노드가 결합된 기능을 갖고 있습니다. 이는 캔버스를 간소화하는 데 유용하나 다른 소스 노드 유형으로 전환하는 것을 어렵게 만듭니다. 또한 유형 및 필터 작업이 발생하는 것을 알기 힘듭니다.

필터 및 선택 노드를 가능한 한 소스 노드와 가까이 배치

이로 인해 다운스트림 작업에서 레코드 수를 줄일 수 있습니다.

가능한 한 정렬 노드 피하기

Analytic Server는 정렬되는 데이터에 의존하는 노드(합치기 노드 등)에서 최적화를 지원하지 않습니다. 따라서 중간 스트림 정렬 노드는 거의 쓸모가 없습니다. 정렬 노드는 최상위 N 또는 최하위 N 레코드를 가져오기 위해 표본 노드 바로 앞에서 사용될 때 유용합니다.

사용될 필드의 변수만 계산

필드의 변수를 계산하고 바로 필터링하지 마십시오.

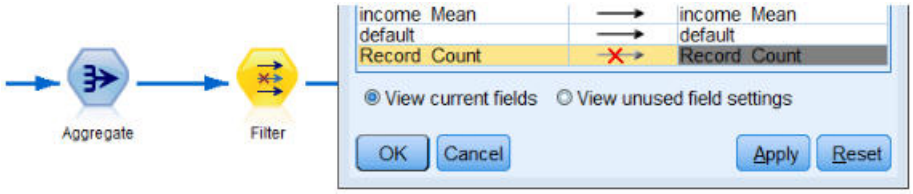

그림 *6. Modeler* 필드 옵션

가능한 한 이해하기 어려운 표현식을 작성하지 말고 수많은 임시 필드를 작성하는 것을 피하십시오. 예를 들어, 다음과 같이 예를 정의하는 것을 피하십시오.

```
now = datetime_now()
birthdate = datetime_date(bYear, bMonth, bDay)
age = date_years_difference(birthdate, now)
```

대신 다음과 같이 예를 정의하십시오.

```
age = date_years_difference(datetime_date(bYear, bMonth, bDay), datetime_now())
```

이런 방법으로 임시 표현식을 인라인 표현식으로 중첩하여 많은 수의 필드가 변환될 때 성능을 개선할 수 있습니다.

데이터 소스에서 저장 공간 설정

필드의 저장 유형을 변경하는 작업(예를 들어, 문자열에서 정수로)에서 중간 스트림이 전체 성능을 결정할 수 있습니다. Analytic Server 콘솔에서 데이터 소스를 정의할 때 이러한 변환을 반복하지 않도록 필드에 대한 저장 공간을 설정할 수 있습니다.

소형 데이터에 대해 작업할 때 SPSS Modeler 사용

Analytic Server를 사용하여 대형 데이터를 조작한 다음 SPSS Modeler를 사용하여 소형 데이터에 대한 계산을 완료하십시오.

적절한 Analytic Server 관련 스트림 특성 선택

관련 스트림 특성을 구성하고(**도구** > **옵션** > **스트림 특성** > **Analytic Server**) 노드가 Analytic Server에서 실행될 수 없는 경우에 데이터 처리가 Analytic Server에서 중단되고 SPSS Modeler에서 계속될 수 있는지 결정하십시오.

기본적으로 SPSS Modeler는 이 상황에서 오류를 보고하고 실행을 중단하도록 구성됩니다. 설정을 Error에서 Warn으로 변경하고 SPSS Modeler에서 처리될 수 있는 데이터 양의 한계를 조정하여 오류를 우회할 수 있습니다. 예를 들어, 필요한 경우 데이터 전송률을 기본값인 10000 레코드 값에서 업데이트할 수 있습니다. 이 한계는 SPSS Modeler 테이블 노드를 사용하는 결과를 볼 때도 적용됩니다. 한계가 초과되면 SPSS Modeler에서 데이터 페치가 스트림 특성에서 설정된 한계를 초과했음을 보고합니다.

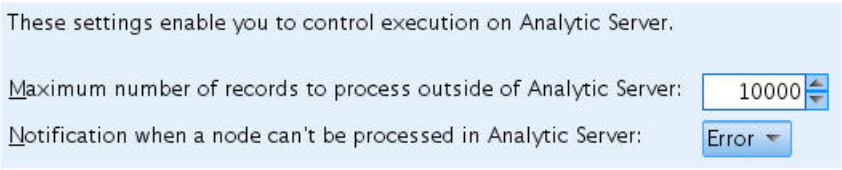

그림 *7. Analytic Server* 설정

Analytic Server 소스 노드 사용

Analytic Server는 다른 데이터베이스 데이터 소스에 연결할 수 있으나 SPSS Modeler의 경우, 전체 스트림이 Analytic Server 작업으로 실행되기 위해 모든 소스 노드가 Analytic Server여야 합니다. 전체 스트림이 Analytic Server에서 실행되려면 데이터베이스 소스 노드가 Analytic Server 소스 노드로 변경되고 Analytic Server 데이터베이스 데이터 소스가 Analytic Server 콘솔에서 작성되어야 합니다.

지원되지 않는 노드를 사용하는 방법 고려

Analytic Server가 모든 노드를 지원하는 것은 아닙니다. 전치 노드가 좋은 예입니다. 전치 작업의 결과를 나머지 스트림과 합쳐서 Analytic Server에서 실행하려면 전치 노드를 포함하는 서브스트림이 Analytic Server 내보내기 노드를 사용하는 Analytic Server 데이터 소스에 작성되어야 합니다. 그런 다음 스트림이 Analytic Server에 기록하기 위해 중단되는 Analytic Server 소스 노드를 첨부해야 합니다.

참고: 전치 작업은 일회성 또는 거의 실행되지 않는 작업에 적합하며 일상적인 스트림 작업에는 사용하지 마십시오.

스트림이 실행되기 전에 Analytic Server에서 작동하는지 여부 판별

스트림이 Analytic Server에서 실행되도록 준비한 후 터미널 노드를 선택하고 SPSS Modeler 미리보기 기능(도구 모음의 **실행 미리보기** 제어)를 사용하여 스트림을 실행하지 않고 터미널 노드 실행과 관련된 모든 노드가 Analytic Server에서 작동하는지 확인하십시오. 메시지 창에 문제가 보고됩니다.

연속 합치기 작업 조합

일련의 합치기 노드가 동일한 키 및 결합 유형을 가진 경우, 단일 노드로 조합하십시오.

동일한 서브스트림 조합

가능한 한 동일한 서브스트림을 조합하도록 시도하십시오. 특히, 합치기 및 정렬과 같이 비용이 많이 드는 조작을 포함하는 경우에 해당됩니다. SPSS Modeler는 이러한 조작을 한 번 수행하며 캐시를 사용하여 성능을 개선합니다. 다음 예에서 스트림이 **newField** 노드까지 동일합니다.

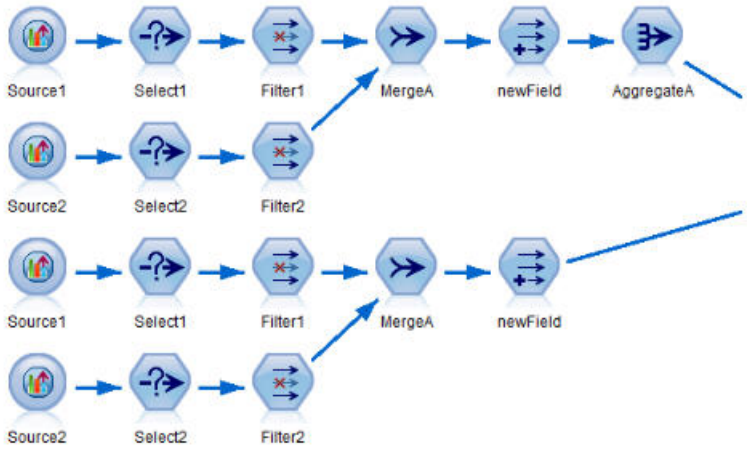

그림 *8.* 스트림 예

서브스트림이 위와 달리 다음과 같은 구조인 경우에 더욱 효과적이며 유지하기 쉽습니다.

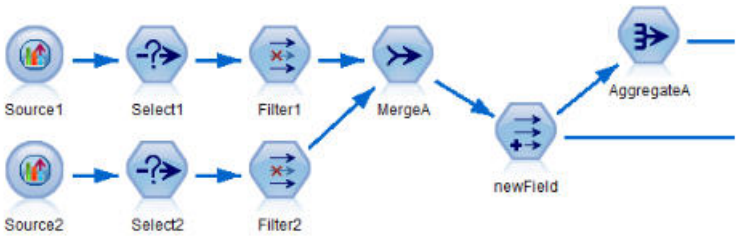

그림 *9.* 스트림 예

추가 유형 노드 제거

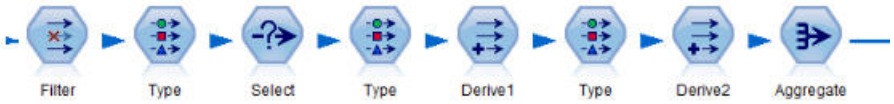

그림 *10.* 스트림 예

Analytic Server에 대해 실행할 때 불필요한 유형 노드를 제거하십시오. 유형 노드의 값 읽기 조작이 MapReduce 작업을 시작합니다. 사용자가 유형 노드 값을 지우지 않는 한, 이는 일반적으로 일회성 저장입니다.

각 스트림 완전 문서화

다음은 수많은 서브스트림을 포함하는 복합 스트림을 표시하는 예입니다.

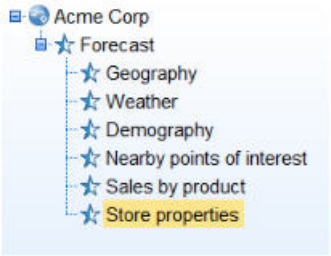

그림 *11.* 서브스트림 예

각 경우마다 수퍼노드의 이름을 적절히 지정하고 코드를 문서화하듯이 스트림을 문서화하는 것이 중요합니다. 정확한 주석은 스트림을 읽거나 유지보수하는 다른 분석가에게 귀중한 정보를 제공할 수 있습니다. 예를 들어, 다음과 같습니다.

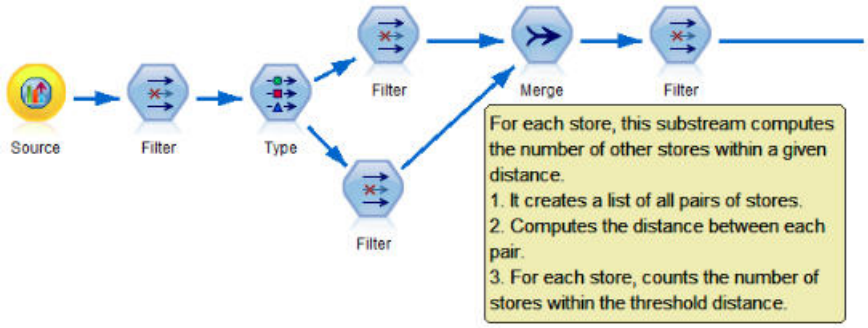

그림 *12.* 주석이 있는 스트림 예

스트림을 개발할 때 SPSS Modeler 캐시를 사용하여 중간 결과를 신속하게 저장

Analytic Server에 대해 실행되는 스트림에서 노드 캐싱은 스트림의 특정 파트의 데이터를 SPSS Modeler 서버에 저장하는 것과 반대로 HDFS의 임시 파일에 저장하여 작업합니다. 캐시가 대형 데이터에 대해 잘 작동하며 Analytic Server에서 실행되는 스트림을 안전하게 사용할 수 있습니다.

제 5 장 문제 해결

Analytic Server에서는 문제점 판별에 도움이 되는 도구를 제공합니다.

로그 기록

Analytic Server에서는 문제점 진단에 도움이 되는 고객 로그 파일과 추적 파일을 작성합니다. 기본 Liberty 설치로 {AS_ROOT}/ae_wlpserver/usr/servers/aeserver/logs 디렉토리에서 로그 파일을 찾을 수 있습니다.

기본 로깅 구성에서는 매일 롤오버되는 두 개의 로그 파일을 생성합니다.

as.log
: 이 파일에는 정보 제공용 경고 및 오류 메시지에 대한 상위 레벨 요약이 포함되어 있습니다. 서버 오류 발생 시 사용자 인터페이스에 표시되는 오류 메시지로 문제가 해결되지 않는 경우 이 파일을 검토하십시오.

as_trace.log
: 이 파일에는 ae.log의 모든 항목이 포함되어 있지만 IBM 지원 및 개발 팀의 디버깅용으로 추가 정보가 제공됩니다.

Analytic Server에서는 Apache LOG4J를 기본 로깅 기능으로 사용합니다. LOG4J를 사용하면 {AS_SERVER_ROOT}/configuration/log4j.xml 구성 파일을 편집해서 로깅을 동적으로 조정할 수 있습니다. 지원 센터에서 문제점 진단을 지원하기 위해 이를 수행할 것을 요청받거나 이를 수정하여 보관할 로그 파일 수를 제한할 수 있습니다. 파일을 변경하면 몇 초 안에 자동으로 감지되므로 Analytic Server를 다시 시작하지 않아도 됩니다.

log4j 및 구성 파일에 대한 자세한 정보는 공식 Apache 웹 사이트(http://logging.apache.org/log4j/)의 문서를 참조하십시오.

버전 정보

{AS_ROOT}/properties/version 폴더를 검사하여 설치된 Analytic Server 버전을 판별할 수 있습니다. 다음 파일에는 버전 정보가 있습니다.

IBM_SPSS_Analytic_Server-*.swtag
: 자세한 제품 정보가 포함되어 있습니다.

version.txt
: 설치된 제품의 버전 및 빌드 번호가 있습니다.

로그 콜렉터

로그 파일을 직접 검토해서 문제를 해결할 수 없으면 모든 로그를 번들로 만들어 IBM 지원 센터에 보내주십시오. 필요한 모든 데이터를 간편하게 수집할 수 있는 유틸리티를 제공합니다.

명령 쉘을 사용하여 다음 명령을 실행하십시오.

```
cd  {AS_ROOT}/bin
run  >sh ./logcollector.sh
```

이러한 명령을 사용하면 {AS_ROOT}/bin에 압축 파일이 작성됩니다. 압축 파일에는 모든 로그 파일과 제품 버전 정보가 포함됩니다.

일반 문제

이 절에서는 몇 가지 일반 관리 문제와 이를 해결하는 방법에 대해 설명합니다.

스트림 실행

R 작업이 영어가 아닌 단어를 유니코드로 변환함

Cloudera 클러스터에서 Hadoop 서버의 시스템 인코딩이 UTF-8이 아닌 경우 R이 영어가 아닌 단어를 유니코드로 변환합니다.

1. Cloudera Manager 콘솔에서 YARN 구성 탭으로 이동하십시오.
2. "NodeManager 환경 고급 구성 스니펫(안전 밸브)" 필드에 다음 설정을 추가하십시오.

```
LC_ALL=""
LANG=en_US.utf8
```

PySpark 작업 실행 실패

Spark 서비스가 모든 Analytic Server 노드 및 모든 노드 관리자에 배포되었는지 확인하십시오.

Kerberos 사용 환경에서 PySpark 작업 실행 실패

PySpark 테스트가 실행되기 전에 kinit 명령을 실행한 후 Analytic Server를 다시 시작해야 합니다. 예를 들어, 다음과 같습니다.

HDP Kerberos

```
cd /etc/security/keytabs/
sudo -u as_user kinit -k -t as_user.headless.keytab as_user/lyrh1.fyre.ibm.com@IBM.COM
```

CDH Kerberos

```
cd /run/cloudera-scm-agent/process/387-analyticserver-ANALYTIC_SERVER
sudo -u as_user kinit -k -t analyticserver.keytab as_user/cdh12-1.fyre.ibm.com@IBM.COM
```

메모리 오류

실행기 메모리 오류 이후에 YARN 구성

필수 실행기 메모리가 최대 임계값을 초과하는 경우 다음 오류가 발생할 수 있습니다.

```
Caused by: com.spss.mapreduce.exceptions.JobException:
    java.lang.IllegalArgumentException: Required executor memory (1024+384  MB) is above the max
    threshold (1024 MB) of this cluster! Please  increase the value of
    'yarn.scheduler.maximum-allocation-mb'.
```

다음은 문제를 해결하는 데 필요한 YARN 구성 설정을 제공하는 단계입니다.

Ambari의 경우

1. Ambari 사용자 인터페이스에서 **YARN** > **구성** > **설정**으로 이동하십시오.
2. 메모리 **노드(모든 YARN 컨테이너에 대해 할당된 메모리)**를 8192MB로 늘리십시오.
3. 컨테이너 값을 늘리십시오.
   - **최소 컨테이너 크기(메모리)**를 682MB로
   - **최대 컨테이너 크기(메모리)**를 8192MB로
4. **최대 컨테이너 크기(VCores)**를 3으로 늘리십시오.
5. YARN, Spark 및 Analytic Server 서비스를 다시 시작하십시오.

Cloudera의 경우

1. yarn.nodemanager.resource.memory-mb를 8GB로 늘리십시오.
   - Cloudera Manager 사용자 인터페이스에서 **Yarn 서비스** > **구성** > **컨테이너 메모리 검색**으로 이동하여 값을 8GB로 늘리십시오.
2. Cloudera Manager 사용자 인터페이스에서 **YARN 서비스** > **빠른 연결**로 이동하여 **동적 자원 풀**을 선택하십시오.
3. **구성** 아래에서 사용 가능한 각 풀에 대해 **편집**을 클릭하고 **YARN** 아래에서 **실행 중인 최대 앱** 값을 4로 설정하십시오.
4. YARN, Spark 및 Analytic Server 서비스를 다시 시작하십시오.

Hadoop 및 Apache Spark 2.x가 함께 있음

- Hadoop 및 Apache Spark 2.x가 동일한 환경에 있는 경우 대부분의 forcespark 및 forcehadoop 작업이 실패합니다. java.lang.NoClassDefFoundError: org/apache/hadoop/fs/FSDataInputStream 오류가 Yarn 애플리케이션 로그에 표시됩니다.

  다음과 같이 /etc/spark2/conf/spark-defaults.conf 파일을 수동으로 편집하여 문제를 해결할 수 있습니다.

  ```
  #spark.hadoop.mapreduce.application.classpath=
  #spark.hadoop.yarn.application.classpath=
  ```

- 두 가지 JDK 버전이 동일한 시스템에 설치되어 있는 경우 Cloudera는 JDK 1.7을 사용하고 Spark 2.x는 JDK 1.8을 사용합니다. Apache Spark 2.x가 설치된 상태에서 forcespark 또는 forcehadoop 작업을 실행하면 모든 작업이 실패할 수 있으며 다음과 같은 오류 메시지가 표시됩니다.

  실행 실패. 이유: org/apache/spark/api/java/function/PairFunction : 지원되지 않는 주.부 버전 52.0

Cloudera의 경우 Cloudera Manager의 **server.env에 대한 Analytic Server 고급 구성 스니펫(안전 밸브)** 섹션에 다음 행을 추가하십시오.

```
JAVA_HOME=/usr/java/jdk1.8.0_152
```

Apache Hive UDF 사용자에게 *admin* 권한 부여

Analytic Server Apache Hive UDF 등록 후에 Invalid function 오류가 발생할 수 있습니다. 기본적으로 Hive 역할은 두 가지입니다(admin 및 public). Hive 사용자는 public 역할에 속합니다. Hive UDF를 사용하려면 등록된 사용자가 admin 권한을 갖고 있어야 합니다(Hive 보안이 사용으로 설정되어 있음).

Hive UDF 사용자에게 admin 권한을 부여하려면 다음을 수행하십시오.

1. Hive로 Beeline에 로그인하십시오.

   ```
   !connect jdbc:hive2://localhost:10000/default;principal=hive/cdh51501.fyre.ibm.com@IBM.COM
   ```

2. Beeline에서 다음 명령을 실행하십시오.

   ```
   grant admin to user hive WITH ADMIN OPTION;
   ```

   참고: 다른 유용한 SQL 명령에는 다음이 포함됩니다.

   hive 사용자에게 이미 지정된 역할을 표시합니다.

   ```
   show role grant user hive;
   ```

   public 역할에 지정된 사용자를 표시합니다.

   ```
   show principals public;
   ```

3. Hive를 다시 시작하고 Analytic Server Hive UDF를 다시 등록합니다.

   ```
   sudo -u hive kinit -k -t hive.keytab hive/cdh51501.fyre.ibm.com@IBM.COM
   sudo -u hive hive -f /opt/cloudera/parcels/AnalyticServer/bin/udfUnregister.sql
   sudo -u hive hive -f /opt/cloudera/parcels/AnalyticServer/bin/udfRegister.sql
   ```

HiveDB 오류

HiveDB에 기록할 때 다음 오류가 발생할 수 있습니다.

```
(AEQAE4805E) Execution failed. Reason: com.google.common.io.Closeables.closeQuietly(Ljava/io/Closeable;)
```

이 오류는 Hadoop Cluster에 guava-*.jar 파일의 버전이 여러 개인 경우 발생합니다. 이 오류는 다음 단계를 수행하여 해결할 수 있습니다(예에서는 HDP 3.1을 사용함).

1. Ambari 콘솔을 열고 Analytic Server 서비스를 중지하십시오.
2. /usr/hdp/3.1.0.0-78/spark2/jars/guava-14.0.1.jar을 {AS_ROOT}/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib에 복사하십시오.
3. Ambari 콘솔에서 Analytic Server 서비스를 새로 고친 후 Analytic Server 서비스를 시작하십시오.

성능 튜닝

이 절에서는 시스템 성능을 최적화하는 방법에 대해 설명합니다.

Analytic Server는 HDFS, Yarn, Spark과 같은 다른 구성요소를 활용하는 Ambari 프레임워크의 구성요소입니다. Hadoop, HDFS 및 Spark의 일반적인 성능 튜닝 기술이 Analytic Server 워크로드에 적용됩니다. Analytic Server 워크로드는 각기 다르므로 특정 배포 워크로드를 기준으로 튜닝 테스트가 필요합니다. 다음 특성과 튜닝 팁은 Analytic Server 벤치마킹 및 스케일링 테스트의 결과에 영향을 미치는 주요 변경 사항입니다.

Analytic Server에서 첫 번째 작업이 실행되면 서버에서 지속적 Spark 애플리케이션이 시작되며, 이 애플리케이션은 Analytic Server가 종료될 때까지 활성화됩니다. 지속적 Spark 애플리케이션은 Analytic Server 작업이 실행 중이지 않더라도 Analytic Server가 실행 중인 동안 해당 애플리케이션에 할당된 모든 클러스터 자원을 할당 및 보존합니다. Analytic Server Spark 애플리케이션에 할당되는 자원의 양을 신중하게 고려해야 합니다. 모든 클러스터 자원이 Analytic Server Spark 애플리케이션에 할당되면 다른 작업이 지연되거나 실행되지 않을 수 있습니다. 이러한 작업은 사용 가능한 자원이 충분해질 때까지 큐에서 대기할 수 있으며, 이러한 자원은 Analytic Server Spark 애플리케이션에서 사용합니다.

Analytic Server 서비스가 여러 개 구성되어 배포된 경우 각 서비스 인스턴스는 자체적인 지속적 Spark 애플리케이션을 잠재적으로 할당할 수 있습니다. 예를 들어 고가용성 장애 복구를 지원하기 위해 두 개의 Analytic Server 서비스가 배포된 경우, 두 개의 지속적 Spark 애플리케이션이 활성 상태이고 각 애플리케이션은 클러스터 자원을 할당합니다.

또 다른 복잡한 사항은 특정 상황에서 클러스터 자원이 필요한 맵리듀스 작업이 Analytic Server에서 시작될 수 있다는 점입니다. 이러한 맵리듀스 작업에는 Spark 애플리케이션에 할당되지 않은 자원이 필요합니다. 맵리듀스 작업이 필요한 특정 구성요소는 PSM 모델 빌드입니다.

다음 특성은 Spark 애플리케이션에 자원을 할당하도록 구성할 수 있습니다. 이러한 특성이 Spark 설치의 spark-defaults.conf에 설정된 경우, 환경에서 실행되는 모든 Spark 작업에 할당됩니다. 이러한 특성이 Analytic Server 구성의 "사용자 정의 analytics.cfg" 섹션 아래에 사용자 정의 특성으로 설정된 경우, Analytic Server Spark 애플리케이션에만 할당됩니다.

spark.executor.memory

실행기 프로세스당 사용할 메모리 양입니다.

spark.executor.instances

시작할 실행기 프로세스 수입니다.

spark.executor.cores

실행기 프로세스당 실행기 작업자 스레드 수입니다. 이 값의 범위는 1 - 5여야 합니다.

세 가지 주요 Spark 특성을 설정하는 예입니다. HDFS 클러스터에 10개의 데이터 노드가 있고, 각 데이터 노드는 24개의 논리 코어와 48GB의 메모리를 제공하며 유일하게 실행 중인 HDFS 프로세스입니

다. 다음은 이 환경에서 이러한 특성을 구성하는 한 가지 방법입니다. 이때, 이 환경에서는 Analytic Server 작업만 실행 중이고 최대 할당을 단일 Analytic Server Spark 애플리케이션으로 설정하고자 한다고 가정합니다.

- spark.executor.instances=20으로 설정하십시오. 이 경우 데이터 노드당 2개의 Spark 실행기 프로세스를 실행하려고 합니다.
- spark.executor.memory=22G로 설정하십시오. 이 경우 각 Spark 실행기 프로세스의 최대 힙 크기를 22GB로 설정하며, 각 데이터 노드에는 44GB가 할당됩니다. 다른 JVM 및 OS에는 추가 메모리가 필요합니다.
- spark.executor.cores=5를 설정하십시오. 이 경우 데이터 노드당 총 10개의 작업자 스레드에 대해 Spark 실행기당 5개의 작업자 스레드를 제공합니다.

Spark UI에서 실행 중인 작업 모니터

성능에 영향을 미칠 수 있는 디스크로 유출이 발생할 경우 몇 가지 가능한 솔루션은 다음과 같습니다.

- 메모리를 늘리고 **spark.executor.memory**를 통해 Spark 실행기에 메모리를 할당하십시오.
- **spark.executor.cores**의 수를 줄이십시오. 그러면 메모리를 할당하는 동시 작업 스레드의 수가 줄어들지만, 작업에 대한 병렬 처리 양도 줄어듭니다.
- Spark 메모리 특성을 변경하십시오. **spark.shuffle.memoryFraction** 및 **spark.storage.memoryFraction**은 Spark용 Spark 실행기 힙의 할당 백분율입니다.

이름 노드의 메모리가 충분한지 확인

HDFS의 블록 수가 많으며 계속 증가하고 있는 경우 이러한 증가세를 수용할 수 있도록 이름 노드 힙이 증가하는지 확인하십시오. 이는 일반적인 HDFS 튜닝 권장사항입니다.

캐싱에 사용되는 메모리 양 변경

spark.storage.memoryFraction의 기본값은 0.6입니다. 데이터의 HDFS 블록 크기가 64MB인 경우 이 값을 0.8로 늘릴 수 있습니다. 입력 데이터의 HDFS 블록 크기가 64MB 이상이라면 작업당 할당되는 메모리가 2GB 이상일 경우에만 이 값을 늘릴 수 있습니다.

모델 스코어링의 성능 튜닝

다음 단계를 수행하여 Apache Spark 엔진으로 빅 데이터 세트에 대한 모델 스코어링 작업의 성능을 향상시킬 수 있습니다. 이 단계는 클러스터에 있는 비Analytic Server 서비스의 작업에 영향을 주지 않아야 합니다.

1. 클러스터의 각 노드에 libtcmalloc_minimal.so{/version}가 이미 설치되어 있는지 확인하십시오.

   ```
   whereis libtcmalloc_minimal.so.*
   ```

2. libtcmalloc_minimal.so가 설치되어 있지 않은 경우 libtcmalloc_minimal 라이브러리가 포함되어 있는 운영 체제별 패키지를 클러스터의 각 노드에 설치하거나, libtcmalloc_minimal을 수동으로 작성하고 설치하십시오. 예를 들어, 다음과 같습니다.

Ubuntu:

```
sudo apt-get install libgoogle-perftools-dev
```

Red Hat Enterprise Linux 6.x(x64):

a. RedHat용 EPEL 리포지토리를 설치하십시오(아직 설치되지 않은 경우).

```
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
sudo rpm -Uvh epel-release-6*.rpm
```

b. `sudo yum install gperftools-libs.x86_64`

수동 작성:

a. https://github.com/gperftools/gperftools/releases 링크에서 gperftools-2.4.tar.gz를 다운로드하십시오.

b. `tar zxvf gperftools-2.4.tar.gz`

c. `cd gperftools-2.4`

d. `./configure --disable-cpu-profiler --disable-heap-profiler --disable-heap-checker --disable-debugalloc --enable-minimal`

e. `make`

f. `sudo make install`

3. 설치된 라이브러리 파일 libtcmalloc_minimal.so{.version}의 위치 중 하나를 메모해 두십시오. 이러한 위치는 하나 이상의 노드에서 다음 명령을 실행하면 리턴됩니다.

```
whereis libtcmalloc_minimal.so.*
```

클러스터에 여러 운영 체제를 실행하는 노드가 있을 경우 이 파일의 위치가 여러 개일 수 있습니다.

4. Ambari 콘솔에서 Analytic Server 구성으로 이동한 다음 사용자 정의 analytics.cfg 섹션에서 라이브러리 위치를 값으로 사용하여 spark.executorEnv.LD_PRELOAD 키를 구성하십시오. 이와 같이 변경한 후에는 Analytic Server 서비스를 다시 시작하십시오. 예를 들어 라이브러리가 /usr/lib64/libtcmalloc_minimal.so.4에 설치된 경우 구성은 다음과 같습니다.

```
spark.executorEnv.LD_PRELOAD=/usr/lib64/libtcmalloc_minimal.so.4
```

여러 개의 위치가 필요한 경우 다음 예와 같이, 위치를 공백으로 구분하십시오.

```
spark.executorEnv.LD_PRELOAD=/usr/lib64/libtcmalloc_minimal.so.4 /usr/lib/libtcmalloc_minimal.so
```

임의의 노드에서 `libtcmalloc_minimal.so` 라이브러리가 구성된 위치 중 하나에 설치되지 않은 경우 오류가 발생하지는 않지만, 이러한 노드에서 모델 스코어링의 성능이 저하될 수 있습니다.

Spark 맵 측 결합

Analytic Server Spark 결합 구현은 맵 측 결합 기능을 지원하지 않습니다. Spark 결합은 주로 리듀스 측입니다. 한 입력이 소형인 경우, 구현이 맵 측 결합을 활용하여 결합을 최적화하지 않습니다. 맵 측 결합을 활용하지 않으면 결국 실패하게 되는 극도로 자원 집약적인 Spark 작업이 됩니다.

Analytic Server Spark 맵 측 결합 또는 가장 작은 RDD를 기반으로 하는 원시 Spark 작업을 실행할 때 결합을 최적화하려면 spark.msj.maxBroadcast 특성을 analytics.cfg 파일(SPSS Analytic Server/Configs/Custom analytics.cfg) 또는 analytics-meta에 추가할 수 있습니다.

주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 이 자료는 IBM에서 다른 언어로 제공할 수도 있습니다. 그러나 자료에 접근하기 위해서는 해당 언어로 된 제품 또는 제품 버전의 사본이 필요할 수 있습니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이센스까지 부여하는 것은 아닙니다. 라이센스에 대한 의문사항은 다음으로 문의하십시오.

07326
서울특별시 영등포구
국제금융로 *10, 3IFC*
한국 아이.비.엠 주식회사
대표전화서비스*: 02-3781-7114*

2바이트(DBCS) 정보에 관한 라이센스 문의는 한국 IBM에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이센스 사용자는 다음 주소로 문의하십시오.

07326
서울특별시 영등포구
국제금융로 *10, 3IFC*
한국 아이.비.엠 주식회사
대표전화서비스*: 02-3781-7114*

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이센스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이센스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이센스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

인용된 성능 데이터와 고객 예제는 예시 용도로만 제공됩니다. 실제 성능 결과는 특정 구성과 운영 조건에 따라 다를 수 있습니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

여기에 나오는 모든 IBM의 가격은 IBM이 제시하는 현 소매가이며 통지 없이 변경될 수 있습니다. 실제 판매가는 다를 수 있습니다.

이 정보는 계획 수립 목적으로만 사용됩니다. 이 정보는 기술된 제품이 GA(General Availability)되기 전에 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 인물 또는 기업의 이름과 유사하더라도 이는 전적으로 우연입니다.

저작권 라이센스:

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 인물 또는 기업의 이름과 유사하더라도 이는 전적으로 우연입니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 그 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

상표

IBM, IBM 로고 및 ibm.com은 전세계 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보"(www.ibm.com/legal/copytrade.shtml)에 있습니다.

Adobe, Adobe 로고, PostScript 및 PostScript 로고는 미국 및/또는 기타 국가에서 사용되는 Adobe Systems Incorporated의 등록상표 또는 상표입니다.

IT Infrastructure Library는 현재 Office of Government Commerce의 일부인 Central Computer and Telecommunications Agency의 등록상표입니다.

Intel, Intel 로고, Intel Inside, Intel Inside 로고, Intel Centrino, Intel Centrino 로고, Celeron, Intel Xeon, Intel SpeedStep, Itanium 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation 또는 그 계열사의 상표 또는 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

ITIL은 미국 특허청(U.S. Patent and Trademark Office)에 등록된 The Minister for the Cabinet Office의 등록상표 및 등록 공동체 상표입니다.

UNIX는 미국 및 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Cell Broadband Engine은 미국 또는 기타 국가에서 해당 라이센스에 의거하여 사용되는 Sony Computer Entertainment, Inc.의 상표입니다.

Linear Tape-Open, LTO, LTO 로고, Ultrium 및 Ultrium 로고는 미국 및 기타 국가에서 사용되는 HP, IBM Corp. 및 Quantum의 상표입니다.