z/TPF EE V1.1
z/TPFDF V1.1
TPF Toolkit for WebSphere® Studio V3
TPF Operations Server V1.2

IBM Software Group

TPF Users Group Fall 2005

z/TPF Test, Drivers, Automation and Performance

QUALITY Is Our Goal

Name: Kevin Jones
Venue: System Control Program Subcommittee

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0
© IBM Corporation 2005

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Agenda

- Introductions
- IBM Test Environment
- Test Tools - Drivers
- Test Tools – Automation
- Other Test Tools
- Project Level Testing
- Release Level Testing
- z/TPF Quality Measurements
- Conclusion
- Question and Answer

© IBM Corporation 2005

IBM Test Environment

- z/990 2084 Model D32 Processor
  - 256GB central storage
  - Divided into LPARs:
    - z/VM production system
      - SUSE SLES 8 Linux Server and SLES 9 Linux Server guests
    - z/VM production system
      - TPF and z/TPF guests
      - generally using VPARS and VTAPE
    - z/OS production system
    - Six coupling facilities (CFs)
    - 22 “native” LPARs for TPF and z/TPF test systems

© IBM Corporation 2005

IBM Test Environment *(continued)*

- Production/Development LPARs can be shutdown off-shift to test:
  - Large storage configurations
    - for example, a single 250GB z/TPF system
  - Destructive hardware testing
    - for example, injecting machine checks

© IBM Corporation 2005

IBM Test Environment *(continued)*

- DASD
  - ESS 2105-F20
  - ESS 2105-800
  - DS8000
  - DS6000
- Tapes
  - Stand-alone 3490E drives
  - Stand-alone 3592 tape drives
  - 3590E drives in a 3494 library
  - B18 VTS with 128 logical drives
  - B10 VTS with 64 logical drives (microcode testing)

© IBM Corporation 2005

IBM Test Environment *(continued)*

- SNA
  - ESCON CTC, 3745 and 3746 Nways Controllers
  - network is comprised of APPN and Subarea nodes
  - traffic driven using TPNS and drivers, mainly between z/OS, z/TPF and remote terminals

- TCP/IP
  - OSA Express and 3746 Nways Controller
  - network is comprised of token ring, ethernet and WAN
  - traffic driven using drivers on z/OS, z/VM, Linux, Microsoft Windows and z/TPF

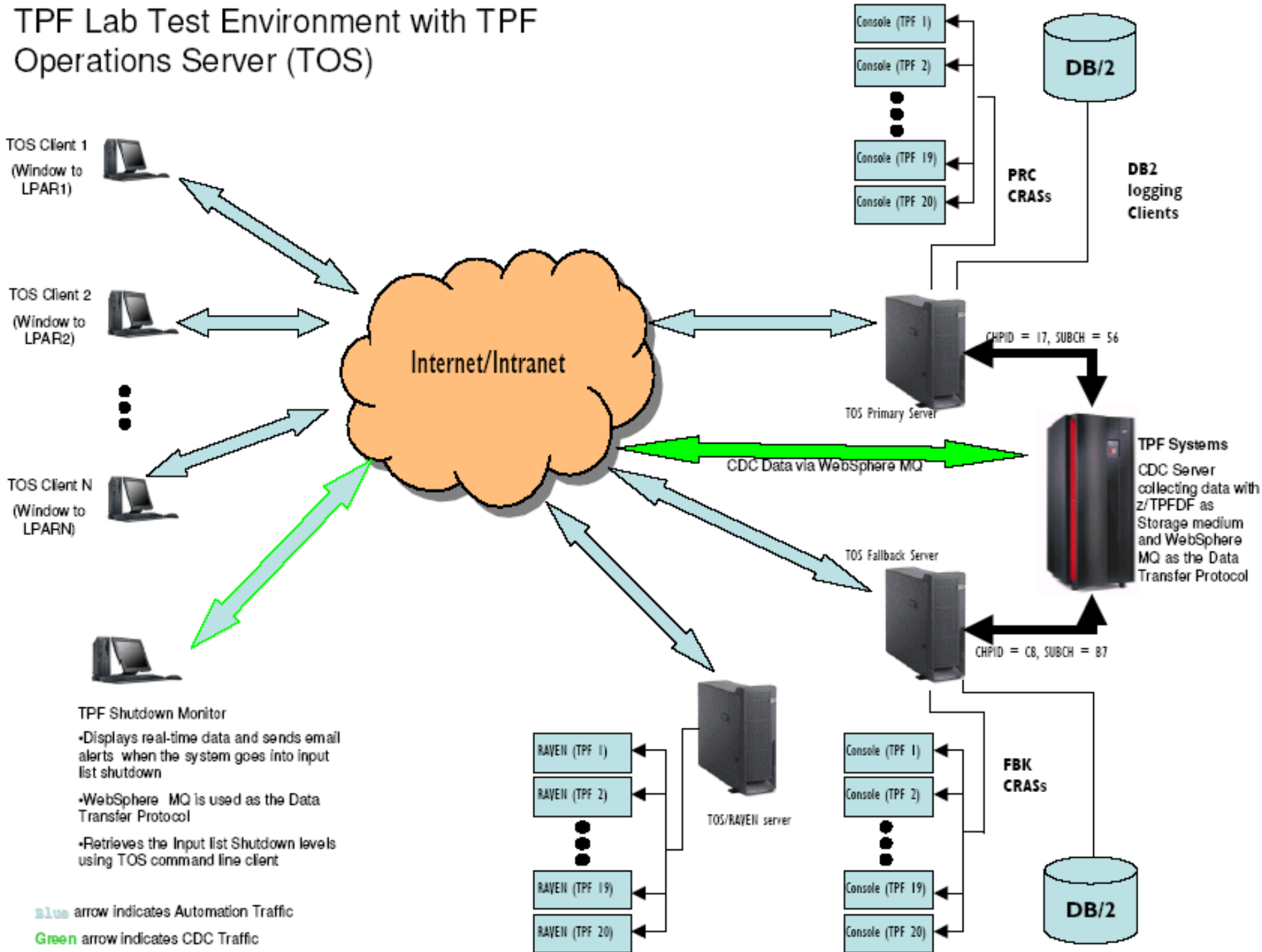
© IBM Corporation 2005

IBM Test Environment *(continued)*

- TPF Operations Server (TOS)
  - Primary and fallback servers connect to TPF and z/TPF providing:
    - Prime CRAS consoles
    - Message log databases (DB2)
    - Flat file logging
    - Security user exit
    - Data encryption between client and server
  - RAVEN automation runs on a TOS/RAVEN server

© IBM Corporation 2005

TPF Lab Test Environment with TPF Operations Server (TOS)



© Copyright IBM Corporation 2005. All rights reserved.

IBM Test Environment *(continued)*

- Five "native" z/TPF test systems with HPO
  - BSS is fully duplicated with 21 prime modules
  - Selectively-duplicated subsystem with 7 prime modules and 3 subsystem-users
  - CTC definitions for 13 loosely-coupled processors
  - One system dedicated to 24x7 testing
- Two "native" z/TPF test systems without HPO
  - Fully duplicated with 4 prime modules
  - One system dedicated to 24x7 testing
- TPF "native" test systems are also available (3 with HPO and TPFDF, 2 without HPO and TPFDF)

© IBM Corporation 2005

IBM Test Environment *(continued)*

- Under z/VM and VPARS, different z/TPF system levels are available:
  - *Current (CUR)* systems are available which include every completed APAR
    - Ensures the latest code is available for testing
    - System is subject to frequent changes
  - *Commit* systems are available which are rebuilt monthly
    - More stable, but latest APARs are not loaded

© IBM Corporation 2005

Test Tools - Drivers

- A *driver* is a collection of z/TPF realtime programs written to test specific z/TPF functionality
- A dedicated team develops and maintains our driver suite
  - Ensures consistency and develops a skills base
- Driver development is fully integrated into our overall process
  - Requirements are defined soon after new, changed and obsolete externals (including APIs) are identified
  - Design, development, code reviews, test and documentation are fully staffed
    - Not an afterthought developed in “spare time”
    - Change control is used for all source code

© IBM Corporation 2005

Test Tools – Drivers *(continued)*

- Attributes of a good driver include one or more of the following:
  - Quiet and verbose options
    - "quiet" – only issue messages if an error occurs
    - "verbose" – issue messages for normal and error conditions
  - Repeatable results for problem determination
    - For example, avoid random number generators

© IBM Corporation 2005

Test Tools – Drivers *(continued)*

- Executes one, many or all test cases
  - One test case is appropriate for Function Test
  - Many or all test cases are appropriate for System Test
- Runs continuously until the driver is explicitly stopped
  - Required for 24x7 test environments
- Executes normal and error conditions
  - Exercise as many z/TPF code paths as possible
- Self-checking
  - Don’t assume return code = 0 is valid!!
  - For example, ensure IPC messages are sent by returning an acknowledgement

© IBM Corporation 2005

Test Tools – Drivers *(continued)*

- Tests all APIs (macros and functions) in a given area
  - Including permutations of each parameter or option
- Simulates customer databases
  - Especially necessary to test Recoup, DBR, Capture and Restore, etc.
- Runs well with other drivers to simulate customer workloads
  - Don’t assume that all system resources are available!
    - For example, use LODIC to control resource usage
- Throttles available to stress hardware or software
  - For example, DASD channels or z/TPF memory management routines

© IBM Corporation 2005

Test Tools – Drivers *(continued)*

- Certain drivers (and more in the future) are available on the z/TPF website
  - http://www.ibm.com/tpf/download/tools.htm
  - All drivers are provided “as is” and without warranty
- Sample drivers are also available on the USB memory sticks distributed at the IBM Hospitality Suite last night
  - \TPFUG October 2005\zTPF Drivers and Automation\Drivers

© IBM Corporation 2005

QUALITY is our goal
z/TPF Test Drivers

- **Drivers to test C/C++ support**
  - EDRIVER, LIBSTDC++ - Standard C++ library
  - *FLOAT* - *Floating Point Conversion*
  - GLIBC - GNU C library
  - *POSIX* - *POSIX Support*
  - *SGNL* - *Signal*
  - *TCT0* - *C++ Exception handling*

- **Drivers to test Globals**
  - *CGLB* - *Format-1 Globals C APIs*
  - *GLF2* - *Format 2 Globals*
  - *GLOB* - *Format-1 Globals Assembler*
  - *TGBS* - *Format-1 Globals Synchonization*

- **Drivers to test z/TPF system services and APIs**
  - **API5** - **z/TPF APIs**
  - *CHER* - *CPU Scheduler*
  - *CORO* - *TPF Transaction Services*
  - *CPXT* - *CP User Exits*
  - *DECB* - *DECB Support*
  - *KPTS* - *48K Keypoint APIs*
  - *MEM5* - *Memory Management*
  - *TAPE, REEL* - *Tapes I/O APIs*
  - *TMSL* - *Time Slice Processing*
  - *DLCK* - *Deadlock Detection*
  - *DUMP* - *Dump Support*
  - *ECBM* - *ECB Resource Monitor*
  - *GDSN* - *General Data Sets*
  - *IEFC* - *Internal Event Facility*
  - **LISH** - **LODIC and Input List Shutdown**
  - *PER0* - *Program Event Recording*
  - *SOLD* - *Shared Objects Linkage*
  - *TLOG* - *Trace Log Support*
  - *TRAG* - *Trace Group Support*
  - *UNIT* - *Unit Record Support*

- **purple bold** = available on z/TPF website http://www.ibm.com/tpf/download/tools.htm
- *green italic* = *tentatively planned for distribution on z/TPF website*
- red = ported drivers

© IBM Corporation 2005

IBM is a trademark of the International Business Machines Corporation in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.

QUALITY is our goal
z/TPF Test Drivers

- **Drivers to test communications**
  - *ECHO, L62A - LU 6.2*
  - *MPIF - Multi-Processor Interconnect Facility*
  - *OSA1 - OSA Express Setup*
  - *ROUTC - Message Routing*
  - **SIPC - System InterProcessor Communication**
  - *SOCK - TCP/IP Socket APIs*
- **Middleware Drivers**
  - *MQS1 - MQSeries Support*
  - *ZMARK - File System*
  - *SOAP - SOAP APIs*
  - *TO2D - z/TPF Collection Support*
- **Database drivers**
  - *DEB5 - Create, maintain, and modify subsystem unique TPF databases*
  - *DFTD, JOB1,JCB1 - TPFDF databases*
  - **POOL - Pool dispense**
  - *RECP - Recoup*
- **Other drivers**
  - *AIR1 - Workload driver for performance measurements*
  - *DB2C - TPFAR*
  - *DBUG - z/TPF Debugger*
  - *HOTR, HOTO - "HOT" DASD Records*
  - ROAST - Platform for file system, threads and signals testing
  - *NUKE - Application Stack Corruption*

- `purple bold = available on z/TPF website http://www.ibm.com/tpf/download/tools.htm`
- *`green italic = tentatively planned for distribution on z/TPF website`*
- `red = ported drivers`

© IBM Corporation 2005

IBM is a trademark of the International Business Machines Corporation in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.

Test Tools – Drivers *(continued)*

- Key drivers developed to support z/TPF testing include;
  - SOLD *(Shared Object Linkage Driver)* – tests all combinations of program linkage between:
    - C Shared Objects (CSOs)
    - BAL shared objects (BSOs)
    - ELF executables (C program with main())
  - LISH *(LODIC and Input List Shutdown)* – ensures that z/TPF LODIC and shutdown levels are working correctly when system resources are depleted
    - Also ensures system recovery when resources are returned

© IBM Corporation 2005

Test Tools – Drivers *(continued)*

- NUKE *(application stack corruption)* – ensures the overall z/TPF system is not affected when an ECB corrupts it's application stack
- MEM5 *(memory management)* – verifies all APIs associated with memory management:
  - ECB heap
  - System heap
  - Application stack
- CHER *(CPU scheduler)* – ensures that programs are dispatched on the correct I-stream based on ECB and program I-stream affinity settings

© IBM Corporation 2005

Test Tools – Drivers *(continued)*

- GLF2 *(format-2 globals)* – verifies all APIs associated with format-2 globals support
- ECBM *(ECB resource monitor)* – verifies the ECBs exceeding resource limits are properly identified by z/TPF
- KPTS *(48K keypoints)* – verifies new keypoint APIs
- TLOG *(trace log)* – ensures ECB trace data is logged properly

© IBM Corporation 2005

Test Tools – Drivers *(continued)*

– Drivers were also ported to support z/TPF testing:

- ROAST – test case platform currently used for file systems, threads and signals
- GLIBC – test cases for C library routines
- LIBSTDC++ - test cases for C++ library routines

Porting drivers reduce development expenses and provide better code coverage. IBM will continue to port drivers as functions are ported to z/TPF.

© IBM Corporation 2005

Test Tools - Automation

- Automation complements drivers by performing repetitive tasks automatically
- The same team responsible for drivers also develops and maintains our automation suite
  - Ensures consistency and develops a skills base

© IBM Corporation 2005

Test Tools – Automation *(continued)*

- Automation development is fully integrated into our overall process
  - Requirements are defined soon after new, changed and obsolete commands are identified (including driver commands)
  - Design, development, test and documentation are fully staffed
    - not an afterthought developed in “spare time”
  - Change control is used for all source code

© IBM Corporation 2005

Test Tools – Automation *(continued)*

- Automation is implemented using the TPF Operations Server (TOS) running on a TOS/RAVEN server
- Three types of automation scripts
  - *VAR* files in REXX which allow:
    - complex logic and decision making
    - flexibility to respond to error conditions
  - *TDR* files which include simple scripting options
    - less flexibility, but easier to code
  - *FIL* files which are simple lists of z/TPF commands
    - no flexibility beyond entering a list of commands
    - easiest to code

© IBM Corporation 2005

Test Tools – Automation *(continued)*

- Many automation scripts will be made available on the z/TPF website
  - http://www.ibm.com/tpf/download/tools.htm
  - All scripts will be provided “as is” and without warranty
- Sample scripts are also available on the USB memory sticks distributed at the IBM Hospitality Suite last night
  - \TPFUG October 2005\zTPF Drivers and Automation\Automation

© IBM Corporation 2005

QUALITY is our Goal
z/TPF Test automation -- TOS/Raven-based platform

- **Automation that needs people**
  - *API5.fil* - *Issue API5 driver test cases*
  - *BIGBROTHER.var* - *Initiate TPF performance monitoring (ZMEAS or ZTRAP)*
  - *CAPTURE.var* - *Setup and start the z/TPF CAPTURE utility*
  - *DSTAT.var* - *Report active/inactive state of background drivers*
  - HOTODRV.var - Start/stop HOTO driver on 256 ordinals
  - MONSTER_LOAD.tdr - Runs 2000 ZOLDR commands
  - *NORMAN.var* - *Report clock value on last cycle to NORM*
  - POSITIVEFB.var - Run Positive Feedback system test cases
  - SIOUX.var - Run CPXT driver to enable/disable CP user exits
  - *SYSCHECK.var* - *Present system test environment settings*
  - TDRVSTAT - Loop of status commands
  - *ZxPGM.tdr* - *Run ZDPGM/ZAPGM system test cases*

- **Automation that makes z/TPF easier to use and enforces system test conventions**
  - *BCKGND* - *Manage background drivers*
  - *CHECKER* - *Set DASD lock space to maximum*
  - *FIX* - *Simple repair/replace/reset various system attributes to support system test*
  - *GETADRIVE* - *ZTVAR ADD a requested number of tape drives*
  - *INITTAPE* - *ZTINT and ZTPFL MOVE to a category a specified list of tapes*
  - *POSTMASTER* - *Create/delete z/TPF test mail userids, passwords, inboxes, ACLs*

green italic = planned for distribution on z/TPF website
http://www.ibm.com/tpf/download/tools.htm

© IBM Corporation 2005
IBM is a trademark of the International Business Machines Corporation in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.

QUALITY is our goal
z/TPF Test Automation -- TOS/Raven-based platform

- **LIGHTS OUT maintenance**

  (unmanned and automatically triggered)

  - *PDUMAINT* - *Runs PDU commands, sends e-mail alert when problems are detected*
  - *TAPEMAINT* - *Daily scratch tape reclamation, sends e-mail report on state of tapes at end of processing*

- **System Test Monitors**

  - *OUTAGEMON* - *Monitor and send e-mail alert for out of NORM state and resources depleted (input list shutdown) conditions*
  - *SNAMON* - *Setup and monitor SNA communications between 2 complexes*

green italic = planned for distribution on z/TPF website
http://www.ibm.com/tpf/download/tools.htm

© IBM Corporation 2005
IBM is a trademark of the International Business Machines Corporation in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.

Test Tools – Automation *(continued)*

- Automation can be used for a wide variety of purposes, not just to reduce operator errors
  - Exercise z/TPF commands
    - *ZxPGM* – run test cases for ZDPGM and ZAPGM
  - Run z/TPF test drivers
    - *API5* – run the API5 test driver
  - Monitor z/TPF test drivers
    - *DSTAT (driver status)* – reports status of all background drivers
  - Run z/TPF utilities as needed
    - *Capture* - setup and run the z/TPF capture utility

© IBM Corporation 2005

Test Tools – Automation *(continued)*

- Run scheduled z/TPF utilities and raise alerts as needed
  - *PDUMAINT* – runs PDU to completion every 6 hours
    - Alerts are sent via e-mail when errors occur such as double or multiple releases
    - PDU may or may not continue depending on the nature of the error
- Detect outages and raise alerts as needed
  - *OUTAGEMON (outage monitor)* – ensures that the z/TPF system is available for processing transactions
    - NORM state, and not in persistent input list shutdown
    - Alerts sent via e-mail when an outage is detected

© IBM Corporation 2005

Test Tools – Automation *(continued)*

- Monitor communications and raise alerts as needed
  - *SNAMON (SNA monitor)* –monitor SNA availability and raise and send via e-mail when an error is detected
- Check for common system problems
  - *SYSCHECK* - runs a series of system status queries to detect possible problems with a z/TPF test system
    - For example, check for active loadsets that may need to be deactivated and deleted, or accepted

© IBM Corporation 2005

Test Tools – Automation *(continued)*

- Drive 24x7 System Test activity
  - *BCKGND* – start a series of test drivers running quietly in the background on a variety of loosely-coupled processors, I-streams, subsystems and subsystem-users
- Manage and maintain system resources
  - *TAPEMAINT* – nightly maintenance and reclamation of scratch tapes
    - E-mail report sent after each run

© IBM Corporation 2005

Other Test Tools

- A Lotus Notes database “Test Tracking Tool” (TTT) is used to track and report test case status
  - Report various counts to date:
    - Planned completed
    - Actual completed
    - Actual attempted
  - Currently assessing Rational tools such as *Test Manager*

© IBM Corporation 2005

Other Test Tools

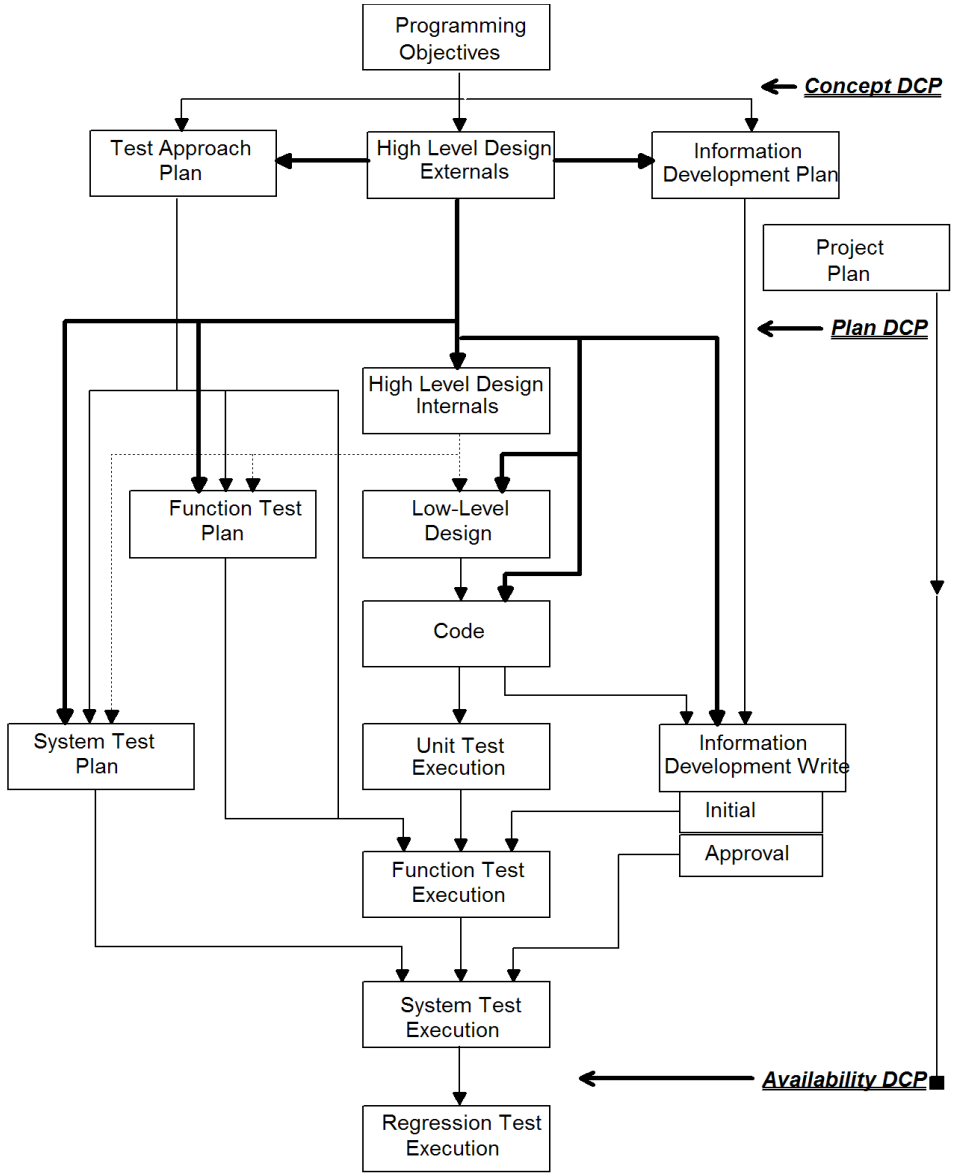
- Problem determination for system and driver errors is done in many ways:
  - IBM TPF Toolkit for WebSphere Studio
    - TPF Debugger
  - CP trace under z/VM
  - PER trace using the ZSPER command
  - Trace and display facilities on the processor's Hardware Management Console (HMC)
  - Dump analysis (using manual restarts if needed)

© IBM Corporation 2005

Project Level Testing

- The z/TPF lab follows an IBM software development process for all projects

- On a typical project, Design and Test account for *85%* of all resource requirements
  - Design – 45%
  - Test – 40%



© IBM Corporation 2005

Project Level Testing *(continued)*

- How can the z/TPF lab afford to spend 85% of all project resources on Design and Test?
  - We can't afford to spend less than 85%!
- The expense required to correct a defect increases dramatically as a project progresses
  - Strong up-front designs ensure code is not implemented, tested or shipped that does not meet specifications
  - Strong test efforts ensure as many defects as possible are corrected before code is sent to customers
    - APARs have a significant impact to customers and IBM

© IBM Corporation 2005

Project Level Testing *(continued)*

- Test, Driver and Automation requirements are considered early in every project
  - Early involvement ensures:
    - Resource requirements are identified and included in project plans and staffing requests
    - Adequate calendar time is available to:
      - Design, develop, and document drivers
      - Design, develop and test automation scripts
      - Write and review test plans
      - Obtain any necessary hardware or software co-requisites

© IBM Corporation 2005

Project Level Testing *(continued)*

- The "High Level Design Externals" (HLDE) is the key document needed to drive test activities:
  - Defines the new behavior of the product
  - Defines new, changed and obsolete externals including macros, functions, messages, commands and dumps
  - Provides migration and other end-user information
- The Test Team is required to review and approve all HLDEs (among other lab personnel)

© IBM Corporation 2005

Project Level Testing *(continued)*

- Once an HLDE is approved, a *Test Approach Plan (TAP)* is written
  - High-level scope of testing to be performed
    - Unit, Function, System, and Regression Tests
  - May include other tests such as Performance or Beta
  - Defines hardware or software requirements
  - Defines tools to be developed (test drivers and automation)
  - Information is reviewed, approved, then reflected in Project and Test plans

© IBM Corporation 2005

Project Level Testing *(continued)*

- Unit Test is typically run by the code developer
  - Tests 100% of the affected code including error paths
- Function Test
  - Goal is to test all new procedures and externals
    - APIs (macros and functions)
    - Commands
    - Output messages
    - System Errors
    - Documentation
  - Requires reviewed and approved test plans
  - Driver and automation requirements are implemented during Function Test planning

© IBM Corporation 2005

Project Level Testing *(continued)*

- System Test
  - Ensures new support works properly when introduced to system workloads and a *24x7* environment
  - Requires reviewed and approved test plans
  - Projects are eventually merged into a common test system with other projects on the same PUT
- Regression Test
  - For lengthy projects, it is necessary to run regularly scheduled regression tests
  - Goal is to ensure that test cases which were previously successful continue to run over time

© IBM Corporation 2005

Release Level Testing

- Specialized tests designed to find defects that may not occur in Function and System Tests
  - Regression Test
    - Verifies most areas of the system on a regular schedule
    - Three to five day focused effort, nine times per year
    - Emphasis is placed on areas affected by recent maintenance (APARs)

IBM has found significant benefit to the repetitive nature of Regression Test, further highlighting our need to continue developing automation

© IBM Corporation 2005

Release Level Testing *(continued)*

- Configurations are varied greatly during Regression Test to ensure the widest code coverage
  - z/VM and VPARS with up to 33 I-streams
  - Nine-way loosely-coupled systems
  - Single LPAR tests using 250GB of storage in different memory configurations (large VFA, large number of 1MB frames, etc.)
  - HPO and non-HPO systems
  - BSS, named-BSS and non-BSS subsystems
  - Multiple subsystem-users

© IBM Corporation 2005

Release Level Testing *(continued)*

- Migration, Coexistence and Fallback
  - Major emphasis for z/TPF
  - Verified that:
    - A vanilla TPF system could be migrated to z/TPF
    - Fallback was possible to TPF
    - TPF and z/TPF can co-exist in the same loosely-coupled complex

© IBM Corporation 2005

Release Level Testing *(continued)*

- Verified the z/TPF Migration Guide by following the documented steps to migrate
- Updated our TPF drivers with single-source APARs, then ensured the resulting source code worked on TPF and z/TPF
- Focus was on databases, utilities and any area where data is shared between loosely-coupled processors
  - verify that you can access a file created on TPF from z/TPF
  - verify that capture tapes from TPF can be restored to z/TPF

© IBM Corporation 2005

Release Level Testing *(continued)*

- *24x7* systems providing a true System Test environment
  - Detects problems that occur only over time
    - for example, a slow memory leak
  - Automation is used to:
    - Create background activity, primarily using drivers
    - Perform system maintenance such as tape and PDU
    - Monitor the system for outages
  - System continues to run even when no projects are in System Test
  - All outages are treated as high-severity and require analysis and corrective action

© IBM Corporation 2005

Release Level Testing *(continued)*

- Field ("beta") Tests – working with customers to install pre-GA versions of code
  - Provide customer with valuable pre-migration experience
  - Identifies defects earlier
  - Provides early feedback to IBM about design and usability
- Hardware Tests
  - Verify new hardware not requiring TPF or z/TPF code changes
  - Support hardware labs within IBM using TPF and z/TPF as a test tool

© IBM Corporation 2005

Release Level Testing *(continued)*

- *Enhanced Error* testing
  - Emphasis for z/TPF as a result of customer feedback
  - Focus on hardware errors and recovery
    - Machine checks (injected using engineering tools)
    - STR *(Sysplex Timer)*
    - DASD
    - Tape
  - Also implemented drivers such as NUKE which causes application stack corruption

© IBM Corporation 2005

Release Level Testing *(continued)*

- Performance Tests
  - Major emphasis for z/TPF
  - AIR1 driver accurately simulates customers workloads
  - Also focused on the performance of C/C++ by testing middleware such as Websphere® MQSeries, Mail and SOAP/XML
  - Significant system changes were implemented as a direct result of these tests

© IBM Corporation 2005

Release Level Testing *(continued)*

IBM estimates that z/TPF and z/TPFDF will, on average, result in a 10% increase (factor of 1.1) in CPU utilization as compared to TPF and TPFDF. This applies when comparing equivalent applications, message rates and use of operating system functions. Performance is influenced by many different factors, so every z/TPF and z/TPFDF system will be impacted differently. Your results may differ from this estimate depending on your system's profile.

© IBM Corporation 2005

z/TPF Quality Measurements

- New quality measurements were needed for z/TPF given the size and scope of the code changes
  - Every z/TPF source file was placed in a *test group*
    - For example, DASD support
    - There were approximately 60 *test groups*
  - Each *test group* was assigned a complexity on a scale of 1 to 5
  - Given a complexity and a count of new, changed and deleted *lines of code*, we projected an expected number of Function and System Test defects for each *test group*

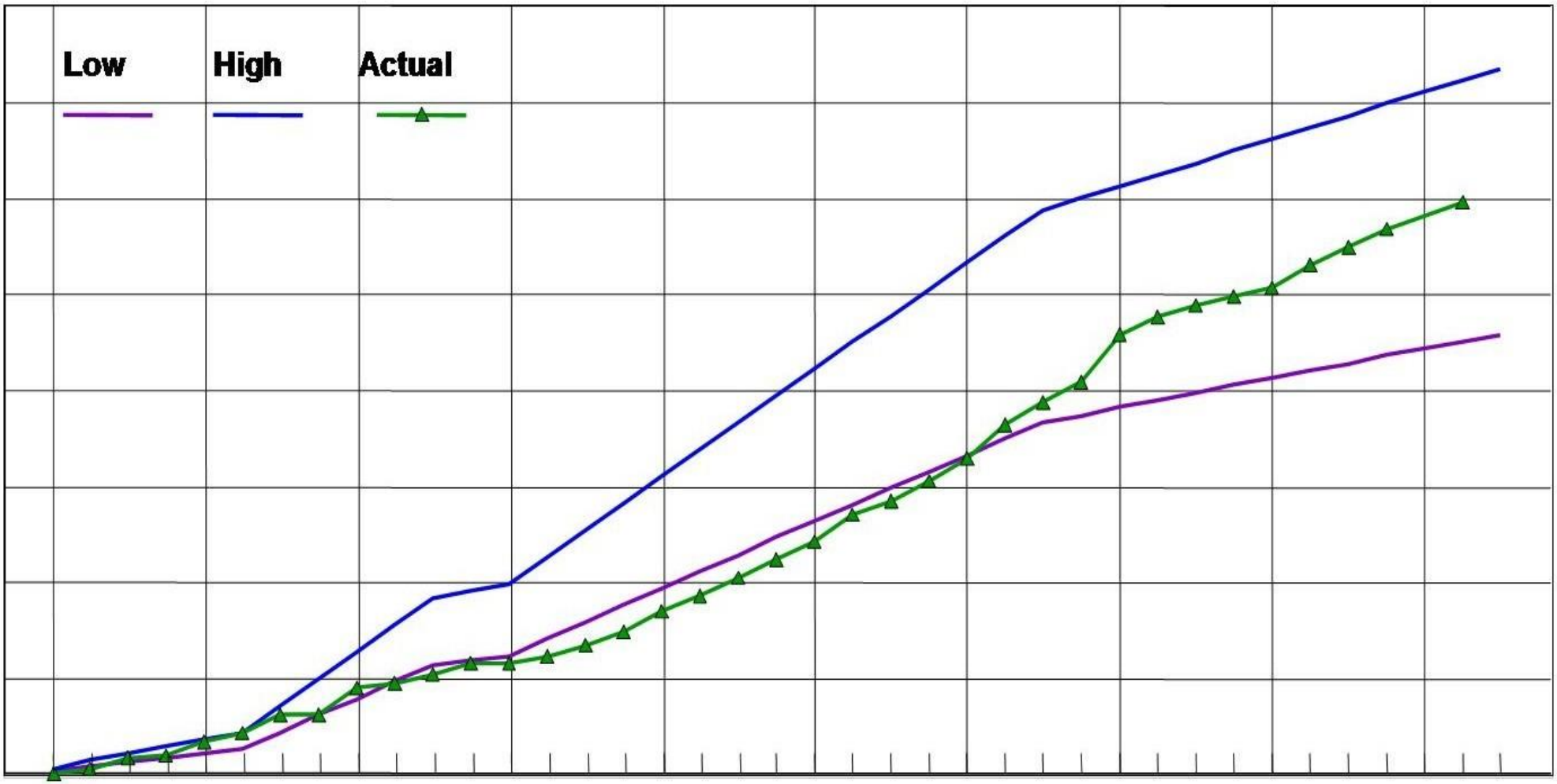
© IBM Corporation 2005

z/TPF Quality Measurements *(continued)*

- Any *test group* yielding too many or too few defects was analyzed, and corrective actions taken
- For System Test, we also:
  - Rolled up the data to ensure the quality of the overall product
  - Examined defect discovery rates to ensure new defects were no longer being found at the end of System Test

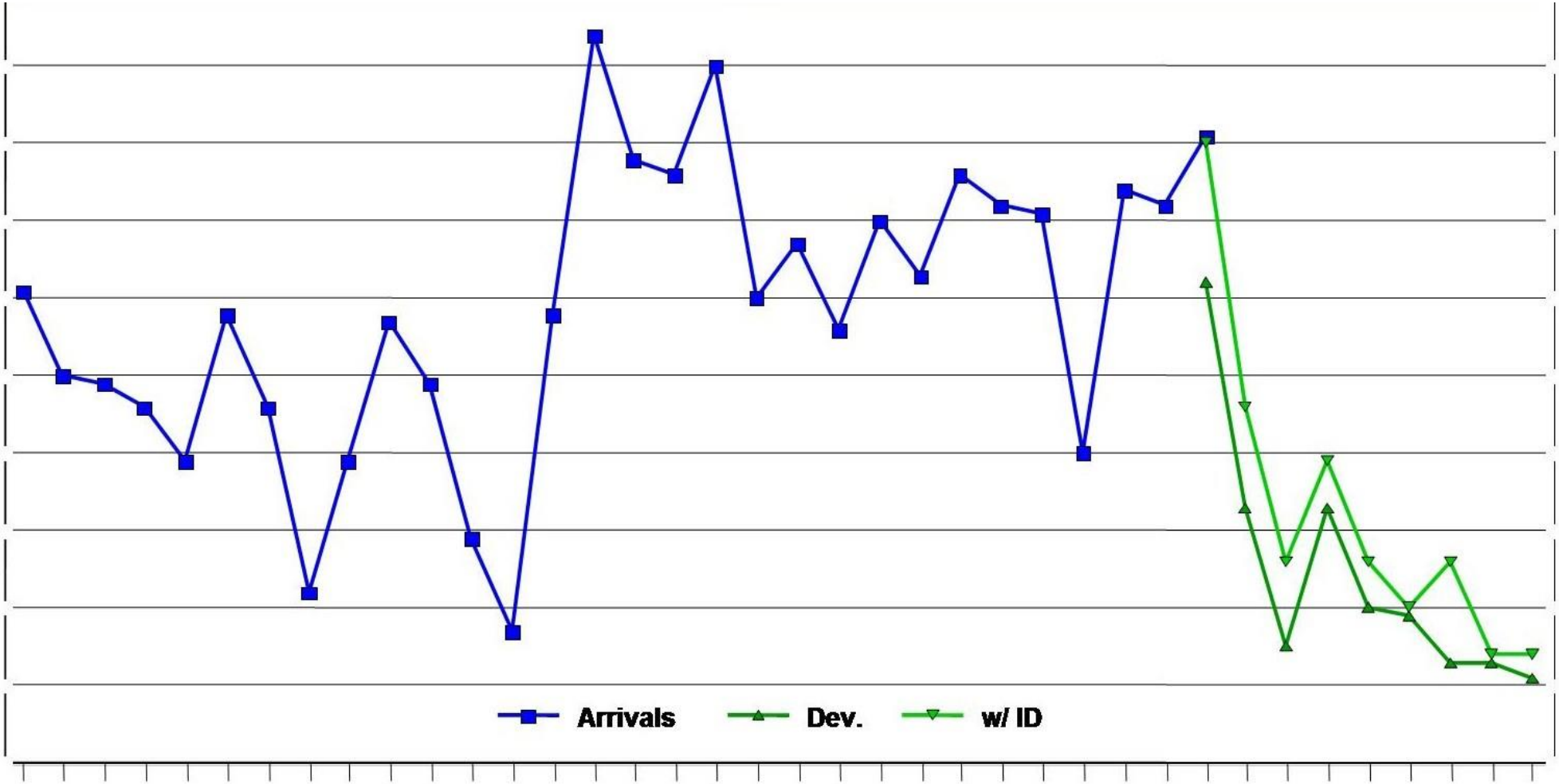
© IBM Corporation 2005

z/TPF System Test Defect Tracking



© IBM Corporation 2005

z/TPF System Test Defect Arrivals



© IBM Corporation 2005

Conclusion

- Successful software test disciplines must include:
  - Ensuring development of tools is properly staffed and integrated into the overall development process
  - Early and ongoing planning and tracking of all test activities is essential
  - Quality must be measured *and acted upon* to ensure the highest quality deliverables
  - Well-defined but flexible process with a focus on "lessons learned" to ensure continuous improvement
  - Implementing many different types of tests in a wide variety of configurations
  - Looking for opportunities to port test applications

© IBM Corporation 2005

Question and Answer

Trademarks

IBM, MQSeries, Websphere, Rational, Lotus, Notes, z/VM, z/OS, Nways, and DB2 are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Notes
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.

© IBM Corporation 2005