IBM Software Group

IBM WebSphere Application Server v6

WebSphere Performance Tools

e business on demand.

IBM Confidential

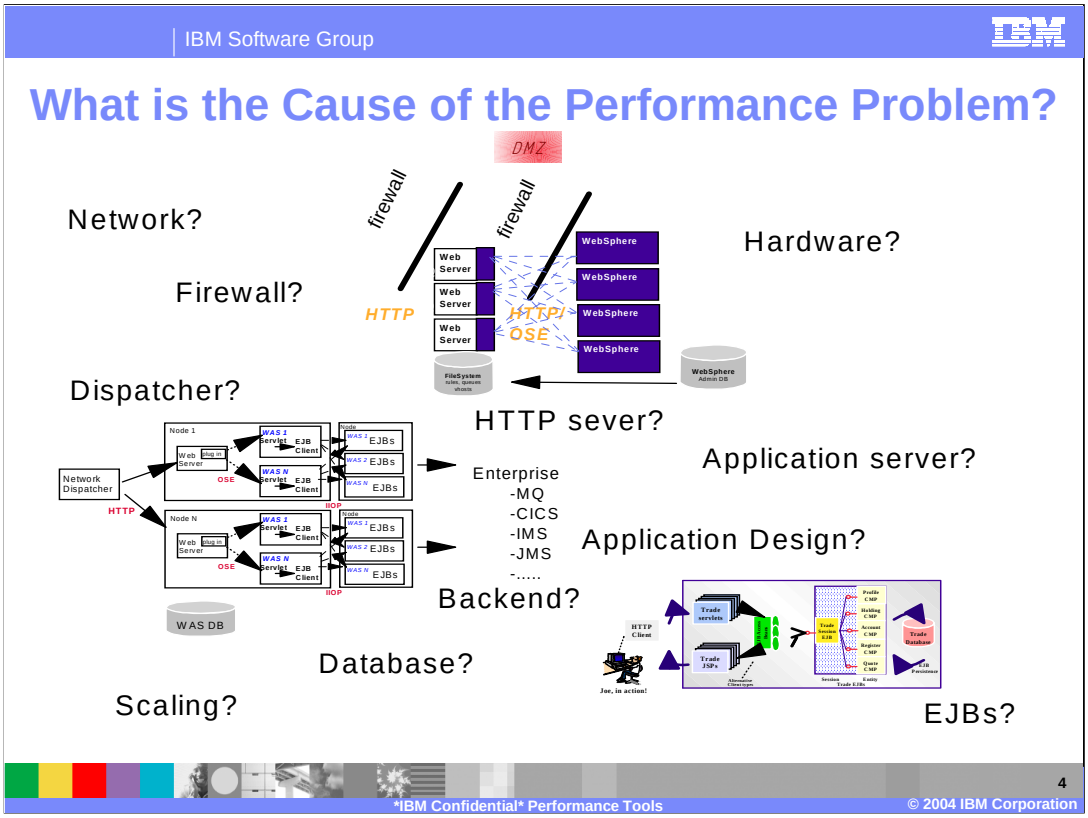
Updated October 12, 2004

Agenda

- WebSphere Environment
- Performance Monitoring Infrastructure (PMI)
- Tivoli Performance Viewer
- Performance Advisors
  - Tivoli Performance Viewer Performance Advisor
  - Runtime Performance Advisor

Section

WebSphere Environment



The problem can be (and has been) almost anywhere within an application. The problem can be network and hardware related, backend system related, it can be actual product bugs, or quite often, application design issues.

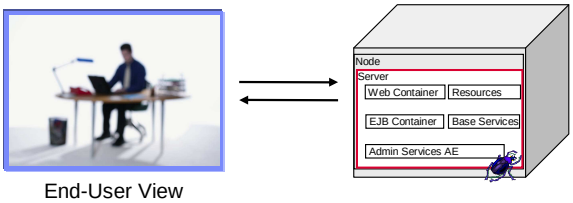
With today's applications becoming increasingly complex it is getting harder and harder to track down these problems. Typical E-business applications now have one or several of the following attributes:

- Multiple logical and/or physical tiers
- Mixture of operating system platforms
- Mixture of hardware architectures
- Fuzzy application boundaries

Today's software is capable of producing massive quantities of numbers; CPU usage statistics, memory usage statistics, queuing statistics, transaction rate statistics, buffer pool statistics etc. The problem is that the large amount of statistics can actually be overwhelming and may hamper, rather than help, a person's ability to understand and analyze performance data.

Today… We have

- Data
- Viewer
- Advisors

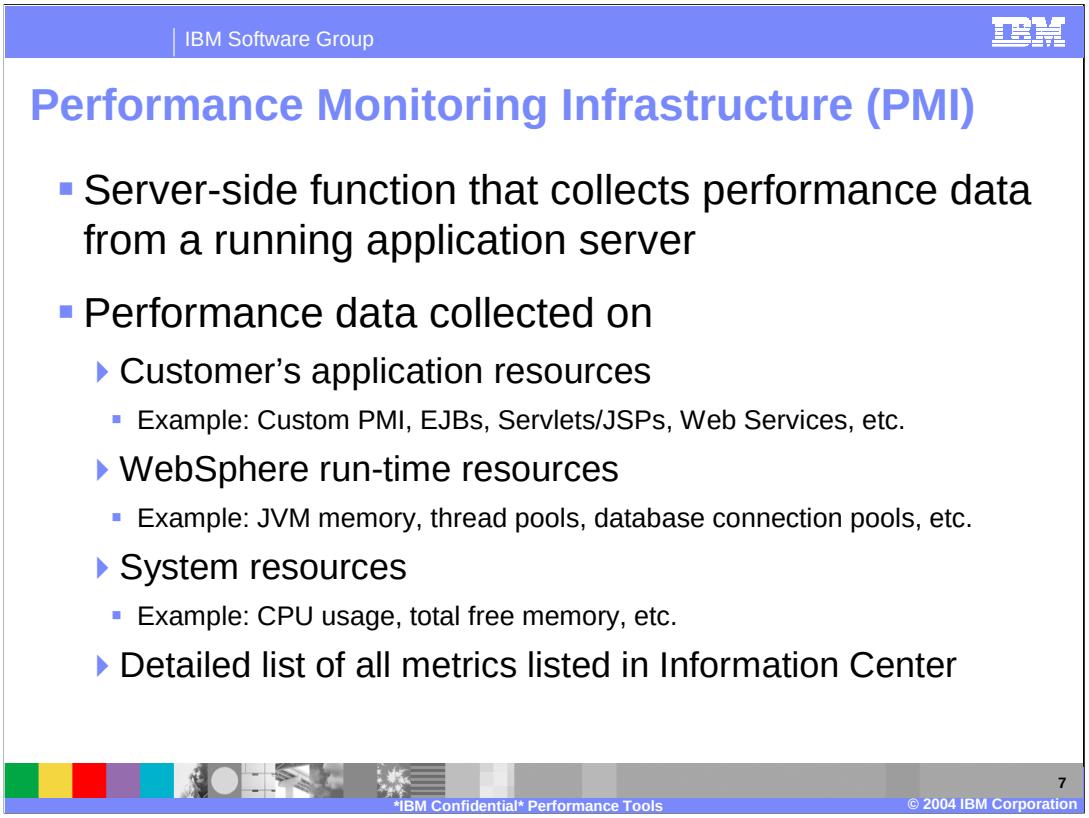


End-User View

WebSphere provides all of these things to the administrator. The Performance Monitoring Infrastructure (PMI) externalizes the performance data. The Tivoli Performance Viewer that is integrated with the Administration Console, allows the Administrator to view the collected PMI data. And the Performance Advisors provided in the Administrative Console or Tivoli Performance Viewer, provide configuration advice based on the collected PMI data and WebSphere's current configuration.

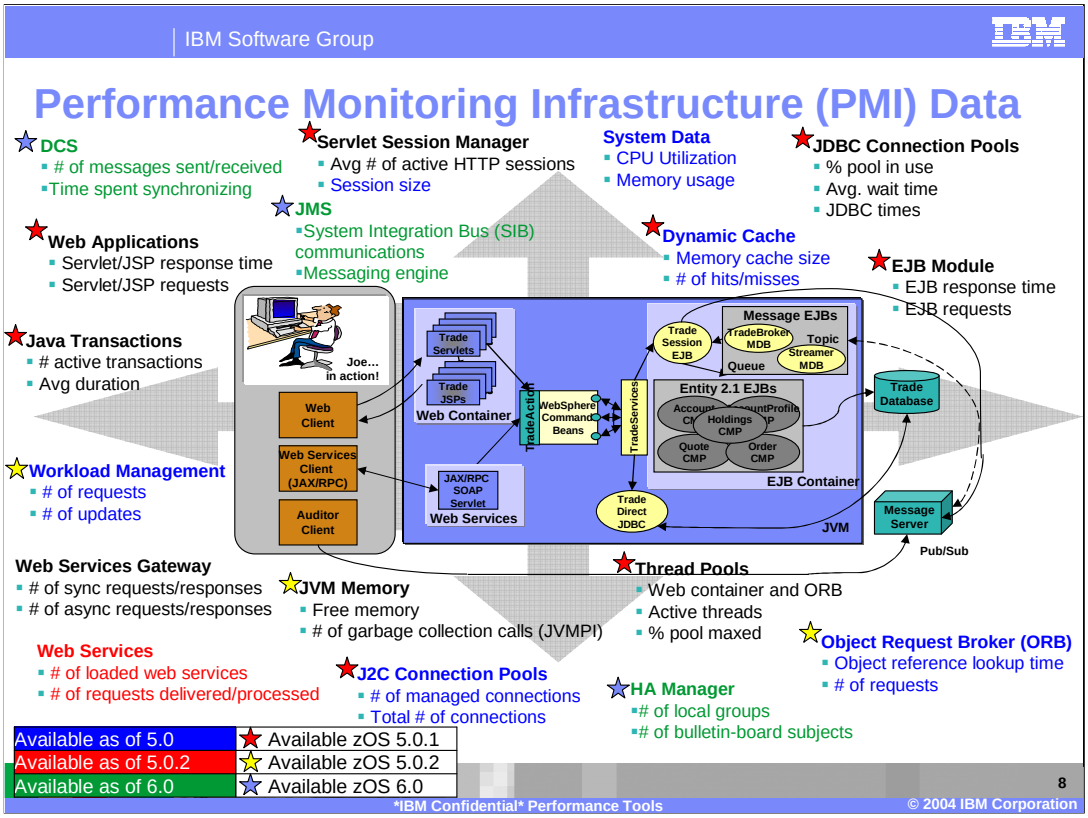
Section

Performance Monitoring Infrastructure (PMI)

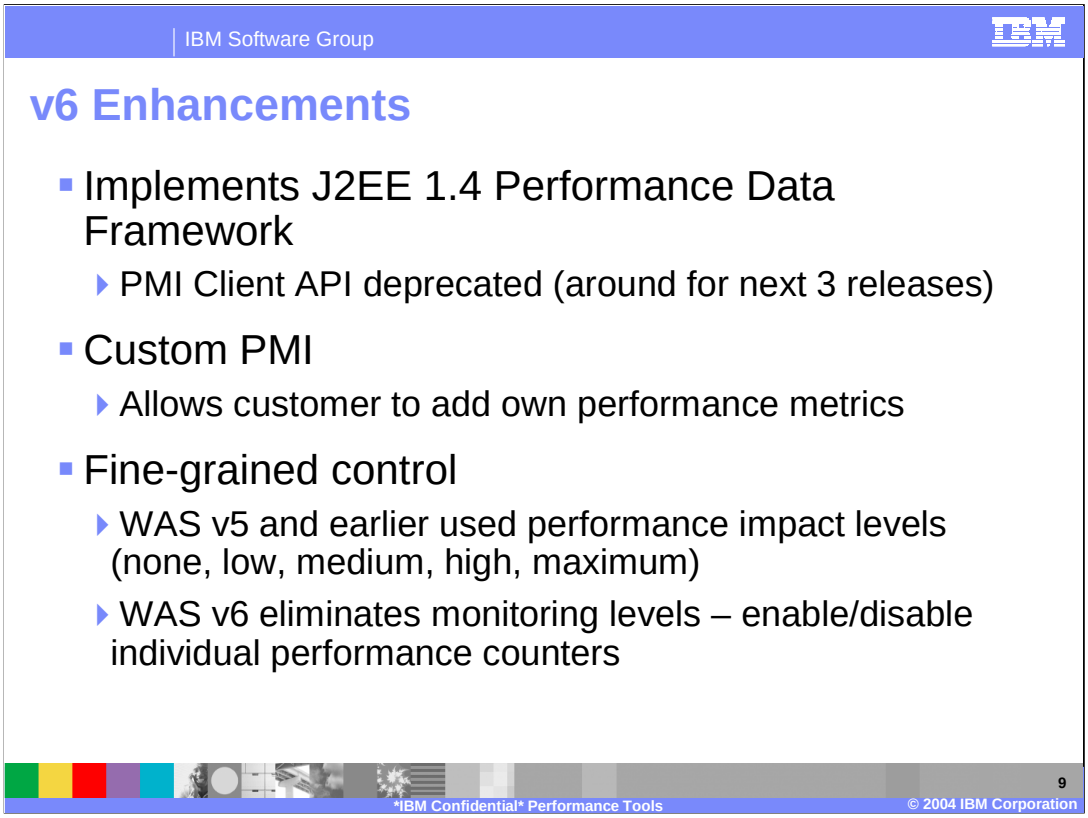


The Performance Monitoring Infrastructure (PMI) allows WebSphere to externalize its performance data to monitoring tools that can use the data to visualize and analyze its current performance.

The performance data that is generated falls into three general categories. The first category that collected PMI data falls into is customer application performance data. This includes information about your installed enterprise beans, servlets, web services, or even custom defined PMI data. We will talk more about the custom PMI data in an upcoming slide. The next category that PMI data falls into is WebSphere run-time resources. This includes information from WebSphere's JVM, active thread pools and database connection pools and most of the other resources that you have defined within WebSphere. The last category that PMI data falls into are performance data collected from the system where WebSphere is currently running. This includes performance data on the system's CPU usage or the total amount of free memory on the system. To see a complete list of all of the PMI data that is available from a WebSphere server, visit the WebSphere Information Center.



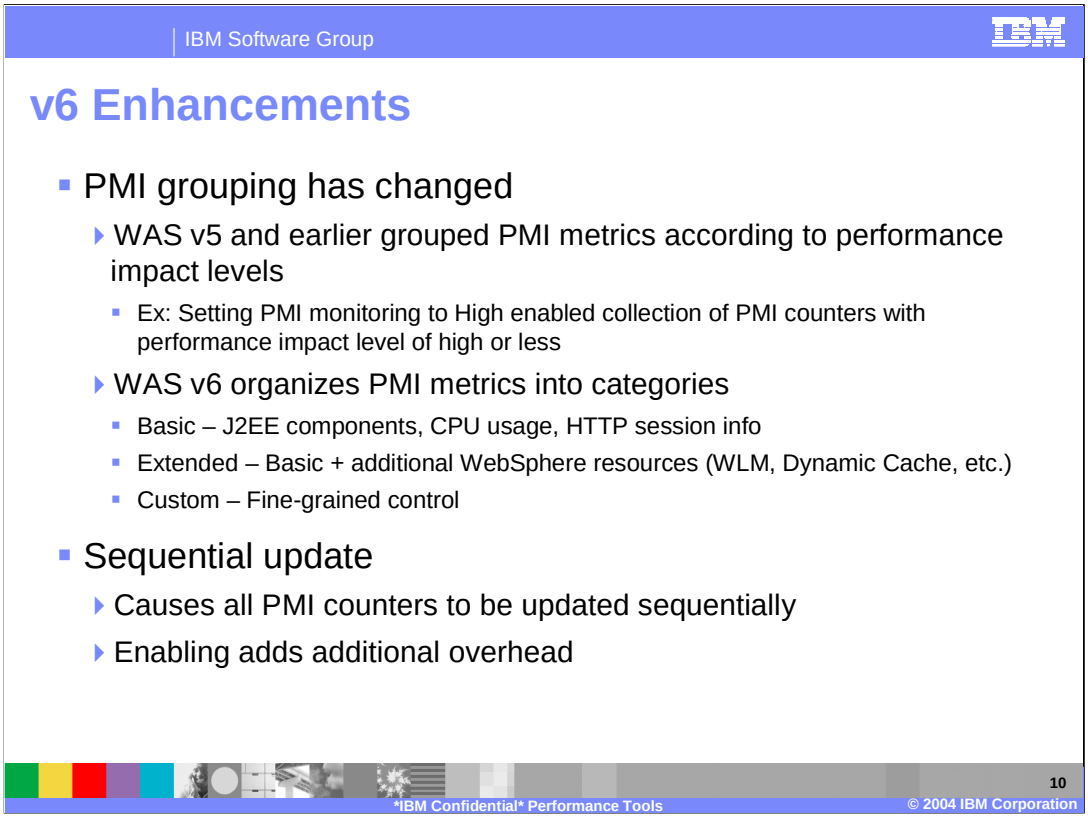
The graphic above gives us a view into the amount of PMI data that is collected by WebSphere. The graphic does depict the complete list of PMI metrics that are collected by the WebSphere environment. The diagram also attempts to depict when PMI data was added across the different releases of the distributed and zOS WebSphere platforms. Again for a complete list of available PMI data, visit the WebSphere Information Center. The Information Center also provides information about the release in which a PMI metric was added.



There have been a number of improvements to the usability and functionality of the PMI service in WebSphere version 6. WebSphere now implements the J2EE 1.4 Performance Data Framework. With the addition of the new standardized API, the older, proprietary PMI client API that has existed since WebSphere version 3.5.5 will be deprecated starting in version 6. This means that the API will be available to WebSphere customers for the next 3 releases of the product.

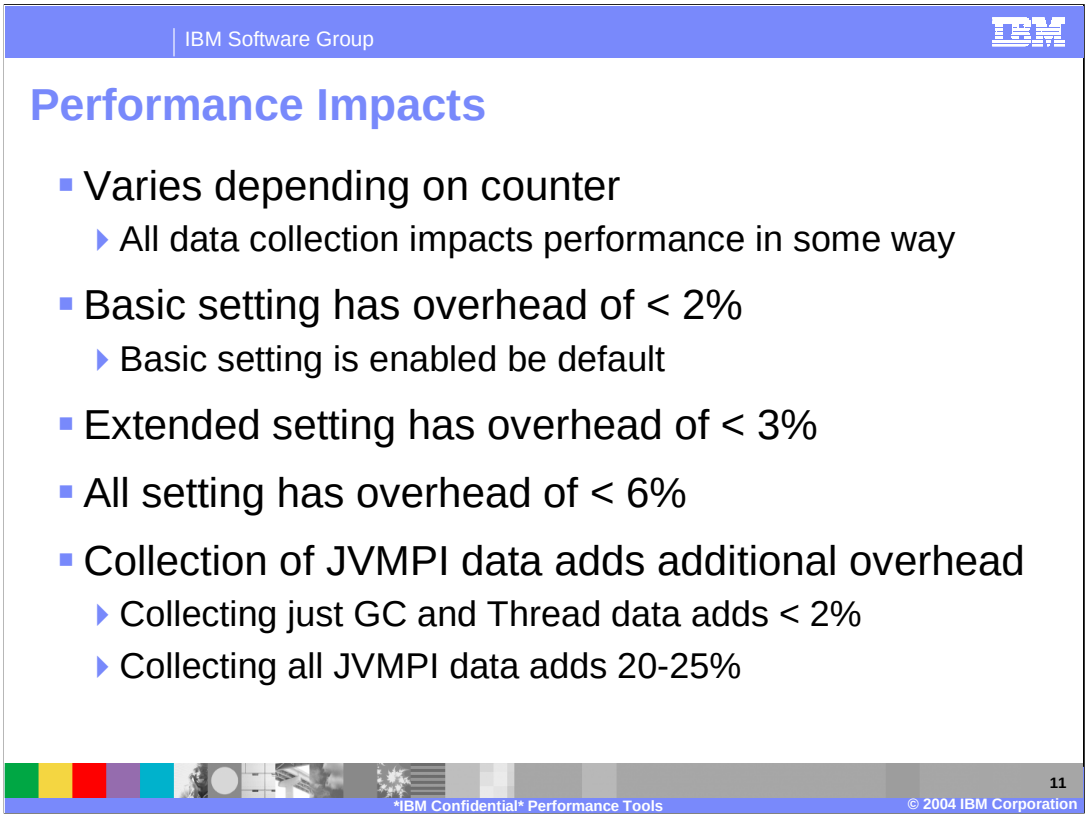
Additionally the ability for a customer to add their own custom PMI counters to WebSphere. This functionality allows a customer to define their own PMI counters based on their application. For example, a PMI counter that computes the shares of stock sold per day could be added to WebSphere for a brokerage application. Care should be taken, however, to not redefine PMI counters that are available by default through the application server. It is important to keep in mind that the more performance counters that are defined within WebSphere, the larger the performance impact on your application server.

The ability to select individual PMI counters has been added in version 6. Past versions of WebSphere used performance impact levels to enable or disable groups of PMI counters. Remember that performance impact levels were the low, medium, high and maximum PMI levels that could be set for the PMI service. WebSphere version 6 provides the ability to enable or disable individual PMI counters instead of groups of PMI counters. The fine-grained control is available through both the Administrative Console and the command line.



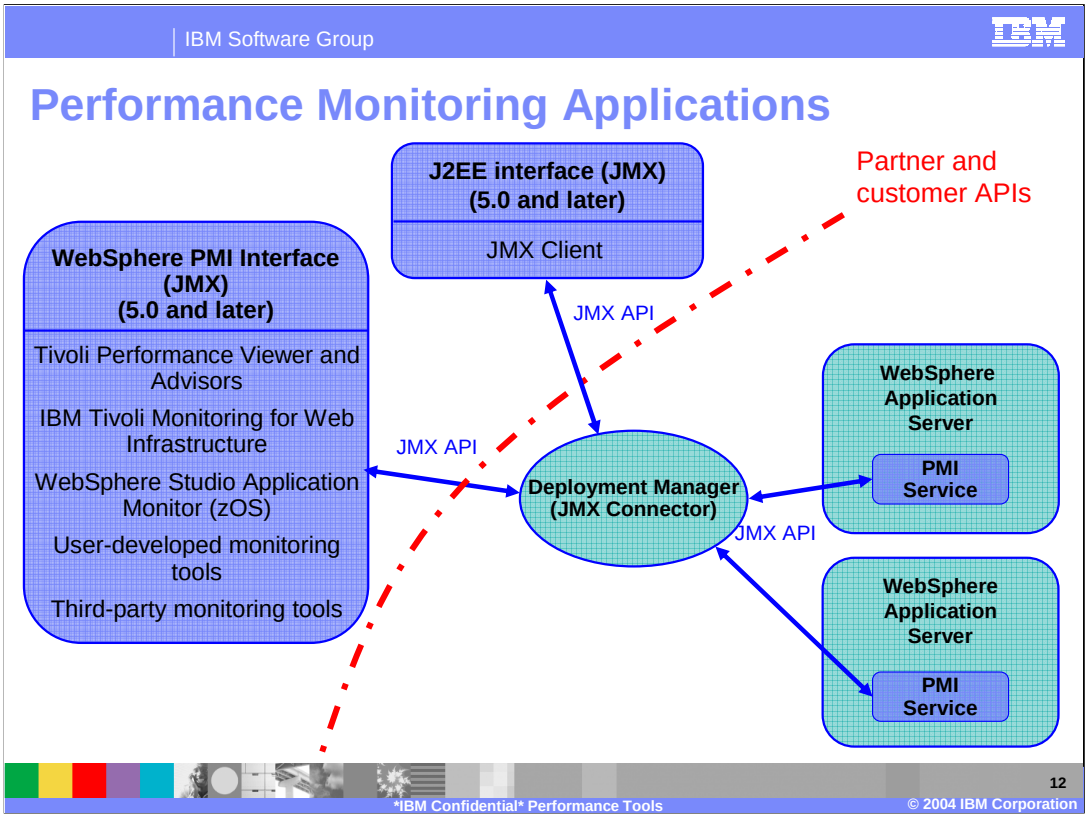
The way in which groups of PMI counters are selected has also changed in WebSphere version 6. Previous versions of WebSphere used performance impact levels to enable or disable groups of PMI counters. WebSphere version 6 has redefined the organization of PMI counters into different categories. In the Administrative Console you can either enable individual PMI counters through the Custom setting or you can enable or disable groups of PMI metrics using the defined categories. The categories that exist are Basic, Extended and All. The Basic setting enables the collection of performance data on the J2EE components, CPU usage and HTTP session information. The Extended setting extends the information collected by the Basic setting by additionally collecting performance data on WebSphere resources.

You can now additionally enable whether all PMI counters are updated sequentially. Sequential updates of counters provides more accurate performance data, but at the same time places additional overhead on the server. By default the application server only sequentially updates the RangeStatistic and BoundedRangeStatistic. When sequential update is enabled, all Performance Data Framework statistic types are updated sequentially. For more information on the sequential update option visit the WebSphere Information Center.



A performance impact is incurred on the application server when PMI data is enabled. The extent of the performance impact is determined by the amount and type of PMI counters that are collected. Counters that add up the amount of times a method is called will cause less of a performance impact than a counter that averages the amount of time spent executing a method call.

You can expect a performance impact of less than 2% when you have the Basic PMI setting enabled. It is also important to note, that the PMI service is enabled by default and set to the Basic setting. This is a change from previous versions of WebSphere. A performance impact of less than 3% can be expected for the Extended setting. And when all PMI counters are being collected you can expect a performance impact of less than 6%. When collecting additional PMI data from WebSphere's JVM using the Java Virtual Machine Profiler Interface (JVMPI), you can expect additional overhead over the value stated above. Collecting just garbage collection and thread PMI data adds an additional performance impact of less than 2% to your application server. This means that if you are already collecting PMI data using the Basic setting and you additionally collect JVM garbage collection and thread data, your application server would incur a performance impact of approximately 4%. Additionally if you collect all JVMPI PMI data, your application server will experience an additional performance impact of approximately 20 to 25%. Because of the JVMPI PMI modules high performance impact on your application server it should only be used in a development environment or in extreme cases in your production environment.



Performance data can be retrieved from WebSphere using the JMX interface. This functionality has remained the same from WebSphere version 5. Also note that the picture does not show the PMI client interface. Remember that the PMI client interface will be available for the next 3 release of WebSphere. Future development of performance monitoring products should be developed against the JMX interface.

The performance monitoring solutions that were available for previous releases of WebSphere remain the same for this release. Take note, however, that the products are being moved to the JMX interface for future compliancy with WebSphere.

Section

Tivoli Performance Viewer (TPV)

Tivoli Performance Viewer (TPV)

- Displays PMI data collected from local/remote application servers
  - Summary reports show key areas of contention
  - Graphical/tabular views of raw PMI data
  - Optionally save collected PMI data to logs
- Provides configuration advice via Performance Advisor section
  - Tuning advice formulated from gathered PMI and configuration data
- Integrated into WebSphere Administrative Console

Tivoli Performance Viewer has gone under several different names throughout the life of the WebSphere product. In WebSphere 3.5.x-4.x it was known as Resource Analyzer. In WebSphere v5.x it was rebranded as Tivoli Performance Viewer.

•Resource Analyzer only works with WebSphere 3.5.x-4.x.

•Tivoli Performance Viewer only works with WebSphere 5.x and higher.

Tivoli Performance Viewer is capable of connecting directly to a base application server or to the Deployment Manager in a Network Deployment configuration.

The Performance Advisors will be discussed at length later in the presentation.

Tivoli Performance Viewer is accessed through the Administrative Console.

In order to receive information in the Summary Reports you need to enable the following PMI monitoring levels.

•Monitoring levels must be set to HIGH in order to receive information in the Servlets, EJBs, Web Container Pools, ORB Pools, and Connection Pools table. To receive information in the EJB Methods table, set the monitoring level to MAXIMUM for the EJB(s) that you want to collect method-level metrics from.

When viewing data in the table and graphs in Tivoli Performance Viewer, you are able to select which performance metrics that you want displayed. This is useful when you want to correlate several different PMI metrics.

Recording logs can be useful in the following situations:

•You need to forward your application server's performance data on to someone in your team or to support for them to evaluate.

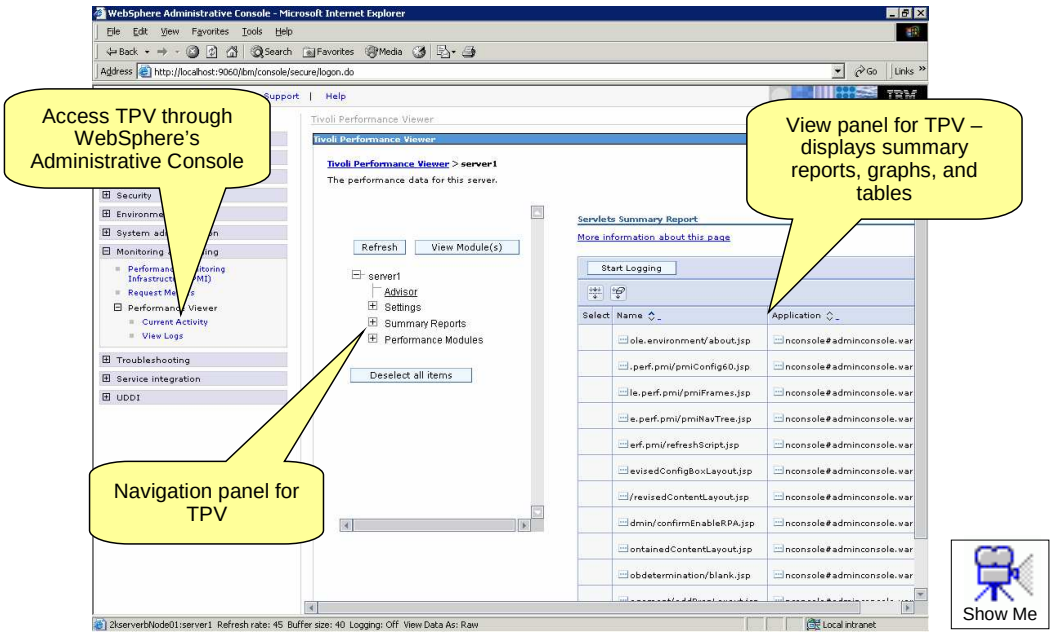
•You need to have historical data for comparing different performance tests.

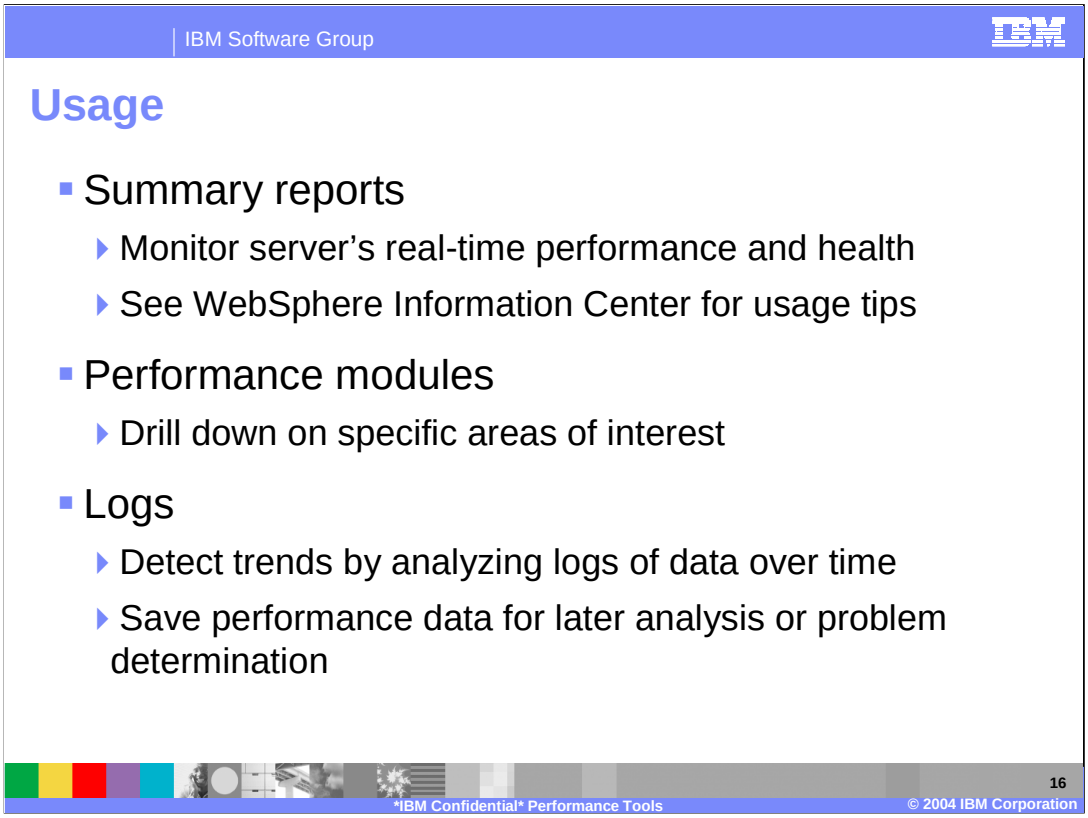
Keep in mind the following when deciding on whether to use binary or XML log files:

•The XML logs will be larger than the binary logs, so if space is a concern for your system you should be aware of this.

•The XML log provides the added flexibility that it could read/replayed by additional tools for further analysis. The binary logs are proprietary and do not offer the same amount of flexibility.

Tivoli Performance Viewer





Tivoli Performance Viewer is instrumental in analyzing your running application server's resources to determine the best settings for pool sizes, JVM heap sizes, caches sizes, etc. The raw metric data is collected along with the Performance Advisor functionality make Tivoli Performance Viewer an invaluable tool when tuning/monitoring your WebSphere environment in development, testing, and production.

For more information on the various metrics available via PMI, see the Information Center.

Monitor real-time performance
- Servlet request response times
- Enterprise bean method calls
- Data sources in use
- Wait time on data sources
- Concurrent waiters on data sources

Determine optimal resource configurations
- Allocated memory
- Database connection pool size
- Enterprise bean objects cache size

Detect trends by analyzing logs of data over time

Gauge application server load
- Percentage of pools in use
- CPU load
- JVM heap size/amount in use

Determine average wait time for clients
- Response times for requests at various containers

Section

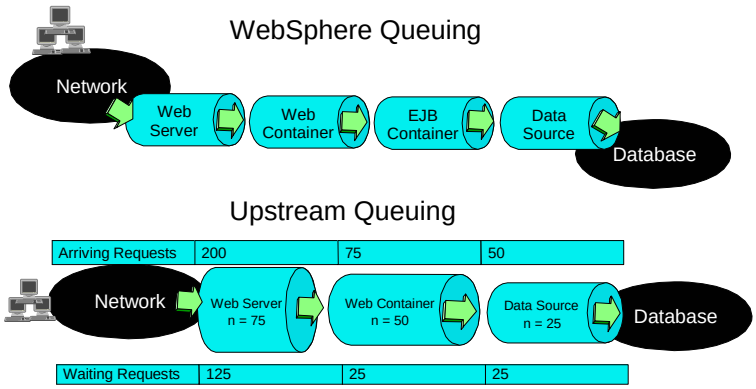
WebSphere Performance Advisors

WebSphere Performance Advisors

- Provide configuration advice based on collected PMI data
- Advice based on basic rules of thumb for tuning WebSphere Application Server
  - Rules are IBM defined and non-configurable
- **Does not** automatically tune WebSphere based on advice
  - Administrator must manually apply recommendations
  - Suggested settings need to be checked against baseline performance to verify improvement

WebSphere Upstream Queuing

- Upstream queuing attempts to allow more work to be done by limiting the number of connections at each tier of the application
- Performance Advisor rule set employs this best practice when generating advice



There are several different levels to set up a funneling affect of the work being done by WebSphere.

Design the pooling system to allow as much work on the database and application server as possible without over-stressing each area.

ORB Thread pool size in the EJB container is a soft limit, whereas all other connections can be implemented as a hard limit.

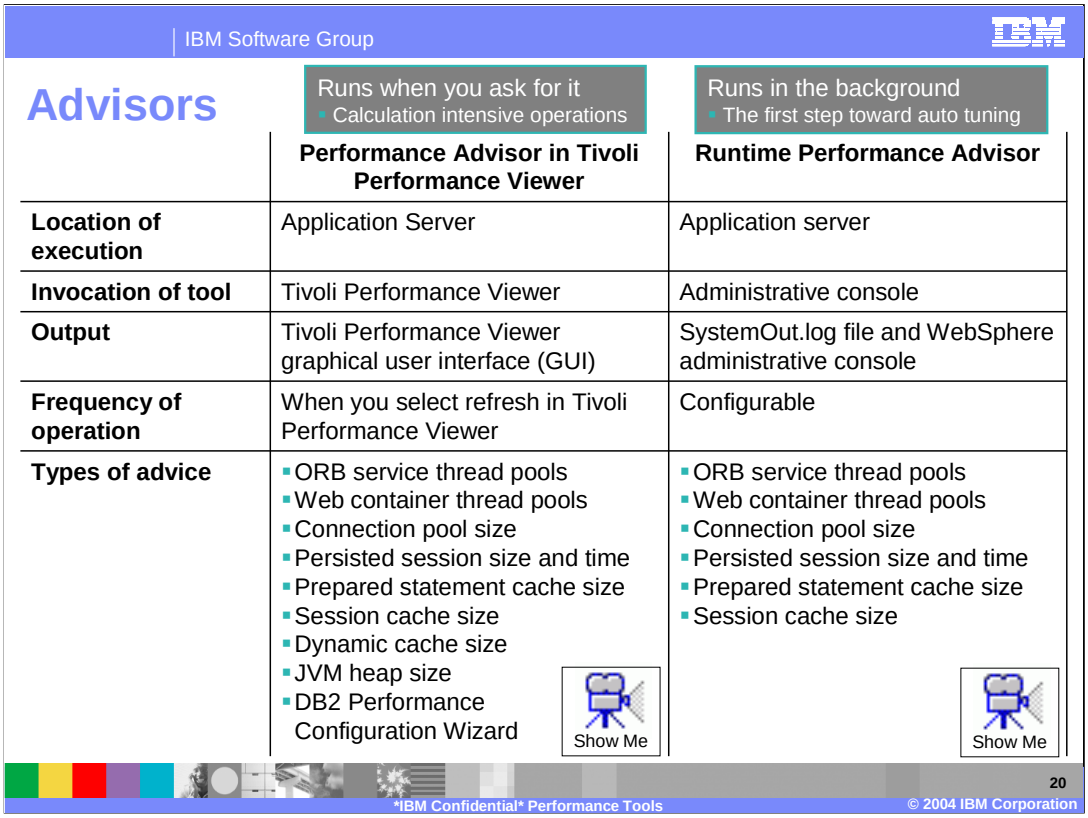
The use of the different pools can be used to funnel the work in order to allow the resources to be used by the proper processes.

It is important not to overload any one tier because then you will have poor performance due to waiting threads. This is one of the most difficult things to do during performance

For instance, if a database is being overworked and causing slow response time, you may need to lower the number of connections. This allows those active connections the resources needed to perform the needed queries.

Most of these settings can be monitored by the Tivoli Performance Viewer and changed through the Administrative Console.

The objective is to achieve maximum throughput within the environment or solution.



Examples of advice that Performance Advisor would give in certain situations follows:

Data Sources

•Situation: Prepared statement discard rate too high and heap space is available.
Advice: Increase statement cache size

Thread pools (Orb, Web Container, Data Source (DS))

•Situation: If number of connections is low (= to the min).
Advice: Decrease pool size

•Situation: If all DS connections are heavily used and heap space is available.
Advice: Increase max pool size

•Situation: Size of the pool is fluctuating a lot (high variance), possibly indicating batch processing and wasted resources
Advice: Decrease pool size

JVM heap size

•Situation: If heap size is too small (< 256 MB)
Advice: Increase the pool size to a value > 256 MB

Unbounded thread pools

•Situation: Threads added to an unbounded pool are not pooled
Advice: If average number of threads is higher than the pool size, then the pool should be increased in order to allow better pooling

Sessions

•Situation: Read/Write time/size too large
Advice: Warn of application problem.

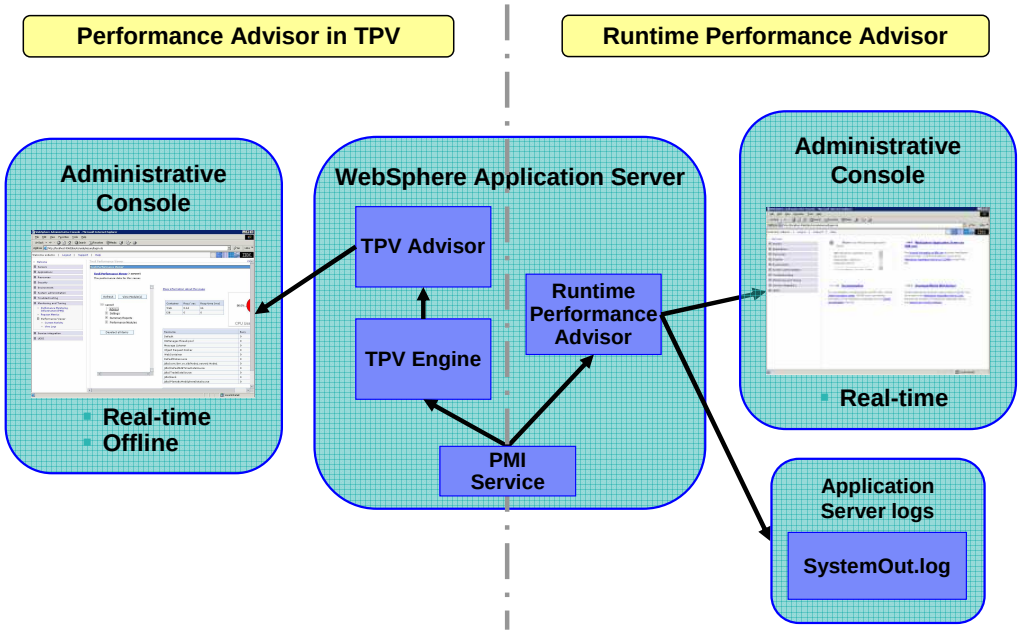
•Situation: Number of live sessions is greater than the session cache and memory is available
Advice: Increase Session Cache

•Situation: Requests turned down because there is no room for new sessions
Advice: There are either two many active sessions or the cache size is too small

DB2 Performance Configuration Wizard

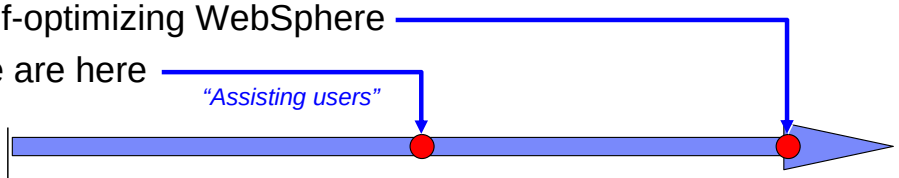
•Situation: DB2 database detected in configuration
Advice: Use the DB2 Performance Wizard to tune the indicated database.

Architecture

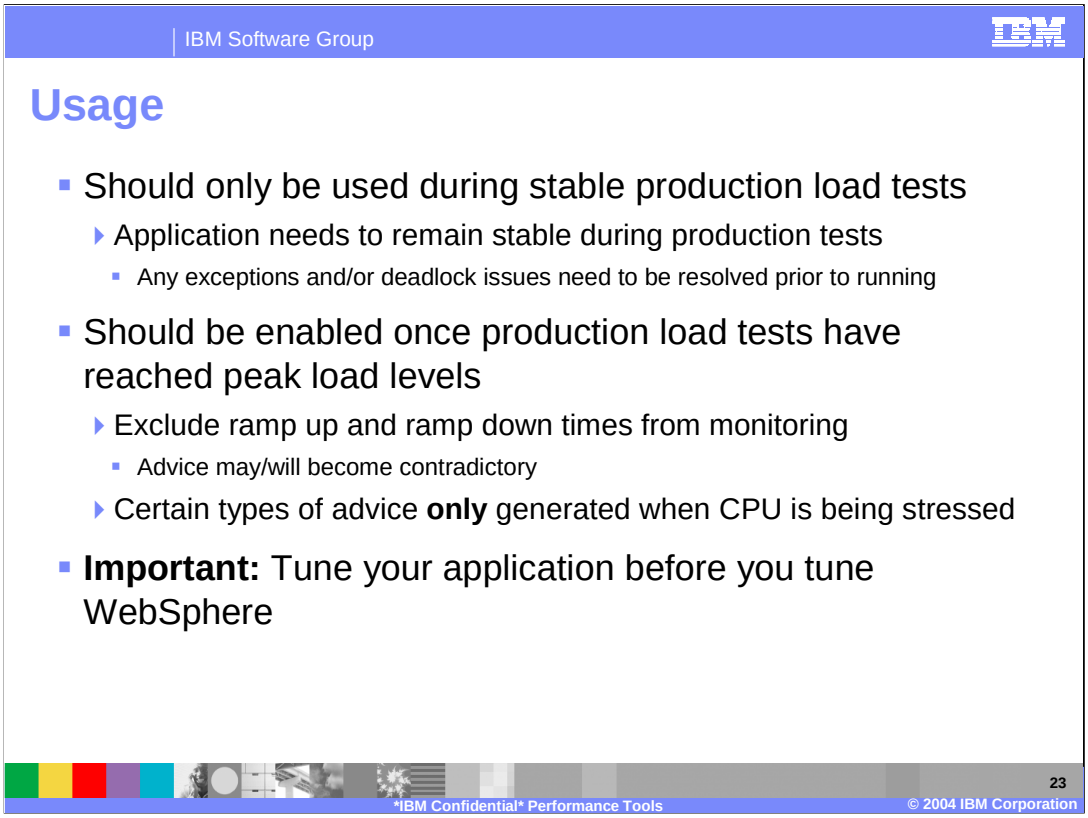


Roadmap to Self-Optimizing

- The goal
  - Self-optimizing WebSphere
  - We are here



Level	1: Basic	2: Managed	3: Predictive	4: Adaptive	5: Autonomic
Description	Rely on reports, product and manual actions to manage IT components	Management software in place to provide facilitation and automation of IT tasks	Individual components and **systems management tools able to analyze and recommend actions**	IT components collectively able to monitor, analyze and take action with minimal human intervention	IT components collectively and **automatically managed by business rules and policies**
Benefits		Greater system awareness Improved productivity	Reduced dependency on deep skills **Faster/better decision making**	Balanced human/system interaction IT agility and resiliency	**Business policy drives IT management** Business agility and resiliency



When using the Performance Advisors, CPU utilization above 50% is needed before the advice is generated. Typically when running your production level load, you would push the CPU Usage to 80-100% before turning on one of the Performance Advisors.

The following are some things to consider when using the Performance Advisor for tuning:

•Contradictory advice will be generated if the load on the system under test changes. This is because the collected PMI data will show a different type of environment and the advice will in turn shift. To avoid this, always run the Advisors while simulating the load WebSphere will experience during deployment (usually peak load).

•If the pool size's minimum and maximum values are the same, the Performance Advisor rules are much more likely to give contradictory advice when load fluctuates.

•Amount of system activity determined by the amount of CPU usage. Doesn't take disk activity, network activity, memory usage, etc. in to account to get a more realistic view of system load.

Recommendations are only generated when CPU load reaches 50% and higher.

•Performance Advisors from different application servers may give contradictory advice on the same node resources. This is because the application servers ONLY take into account how they are currently employing the resource.

In this situation if the advice differs tremendously from the different advisors, take the generated advice into consideration and make the changes that make sense. If all advisors, however, are giving the same recommendations then you should seriously consider the suggested changes.

•If the Performance Advisors suggests setting a pool size to X, it is recommended that the minimum value equal to X/2 and maximum value equal to X.

•If the Performance Advisor suggests setting the Prepared Statement Cache value to a certain setting, check the amount of memory that is available before setting. The advisors do not take into account the amount of actual physical memory available on the system before making the suggestions.

Section

Summary and Reference

Summary

- WebSphere provides many tools to assist in tracking down performance problems
- Performance Monitoring Infrastructure (PMI) externalizes performance data
- Tivoli Performance Viewer (TPV) allows you to visualize and analyze collected PMI data
- WebSphere Performance Advisors provide configuration advice based on collected PMI data

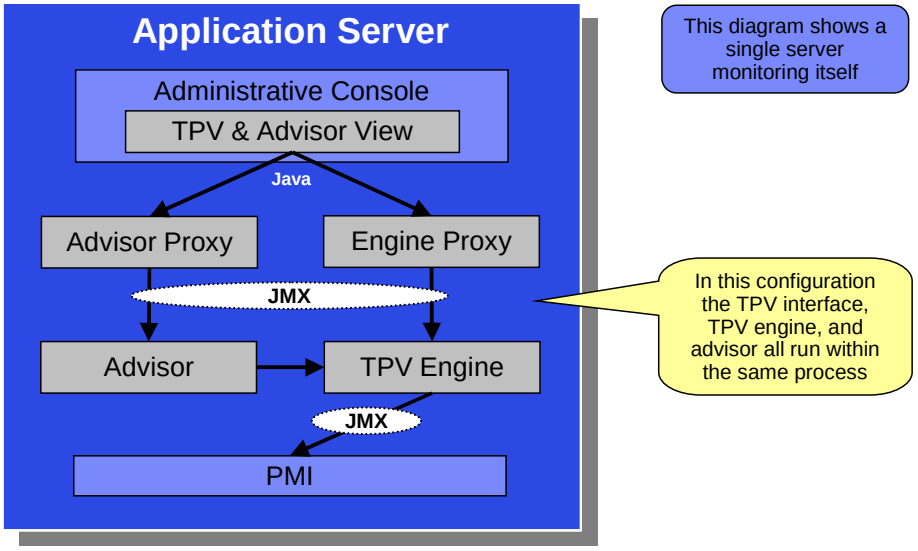
Section

Appendix

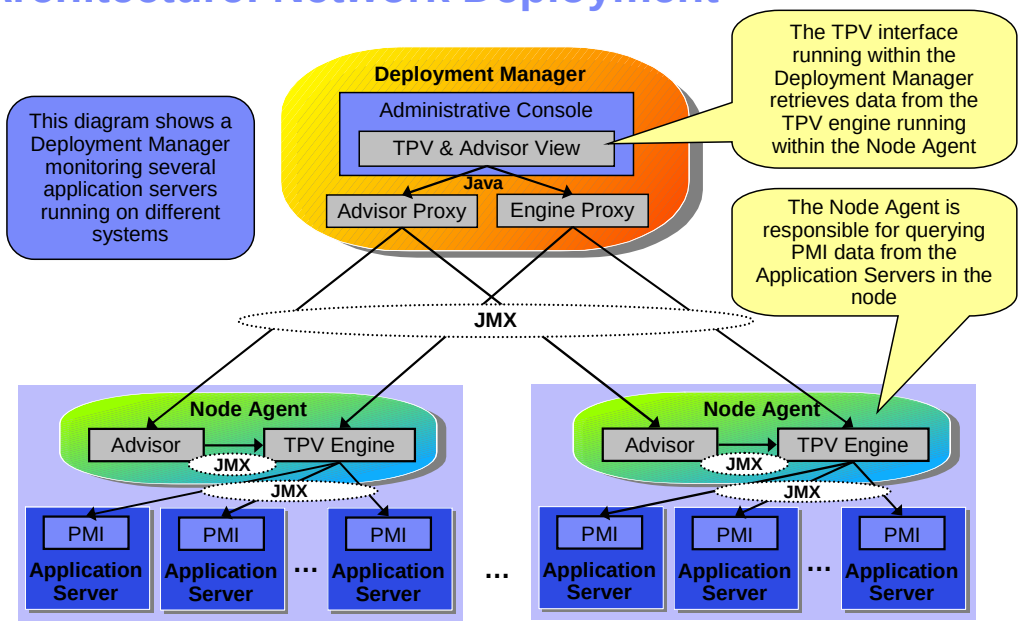
Section

Tivoli Performance Viewer Architecture

Architecture: Single Server



Architecture: Network Deployment



Integrated Client Components

- TPV and performance advisor GUI
  - Displays PMI and Advisor data in Administrative Console
  - Administrative Console plug-in
- Advisor and TPV Engine Proxies
  - Allows Administrative Console to communicate with Advisor and/or TPV Engine

Integrated Client Components (cont.)

- TPV engine
  - Collects active PMI metrics
  - Buffers data points to provide a brief history
  - Transforms data to display either current value, change of value over previous point, or change of value since buffer reset
  - Logs data to file for persistence
- TPV advisor
  - Analyzes PMI data and gives advice based on provided rules

Template Revision: 10/01/2004 4:46 PM

Trademarks and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	iSeries	OS/400	Informix	WebSphere
IBM(logo)	pSeries	AIX	Cloudscape	MQSeries
e(logo)business	xSeries	CICS	DB2 Universal Database	DB2
Tivoli	zSeries	OS/390	IMS	Lotus

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both. Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are
trademarks of Intel Corporation in the United States, other countries, or both. UNIX is a registered trademark of The Open Group in the United States and other countries. Linux is a registered trademark of Linus Torvalds. Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance 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 or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.