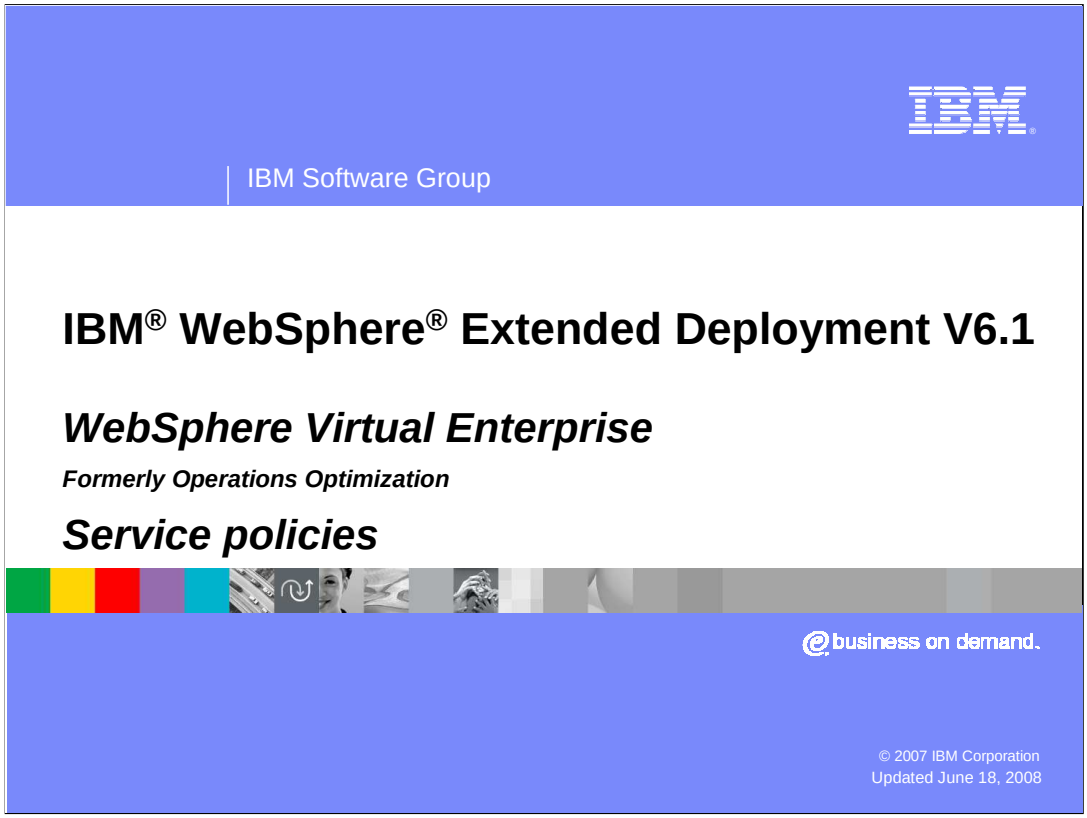
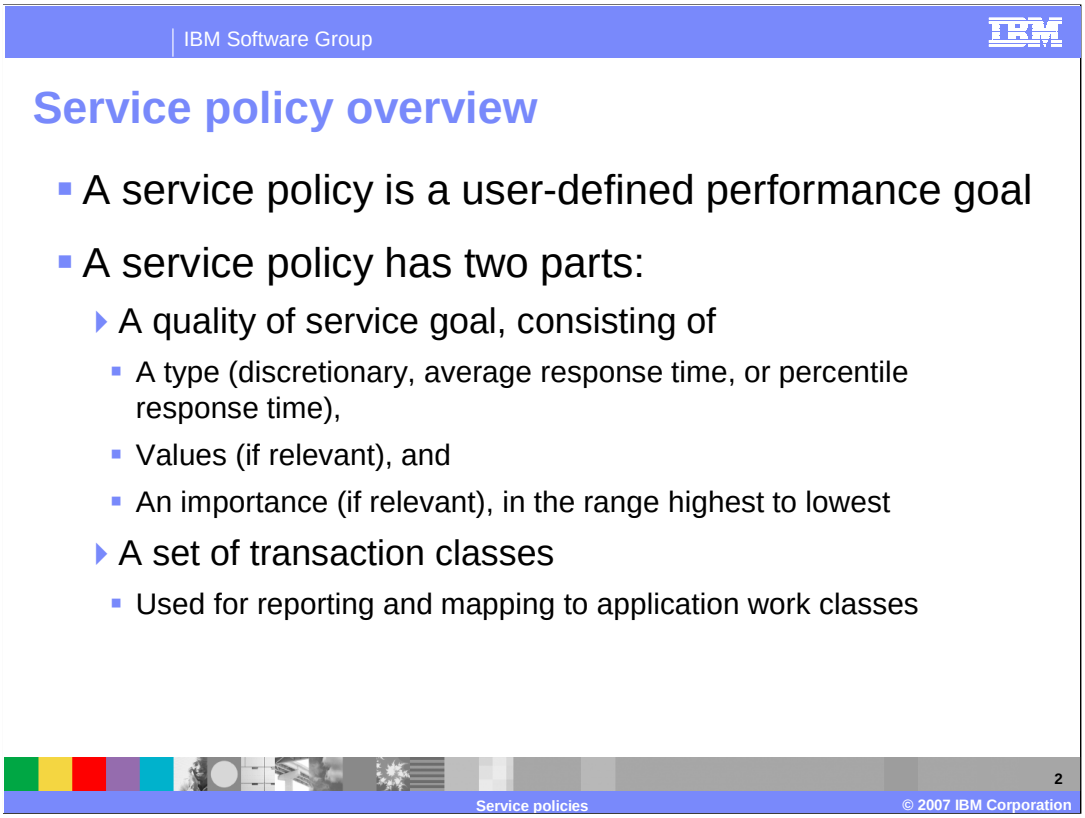


This presentation will cover WebSphere Extended Deployment V6.1 service policies.

This module was originally recorded for WebSphere Extended Deployment Operations Optimization, which is now called WebSphere Virtual Enterprise. Though the module uses the previous names, the technical material covered is still accurate.



Most Web and application server products route requests on a first-come-first-serve basis. However, since not all requests are of equal importance this is not the best policy in all cases. WebSphere Extended Deployment allows you to differentiate application service levels according to your business requirements. As user requests enter the on demand router, they are classified, prioritized, queued and routed to servers based on application operational policies that are tied to business goals. The on demand router works with other components within WebSphere Extended Deployment to optimize application performance according to these policies.

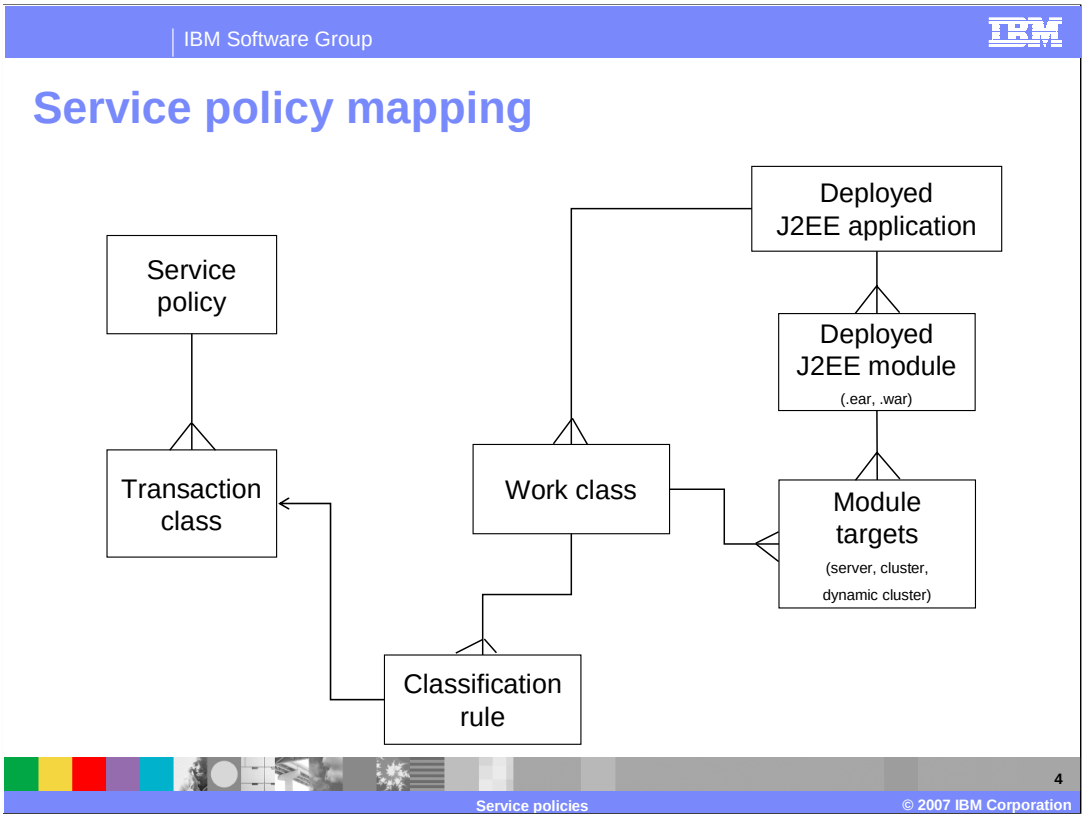
WebSphere Extended Deployment introduces the concept of service policies to decide how to manage work requests according to user-defined performance goals. A service policy is comprised of two parts. First, the service policy specifies a performance goal the user desires to be achieved. For OLTP applications, this is a response time based goal, such as 500 milliseconds average response time for associated requests. For long running applications this is a queue time based goal. For instance, you are willing to wait up to 10 minutes for this job to complete processing. This service goal also includes an importance level to inform Extended Deployment of the relative priority of different classes of work. In addition to defining goals, each service policy contains one or more transaction classes that are used for reporting and mapping to application work classes.

Transaction classes

- Each service policy contains one or more transaction classes
  - A default transaction class is created for each service policy
    - This is sufficient unless you need finer-grained monitoring or reporting
  - Transaction classes enable work classes to be mapped to the service policy
  - Enables finer-grained monitoring by application, server, or work class

© 2007 IBM Corporation

Each service policy contains a single transaction class by default. However, you can create multiple transaction classes within a service policy, enabling you to differentiate between requests that you want to handle with the same class of service. Transaction classes enable work classes to be mapped to the service policy, so work requests will be handled according to their corresponding user-defined goal.



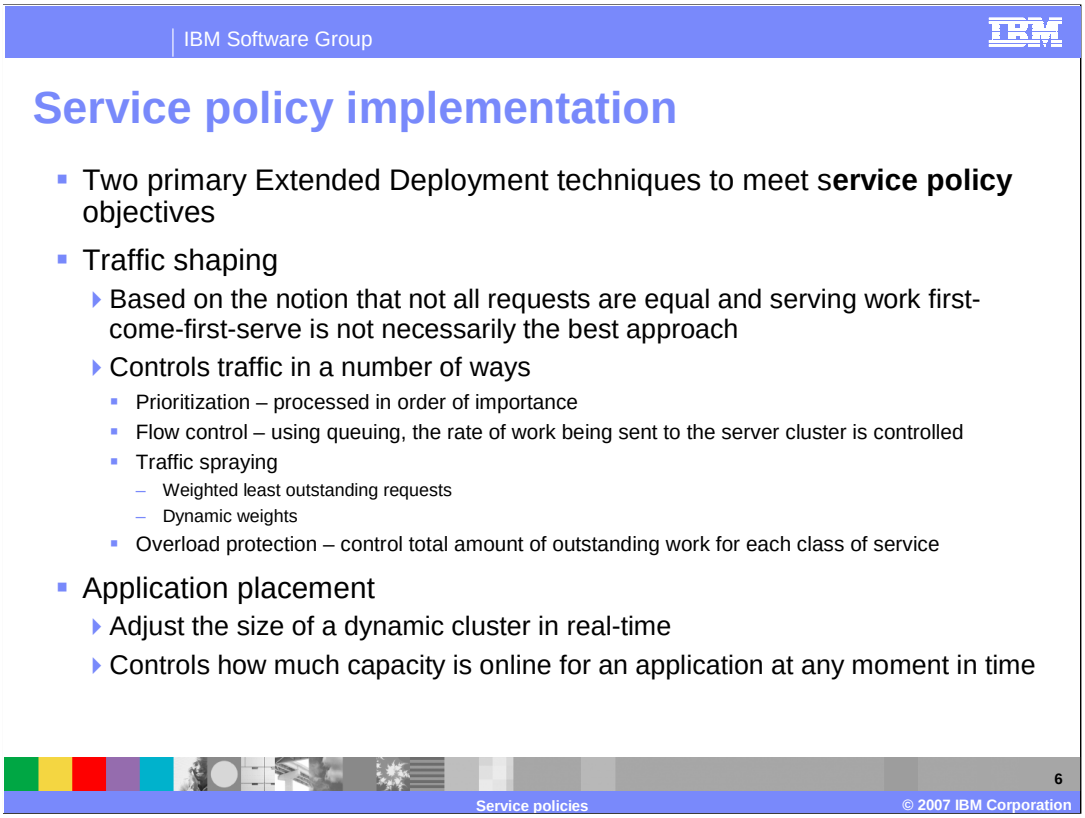
This slide illustrates how a service policy maps to applications. First, work classes can define classification rules for HTTP and SOAP requests. Work classes may also define classification rules for J2EE modules like a Web application or EJB jar file. Simply put, a work class is a set of rules used for mapping incoming requests to transaction classes. The associated transaction class is, in turn, contained by a service policy that defines the goals and importance for that request. Essentially, the service policy mapping gives the work class its goal.

Example service policy

- Messages:
  - Service policy A contains members:
    - Transaction class A, associated with:
      - /trade/* - all HTTP URLs in “trade” J2EE module
      - /portfolio/* - all HTTP URLs in “portfolio” J2EE module
    - Transaction class B, associated with: …
  - Goal: no more than 5% of the response times should be over 1 second, with importance of “very high”

© 2007 IBM Corporation

Here is an example of a service policy. Service policy A contains two transaction classes, transaction class A and transaction class B, that will be used to determine which work requests will use this policy. Transaction class A specifies that service policy A should be used for incoming requests from all HTTP URLs in the trade J2EE module and for all HTTP URLs in the portfolio J2EE module. The service policy also has set goals to dictate that this work is of “very high” importance and that no more than 5% of the response times should be over 1 second. Note that when creating your own service policies, ensure that the goals you set are consistent with both the capabilities of your application and reasonable end-user expectations. The on demand router will attempt to meet your specified goals, but cannot make your application run faster.



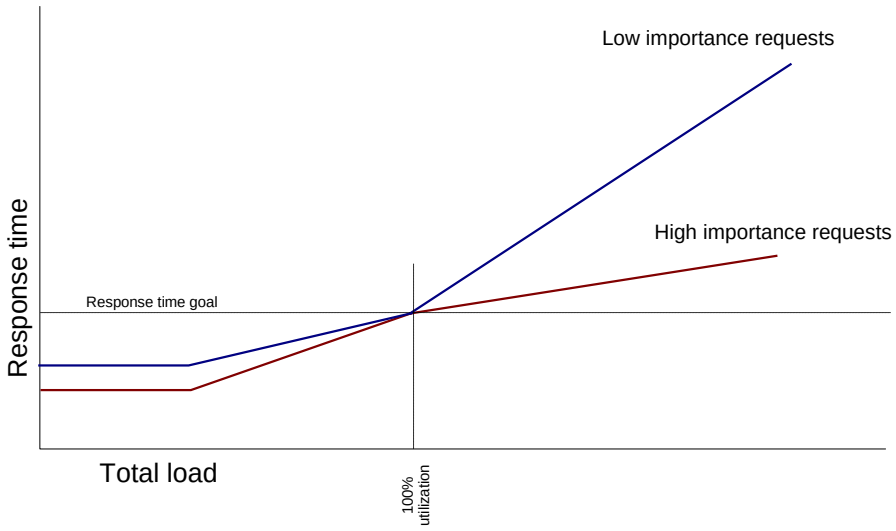
WebSphere Extended deployment employs two techniques to ensure service policy objectives are met, traffic shaping and application placement.

As user requests enter the on demand router, they are classified, prioritized, queued and routed to servers based on operational policies that are tied to business goals. The application placement controller will attempt to ensure that target servers for the requests are not overloaded by adjusting dynamic workload management routing weights and, if necessary, starting or stopping server instances.

Traffic shaping attempts to handle different workloads with different priorities to ensure all work requests are processed efficiently, as dictated by the service policies you set.

In WebSphere Extended Deployment for z/OS, traffic routing works in conjunction with zWLM to extend its capabilities.

Heavy load conditions



© 2007 IBM Corporation

In a heavily loaded system it may be impossible to meet the service policy goals for all applications. In this case the on demand router will use the relative importance specified in each service policy to prioritize requests. The on demand router will not starve low priority requests, but will attempt to “share the pain,” so more important requests come closer to meeting their goals than do less important requests.

Summary

- Service policies define a level of importance and a response time goal
- Each service policy contains one or more transaction classes
  - A default transaction class is created for each service policy
    - This is sufficient unless you need finer-grained monitoring or reporting
  - Transaction classes enable work classes to be mapped to the service policy

© 2007 IBM Corporation

In summary, a service policy defines a level of importance and a response time goal. A service policy also has one or more transaction classes associated with it. Transaction classes enable work classes to match incoming requests to classification rules in order to decide which policy applies to a given request. Ultimately, a service policy allows you to describe how work requests in your environment should be treated.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject= Feedback about XD61_ServicePolicies.ppt

© 2007 IBM Corporation

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

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

IBM WebSphere

J2EE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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 or changes in the products or programs 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 (for example, 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.

© Copyright International Business Machines Corporation 2007. All rights reserved.

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.