1.	Record Nr.	UNINA9910811826403321
	Titolo	IBM WebSphere V5.1 performance, scalability, and high availability / / [Birgit Roehm ... et al.]
	Pubbl/distr/stampa	White Plains, NY, : IBM, International Technical Support Organization, c2004
	Edizione	[2nd ed.]
	Descrizione fisica	xxvi, 980 p. : ill
	Collana	WebSphere handbook series IBM redbooks
	Altri autori (Persone)	RoehmBirgit
	Disciplina	005.3/76
	Soggetti	Web servers - Computer programs Web site development
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	"June 2004." "SG24-6198-01."
	Nota di bibliografia	Includes bibliographical references (p. 935-945) and index.
	Nota di contenuto	Front cover -- Contents -- Notices -- Trademarks -- Preface -- The team that wrote this redbook -- Become a published author -- Comments welcome -- Summary of changes -- June 2004, Second Edition -- Part 1 Getting started -- Chapter 1. Overview and key concepts -- 1.1 Objectives -- 1.1.1 Scalability -- 1.1.2 Workload management -- 1.1.3 Availability -- 1.1.4 Maintainability -- 1.1.5 Session state -- 1.1.6 Performance impacts of WebSphere Application Server security -- 1.2 WebSphere Application Server architecture -- 1.2.1 WebSphere Application Server Network Deployment components -- 1.2.2 Web clients -- 1.2.3 Java clients -- 1.3 Workload management -- 1.3.1 Web server workload management -- 1.3.2 Plug-in workload management -- 1.3.3 Workload management using WebSphere clustering -- 1.3.4 Enterprise Java Services workload management -- 1.4 Managing session state among servers -- 1.4.1 HTTP sessions and the session management facility -- 1.4.2 EJB sessions or transactions -- 1.4.3 Server affinity -- 1.5 Performance improvements over previous versions -- 1.6 The structure of this redbook -- Chapter 2. Design for scalability -- 2.1 Scaling your infrastructure -- 2.2 Understanding the application environment -- 2.3 Categorizing your workload -- 2.3.1 Workload patterns and Web site classifications -- 2.3.2 Workload

characteristics -- 2.4 Determining the most affected components -- 2.5 Selecting the scaling techniques to apply -- 2.5.1 Using a faster machine -- 2.5.2 Creating a cluster of machines -- 2.5.3 Using appliance servers -- 2.5.4 Segmenting the workload -- 2.5.5 Batch requests -- 2.5.6 Aggregating user data -- 2.5.7 Managing connections -- 2.5.8 Caching -- 2.6 Applying the technique(s) -- 2.7 Re-evaluating -- Chapter 3. Introduction to topologies -- 3.1 J2EE tiers model -- 3.2 Topology selection criteria -- 3.3 Strategies for scalability.

3.4 Single machine topology -- 3.5 Separating the Web server -- 3.6 Separating the database server -- 3.7 Vertical scaling -- 3.8 Horizontal scaling with clusters -- 3.8.1 Horizontal scaling with IP sprayer -- 3.9 One WebSphere administrative cell versus many -- 3.10 Multiple clusters on one node versus one cluster per node -- 3.11 The sample topology -- 3.12 Topologies and high availability -- 3.12.1 Using WebSphere Load Balancer custom advisor -- 3.13 Closing thoughts on topologies -- 3.14 Topology selection summary -- Part 2 Distributing the workload -- Chapter 4. Web server load balancing -- 4.1 Introduction -- 4.1.1 Scalability -- 4.1.2 Availability -- 4.1.3 Performance -- 4.2 IBM WebSphere Edge Components -- 4.3 Load Balancer overview -- 4.3.1 Load Balancer topologies -- 4.3.2 Installation and configuration -- 4.3.3 Setting up the cluster machines -- 4.3.4 Configuring a Web server cluster -- 4.3.5 Testing the configuration -- 4.4 Advisors -- 4.4.1 Custom advisors -- 4.4.2 Using WebSphere Application Server sample custom advisor -- 4.5 Server affinity -- 4.5.1 "Stickyness" to source IP address -- 4.5.2 Passive cookie affinity -- 4.5.3 Active cookie affinity -- 4.5.4 URI -- 4.5.5 SSL session ID -- 4.6 Caching Proxy -- 4.6.1 Forward proxy -- 4.6.2 Reverse proxy (IP forwarding) -- 4.6.3 Load Balancing -- 4.6.4 Dynamic caching -- Chapter 5. Plug-in workload management and failover -- 5.1 Introduction -- 5.2 The embedded HTTP transport -- 5.3 Setting up the Web containers -- 5.3.1 Virtual hosts -- 5.3.2 Transports -- 5.3.3 Creating clusters and cluster members -- 5.4 WebSphere plug-in workload management -- 5.4.1 Processing requests -- 5.4.2 The plug-in configuration file -- 5.4.3 Generation of the plug-in configuration file -- 5.4.4 Plug-in workload management and failover policies -- 5.5 Session management -- 5.5.1 Session affinity.

5.5.2 Session failover inside the plug-in -- 5.5.3 Session identifiers -- 5.5.4 Session persistence and failover -- 5.6 Troubleshooting the Web server plug-in -- 5.6.1 Logging -- 5.6.2 Trace -- 5.7 Web server plug-in behavior and failover -- 5.7.1 Normal operation -- 5.7.2 Failover operation -- 5.7.3 Tuning failover -- Chapter 6. EJB workload management -- 6.1 Enabling EJB workload management -- 6.2 EJB types and workload management -- 6.2.1 Stateless session beans -- 6.2.2 Stateful session beans -- 6.2.3 Entity beans -- 6.3 Naming and name spaces -- 6.3.1 Looking up an EJB home with JNDI examples -- 6.4 How EJBs participate in workload management -- 6.4.1 Initial request -- 6.4.2 Subsequent requests -- 6.4.3 Cluster run state changes -- 6.5 EJB server selection policy -- 6.5.1 Server weighted round robin routing configuration -- 6.5.2 Prefer local configuration -- 6.6 EJB workload management behavior -- 6.6.1 WLM behaviors using server weighted round robin -- 6.6.2 Prefer local -- 6.6.3 Process affinity -- 6.6.4 Transaction affinity -- 6.7 EJB workload management failover -- 6.7.1 Exceptions triggering automatic failover -- 6.7.2 Exceptions thrown by WLM to the application -- 6.8 Backup Cluster support -- Part 3 Implementing the solution -- Chapter 7. Implementing the sample topology -- 7.1 Overview -- 7.1.1 Software products -- 7.1.2 The sample topology -- 7.1.3 Applications used in