

GemStone Facets : Real-Time Performance for Strategic Applications

GemStone Facets Highlights

- ▶ Dramatically improved performance for distributed applications
- ▶ Fast, efficient development of strategic e-business applications
- ▶ Non-stop availability of critical business data

GemStone Facets™ is the industry's first shared transactional Java™ workspace, software that dramatically enhances the performance of complex, distributed e-business applications. Applications deployed on Web servers, J2EE™ application servers, and Java messaging systems use GemStone Facets to achieve real-time performance by separating transaction control from the delay of communication with backend systems. GemStone's patented technology provides thousands of users and multiple distributed applications with a logically consistent, recoverable real-time view of critical business data across islands of information.

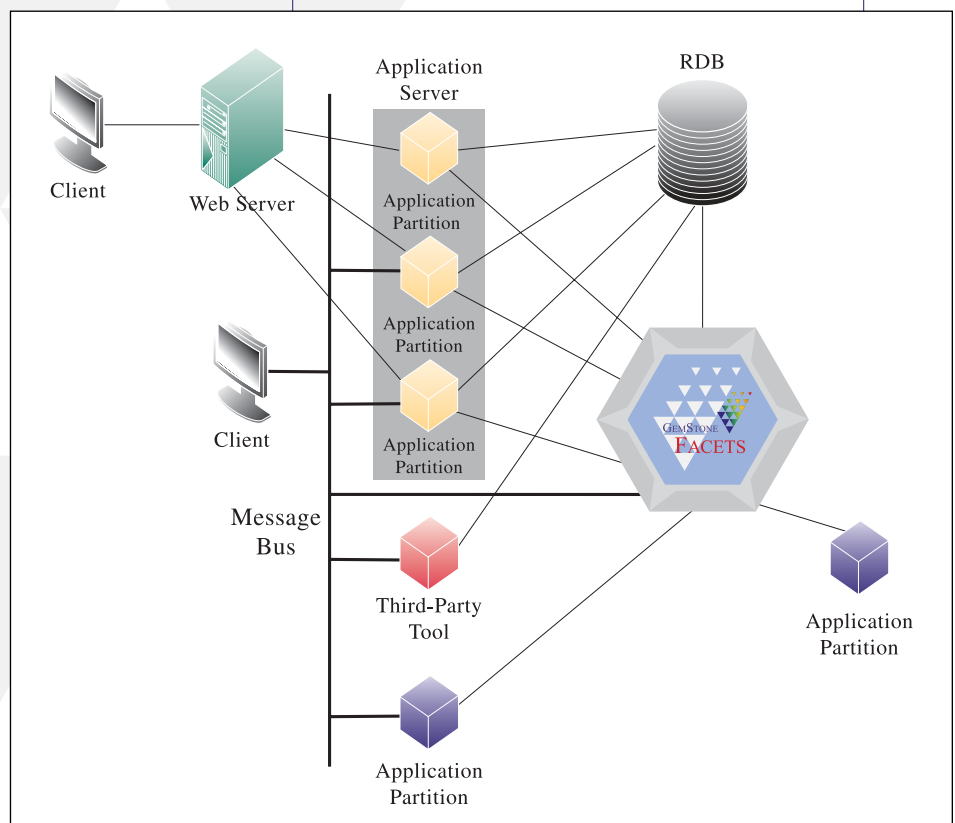
servers, and Java messaging systems. Transparent persistence mechanisms, both native Java and JDO, improve performance by deferring data conversion overhead at run-time, provide recoverability for working data, and speed development by reducing backend mapping for working data and metadata.

Handle 4,000 to 4,000,000 a Day, Flawlessly

GemStone Facets' patented caching technology can support thousands of concurrent user interactions and millions of transactions per day, ensuring superior performance under the highest loads.

GemStone Facets: The Java Booster

GemStone Facets' unique architecture boosts performance by eliminating multi-casting overhead while maintaining a consistent, shared transactional view of data across virtual and physical machines in distributed Java applications. It integrates smoothly with application architectures based on JSPs/servlets, J2EE application



GemStone Facets Integration

Transaction control at the object level minimizes round trips to backend data sources, allowing backend storage to be handled independently. Instead of multi-casting copies of data, user sessions and other processes share access to a single copy through a shared cache on each machine. This minimizes communication overhead, thus improving application performance.

Develop and Maintain Complex Applications with Ease

Integration with back-end sources can be the most costly part of a development project. GemStone Facets relieves that burden, because the shared transactional workspace transparently interfaces with a large, disk-based Java object repository. The repository provides both native object persistence and support for the JDO standard persistence mechanism. Any Java object referenced by a persistent object—whether accessed through EJBs, JSPs, servlets, or pure Java—is automatically made persistent, with no coding effort required. This transparent persistence speeds and lowers the cost of development projects because working data and metadata need never be mapped to backend data stores, and fewer programmers with JDBC skills are needed to develop mapping for the remaining data. Persistence in the shared workspace makes applications more adaptable, as the application can be modified on line.

Choose a Workspace That Holds All the Information You Need

In today's strategic applications, information is often assembled from diverse and far-flung sources: from partners or customers, back-office systems, and legacy stores. Gathering an application's working data from all these sources takes

time and network resources. If the working set size is limited by available memory, repeated round trips to data sources can seriously degrade application performance and network availability.

GemStone Facets overcomes this barrier with an object repository capable of handling billions of objects and hundreds of gigabytes of working data. Using the repository, applications can assemble large working sets from distributed sources and use them as needed, avoiding excess communication with backend systems. GemStone Facets also features scalable collection technology so applications can efficiently handle data sets with millions of entries.

Count on Working Data to Keep Working

Every moment of downtime is money lost. GemStone Facets' fault tolerance features ensure continuous operation of your applications. For high availability at the system level, GemStone Facets can be integrated with hardware and software solutions that provide automatic failover of system resources. In the worst-case scenario, system failure, GemStone Facets ensures the availability of working data, as persisted information is recovered from the Facets repository without repopulating Java objects from backend sources.

GemStone Facets Features

J2EE Integration

GemStone Facets boosts performance of J2EE applications by providing a transactional workspace where massive numbers of Java objects can be populated from widely distributed, heterogeneous sources and shared across physical and virtual machines. Transparent persistence streamlines

development by reducing mapping requirements to backend data stores.

Messaging System Integration

Messaging systems are ideal for communication and coordination between processes. But passing large amounts of data over a messaging bus is not recommended because it can degrade performance as the bus fills, application adapters work overtime converting data to and from the messaging format, and processes are delayed waiting on distributed two-phase transaction control. When applications and processes share working data through the GemStone Facets workspace, these conversion delays are eliminated and the message bus can again be a dedicated communication resource.

Caching

GemStone Facets includes prepackaged caching policies that optimize communication with backend data sources. With asynchronous write-through caching, for example, applications operate from data in the shared workspace while Gemstone Facets updates backend storage asynchronously as resources are available. Aged-out caching flushes unused data from the shared workspace after a time, optimizing access speed and available storage.

Transaction Coordination

GemStone Facets includes patented technology that simplifies application design and improves performance through optimistic concurrency control. Optimistic concurrency control allows critical business data to be updated without locking. These services improve performance because critical resources don't have to be locked, so applications can make full use of system resources. Applications

are easier to design because the underlying system handles the concurrent updates.

Security

GemStone Facets security is compatible with a J2EE container or other Java security mechanisms. For example, a Java servlet's client ID (X.500 name) can be sent from a container or session and established as the security principal for GemStone Facets operations. GemStone's security mechanisms provide both user ID/password security and digital certificate-based authentication and authorization, including access control.

High Availability

GemStone Facets provides out-of-the-box fault tolerance through continuous monitoring and automatic restarting of processes. Critical processes can be replicated for automatic failover. For system-level availability, GemStone Facets can be integrated with high availability hardware and software solutions such as Sun Cluster 3.0 and Integratus , which provide automatic failover of system resources and protection from media corruption and data loss.

Administration Tools

GemStone Facets' GUI-based administration tools include a Performance Monitor and an Administration Console for system configuration and management. A full administration API is available for scripted administration tasks.

GemStone Facets also includes a graphical Visual Statistics Display (VSD) for precision tuning of application performance. Statistics are generated automatically by all system components and then

stored in shared memory for maximum efficiency. They are afterward asynchronously written to the system disk. VSD enables system administrators to correlate events in the statistics stream, identify bottlenecks, and tune accordingly.

Specifications and Requirements

Java Standards Support

- JNDI (Java Naming and Directory Interface) 1.2
- JDO (Java Data Objects)
- JTA (Java Transaction API) 1.0.1*
- Java security standards (JSA, JCE, JCA)
- Integrates with J2EE application servers and Java messaging products via Java Connections Adapter (JCA)
- 64 bit Java Virtual Machine on Solaris

Database Connectivity

- Plug and play with standard JDBC™ drivers for your database of choice
- WebGain TopLink™

Security Standards Support

- Supports all standards compliant Java security products
- Supports all Java 1.4.2 security specifications

Web /Application Server Integration

- BEA® WebLogic 8.1
- Sun J2EE Reference Implementation
- IBM WebSphere® 5.0
- Apache Tomcat 5.0 Web Server
- New Atlanta ServletExec™
- JBoss Application Server 3.2

XML

- Supports the generation and storage of XML documents

Development Tools

- JDK™ 4.2 compliant (supports all core APIs)
- Sun Microsystems NetBeans™ for Java
- Borland JBuilder™
- Eclipse® for Java
- Any 100% Pure Java™ solution

Development/Deployment Platforms

- Solaris™ 2.8 and above Operating Environment software
- Red Hat Enterprise Linux AS 3
- Suse Linux Enterprise 8