



THE VALUE OF GEMFIRE ENTERPRISE

- Speed up overall application performance
- Increase overall system scalability (without additional hardware)
- Reduce Network traffic
- Store data closer to the point of use
- Coordinate activities for multiple processes in near real-time.
- Even across multiple computers. Even across multiple computer languages. Avoid the bottlenecks inherent in using databases for cross-application data sharing.

GemFire Enterprise Technology Use Cases

GemFire Enterprise, a component of the GemFire Enterprise Data Fabric (EDF) is a flexible, high performance, cross-platform, cost-effective solution for data distribution, distributed caching and management across the enterprise in real-time. It makes data available on demand to your application regardless of underlying data sources or formats. GemFire Enterprise is built on the industry's fastest and most reliable data distribution and caching system. It connects databases, analytic tools, application stores, messaging systems, EII, other middleware, and mainframes. GemFire increases information effectiveness by dynamically provisioning consistent data at the right place, at the right time, and in the right format in memory.

GemFire Enterprise, a component of the GemFire Enterprise Data Fabric (EDF) is a flexible, high performance, cross-platform, cost-effective solution for data distribution, distributed caching and management across the enterprise in real-time. It makes data available on demand to your application regardless of underlying data sources or formats. GemFire Enterprise is built on the industry's fastest and most reliable data distribution and caching system. It connects databases, analytic tools, application stores, messaging systems, EII, other middleware, and mainframes. GemFire increases information effectiveness by dynamically provisioning consistent data at the right place, at the right time, and in the right format in memory.

For efforts such as C4 and Intelligence community SOA and collaboration initiatives, the goal is to make information available to anyone, anytime, anywhere across organizational boundaries. This includes both a source and destination focus simultaneously. Business

specific events cause many people to want to see the same information at the same time in massively distributed environments. This ubiquitous access to information puts considerable strain on existing IT and operational infrastructure. As computing architectures allow more access to the entire community, loads on these systems become unpredictable especially when new events occur and as a result, significant excess capacity needs to be built into these systems to handle the peak demands. This directly translates into substantial costs; larger database servers, more network bandwidth, more hardware, more administrators, etc.

To allow systems to meet the challenges of these issues what is needed is an "Information Utility" or "pervasive data fabric" that supports distributed and real-time decision making that can be delivered with low insertion and low cost impact. This utility should provide data access that is "on tap, on demand" across the enterprise and is decoupled from



the applications or infrastructures that require it. It should increase overall application performance, reliability and scalability. GemFire Enterprise is a language and infrastructure neutral data management, distribution and caching software that has enabled GemStone System's customers and partners to achieve these goals.

GemFire embodies the lessons learned from 20 years of development in object data base, virtual machine, application server technologies and in the use of shared memory for optimizing data management in distributed systems. GemFire was created to focus on a gap in data management, which is the need for real-time in memory data management and data distribution that is language and infrastructure neutral and can address the need to share data across existing process, software and hardware boundaries.

COMMON USE CASES FOR GEMFIRE ENTERPRISE

1) Overcome limits in current Service Oriented Architectures (SOAs).

Since SOAs rely on a shared services model, data movement across distributed services is of paramount importance. Fetching the same data multiple times creates significant performance problems in SOA applications. Whether it is checking user credentials, fetching reference data, or retrieving session state across services on different computers, data access can be a major bottleneck in SOA applications. SOAs break single applications into independent processes (which may run on different computers).

Before SOAs, applications only performed authentication once, at the beginning of the process. Now they must check credentials with each step. Before SOAs, applications shared data across steps by holding onto data in the process memory. Now the application has been decom-

posed into several processes and data that another step needs must be persisted in some shared space (usually a DBMS). If SOA applications depend on communicating to other applications to retrieve data, this can create strains on the back-end applications, as they may not have been built to handle the loads that a SOA may impose on that system.

GemFire Enterprise supports a nimble distribution model that can move data on demand between services, especially when there is shared information between the services. In these kinds of scenarios, data that is stored in the distributed GemFire Enterprise layer is instantaneously available to any application that needs it. The result is an application that can run tens or hundreds of times faster than if it had to share data using an RDBMS and can scale across multiple computers.

2) Enable Cross-Application Collaboration.

GemFire Enterprise enables distributed in memory, cross application, data exchange via C, JAVA or XML. Data can be cached by one process and retrieved by others. This data exchange can cross processes, applications, computers, and operating systems. Through a unique set of features it enables systems to control how distribution and synchronization of data occurs. GemFire can replace existing mechanisms that use databases, file systems, or other mechanisms to share information. Performance can be increased by a factor of ten or more in these cases.

3) Avoid redundant data access and enhance application performance.

This reduces data access times significantly for redundant data regardless of where the request originates, thus significantly reducing access time and system load which is critical when events increase the demand for such information. GemFire Enterprise enhances the performance

of portals, applications, application servers, and systems by storing data that is accessed multiple times by programs, groups of users or by a single user who is massaging the data. The areas of enhancement include reduced computing time for the applications, reduced network traffic between applications and other data sources, and reduced compute time for infrastructure applications that feed the system (parsers, EAI software, databases, etc).

Examples of this include:

- **Trusted Guards** can reduce traffic across them by caching data that has already crossed the guard, thus minimizing security risks and reducing load on manual information security review.
- **Parsed messages** can be stored in GemFire in the structure used by multiple applications thus being parsed and convert to a program structure only one time (instead of multiple times).
- **Results returned from a Web Services invocation** can be cached to reduce the time it takes to compute them. The data distribution mechanisms of GemFire Enterprise provides a means to make it easier to cluster machines that respond to Web Service requests.
- **Information on portals that is dynamically generated** from back end data sources (databases, etc).
- This provides several benefits, faster data access, reduced application computation (especially of mapping between database formats and object or XML documents), off-loading computation from the database, and reduced network traffic.

3) Reduce network load.

In a WAN environment the use of a distributed cache can significantly reduce the amount of traffic across the network. For data that represents redundant data access by one or more users, it is transmitted across a WAN as each user needs to access it. With GemFire Enterprise, once data has been transmitted across the WAN for one user, it

is available for all users. The reduction in bandwidth usage can be significant in low bandwidth environments both from an application performance viewpoint and from an overall view to maximizing the amount of data/bandwidth available.

5) Enhance survivability of distributed applications.

Due to the distributed nature of the cache and the ability to replicate data objects on more than one computer, the cache can enhance the survivability of applications by enabling them to have access to some data even if the network disappears. The application enables applications to continue to work with data that was retrieved before the network outage. It is often better to have some data rather than none.

Overall, GemFire Enterprise can enable organizations to combine many of these benefits. The ability of GemFire to provide data to an application server from a database enables applications to be forward deployed without requiring a local database or continuous communications to a database. This provides the combined benefits of performance, reduced network load, enhanced survivability, and reduced system administration overhead.

6) Turbo-charge Web Services based applications.

GemFire Enterprise significantly enhances XML performance in a web services environment. GemFire Enterprise supports XML in several key ways. It provides for storage of XML documents in cache in an optimized DOM to minimize data access time and storage space in memory. It provides for access to the XML storage for reads, writes, searches, etc through a variety of mechanisms. Access to XML collections can be enhanced through the use of multiple index types on XML document collections to optimize search performance. Together with GemFire's optimized DOM model this can lead to significant speed up in XML performance for applications. GemFire also enables flexible data manipulation in the web service

access layer through the use of Interceptors and Interceptor Chains to allow for operations on incoming/outgoing XML. The operations could include data transformation, conversion from XML to Java, or other user defined operations. Basic tools to handle both data transformation and XML to Java conversion are provided with GemFire. GemFire provides numerous ways in which to access XML data, including a web services interface, a Java interface with the XML:DB API's, Xpath access, JAX-RPC access, and other standard XML interfaces.

7) Boost performance and scalability for Java and J2EE applications.

GemFire enhances the J2EE environment. This is accomplished several ways.

GemFire can help applications deal with the challenges of Object-Relational mapping performance. By creating an object layer that is synchronized with the backend system, application level object access can be directly to objects in memory (or on disk) so that object references are never more than one I/O and hundreds of times faster than a direct call to the database. Additionally, new cross object queries are possible with GemFire's support for OQL.

GemFire enables JAVA programs to access additional memory that is represented by objects in the cache. GemFire can help reduce the size of the Heap of a JVM and thus speed-up the JVM significantly since Java Garbage Collection is expensive. GemFire can also be used to implement a high performance means to cluster application servers. A cache may be used to distribute session state across the cluster for failover or load balancing. GemFire's session state cache implements fine-grained distribution of session state, allowing web applications to cache more state with less cost. Each session's attributes are managed transparently by GemFire and are replicated to one or more other nodes in the cluster, or to a separate backup server. GemFire selects appropriate backups and han-

dles load balancing and failover with no more than a few web.xml annotations. HTTP results and other objects can, of course, also be cached in GemFire.

GemFire conforms to many standards including JTA, JCA, JMS, JCache SSL, SOAP, OQL, XML:DB, Xpath, and other XML standards.

8) Enable multiple computers to seamlessly share memory.

GemFire can create a data-space that spans multiple computers memories. This can enable applications on 32 bit computers that would normally be limited to 4 GB of RAM to access larger memory spaces by combining multiple computers and their memories into a single cache.

This capability can be used to allow GemFire caches on 32 bit computers to share memory with 64 bit computers that are virtually unlimited in the amount of memory they can address. GemFire manages the distribution and sharing of data across the computers, so that the data in 64 bit memory will be transferred to the 32 bit memory space on-demand. This can provide for a large cache that can be shared across multiple computers. For example, a set of smaller application servers that needed access to terabytes of cached images, could easily accomplish this by sharing a large GemFire cache on a 64 bit server.

This capability can also be used to allow computers in a GRID or blade server environment to appear to have a single large memory space, instead of having a group of disjointed individual memory spaces.

What if you could

- Increase application performance
- Scale to more concurrent users with less hardware
- Reduce network traffic
- Reduce load on back-end systems (data bases, application servers, message buses etc)


- Reduce overall hardware requirements
- Handle larger chunks of data across the enterprise at mission critical speeds
- Reduce access time and system load when multiple users access the same data
- Provision critical information closer to users regardless of location or proximity to data

KEY FEATURES OF GEMFIRE

- Standards based (support for JCache, J2EE, XML, Xpath, Web Services, OQL)
- Co-exists with and complements existing infrastructure
- Cache support for Java, C, and XML objects
- Cache can be distributed across multiple computers
- Data distribution controls (scope, confirmations, locality of data)
- Cache loader provides automatic loading of cache objects from one or more backend data sources
- Event listeners enable programs to respond to changes in data from other programs/threads.
- Concurrency controls across multiple instances of the cache and multiple concurrent processes
- Sophisticated cross machine garbage collection.
- Overflow to disk to avoid out-of-memory errors
- Control over what is in memory and how items are evicted
- Mirroring for fault tolerance
- Sophisticated monitoring and statistics collection to help optimize the system and monitor/replay system behavior
- Graphical console for monitoring and controlling system



Data Caching
Positions the right data close to your applications and increase performance and availability



Data Distribution
Enables scalable deployment across multiple application nodes



Data Virtualization
Provides data location transparency and de-couples data access logic from applications



High Availability
Increases resilience of application to data source unavailability, network issues, etc. and ensures business continuity



Data Integration
Offers an operational data store for collaboration across multiple applications, using data in different formats



Continuous Event-Processing
Derive new business opportunities by instantaneously analyzing real-time events

Corporate Headquarters:

1260 NW Waterhouse Ave., Suite 200 Beaverton, OR 97006 | Phone: 503.533.3000 | Fax: 503.629.8556 | info@gemstone.com | www.gemstone.com

Regional Sales Offices:

New York | 90 Park Avenue 17th Floor New York, NY 10016 | Phone: 212.786.7328
Washington D.C. | 3 Bethesda Metro Center Suite 778 Bethesda, MD 20814 | Phone: 301.664.8494
Santa Clara | 2880 Lakeside Drive Suite 331 Santa Clara, CA 95054 | Phone: 408.496.0242