



Easily Scale Your Web Applications

With Terracotta and Sun™ platforms

Every day, more applications are created or ported to run on the Web or be hosted in a cloud. It's always difficult to predict when or if an application will take off, which means you need the ability to quickly scale up and down with demand. Web-based applications — serving industries such as travel and reservations, e-commerce, financial services, media, healthcare, IT and service providers, research, education, government, and on-line gaming — must also be reliable to ensure user satisfaction and loyalty. This requires reliable application servers to help ensure that user state is not lost if a server goes down. For example, if a user is adding items to a shopping cart and the application server goes down, the items in the cart need to persist across the failure to help ensure a high quality user experience and maintain customer loyalty.

Most developers write Web applications on their laptops or on one server and think about how to scale the applications and high availability in phase two. In the native Java™ platform, developers need to write the code to scale across multiple servers, which can be complex when dealing with failover scenarios. Java Platform, Enterprise Edition (Java EE platform) can be used, but though powerful, it takes time to learn, which can be detrimental when time to market is of the essence. Some developers turn to other tools such as Spring and Hibernate, but these tools don't have as many features for scalability and reliability as the Java EE platform.

In addition to scalability and reliability, you also need to reduce server count wherever possible to cut operational costs, while still

handling changes in demand. The problem with adding more application servers is that the application performance doesn't always scale because utilization on each server drops due to network and database bottlenecks.

The open-source Terracotta platform running on powerful Sun servers and the Solaris™ 10 Operating System (OS) can help developers deploy scalable Web applications that are reliable, fast, and cost-effective to manage.

Terracotta

Terracotta is an open-source product, entirely written in Java code, that simplifies scalability for Java applications by synchronizing user-defined subsets of Java heap across multiple Virtual Machines for the Java platform (Java Virtual Machine or JVM™) on different physical machines. This enables threads on machines across a cluster to communicate with each other as though they were all in the same local machine, making it easier to cluster and scale a wide variety of Java applications (both new and existing) with little or no code changes, and with dramatic performance benefits in most cases.

Open-source Terracotta is available in three versions:

- ES — open source packaged version of the core project
- EX — ES features with a commercial license and support
- FX — EX with extended features (array striping, operations center, rolling upgrades of your applications) and value-add services)

Highlights

- Deployable in 30 minutes
- Easily develop scalable applications and write less code
- Enables developers to concentrate on logic, rather than infrastructure
- Easily scale performance up, down, and out
- Increase availability and reliability of Java applications
- Incredible performance on Sun™ servers and the Solaris™ Operating System (OS)

Benefits for developers

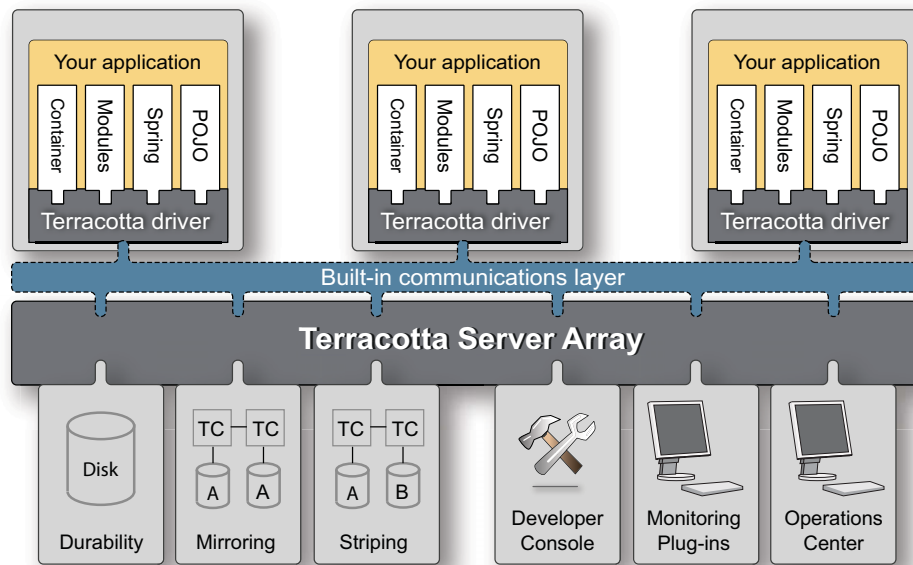
With Terracotta, developers write less code because Terracotta takes care of clustering the application across multiple machines, without interfering with the application code. In a standard Java environment, the developer must deal with the underlying messaging constructs to synchronize the state of a distributed application back to a central database. This clogs both the network and the database and is only exacerbated when more machines are added to the cluster. Terracotta eliminates both the coding process and the communication to the database to update state by transmitting only object state changes to the Terracotta server.

Terracotta supports all major Java platform data structures, supports Oracle Weblogic 10.2, 10.3, and Hyperic HQ, deploys in less than 30 minutes, and includes many powerful features and tools for developers.

- Clustered, reliable Java memory — Terracotta makes object data in your application durable and available across JVMs by transparently sending changes to the Terracotta Server Array where it is persisted to disk. Any change made on one node is automatically persisted to disk asynchronously. If one of the nodes crashes, it recovers from the Terracotta Server Array's persistent store.
- Virtual memory management — Clustered application data is automatically moved to application server JVMs where it is needed, just in time for when it is needed. Data that is used less frequently is stored in the Terracotta Server Array, but removed from live application memory to make room for fast access to the most frequently accessed objects in memory.
- Distributed lock management, often called a *hot set* — Threads on different JVMs can participate in Java locking and thread coordination, either using Java concurrency primitives or the higher-level Java concurrency libraries. Terracotta

coordinates cross JVM activities using just pure Java code, making JVM communication and coordination as simple as possible. This means that access to data occurs in a local heap most of the time for faster performance.

- Developer tools — The Terracotta Developer Console provides a wealth of information about the application cluster for easy debugging and tuning. The console allows developers to see how objects are distributed across the cluster and the contents of these objects. Developers can control which objects are managed by Terracotta through annotations, IDE plug-ins, or a simple configuration file.
- Cloud and grid APIs — Data routing and data presence APIs simplify deployment and boost cloud and grid performance by enabling queries to the Terracotta runtime and optimizing routing decisions based on locality of data.
- Integration — Terracotta works out of the box with the most popular technologies for enterprise applications including the open-source Glashfish™ application server, Spring, Hibernate, Lucene, Quartz and more. Terracotta also works with most Java application servers.



High performance

In a Terracotta environment, the application appears to be reading from local memory, but the data might actually be in local memory or the Terracotta server. The location is optimized to maximize performance.

In general, an object already referenced by a given application server has the object in local heap already. This can significantly decrease latency, thus increasing performance as well as scalability. In addition, all data is in object form, rather than stored as relational data. If a change occurs, only the changed bytes are moved around the network, not the whole object, which relieves the network and the database from performance robbing traffic.

Terracotta architecture seamlessly scales applications across servers and enables application fault tolerance.

In addition, server striping, available in Terracotta FX, enables you to easily increase the capacity and throughput of enterprise Java applications by partitioning data in the Terracotta Server Array across an arbitrary number of physical machines. Adding server array striping requires nothing more than a simple configuration change. No application changes are required.

High availability

Along with scalability, Terracotta also makes applications more highly available and reliable.

- Terracotta keeps critical application data persistent and coherent between JVMs vastly simplifying enterprise application high availability concerns.
- Like RAID for Java memory, the Terracotta Server Array can be mirrored for reliability.
- The Terracotta Operations Center provides a dashboard for systems diagnostics, backup, systems management, and a cluster-wide log viewer. Capabilities previously only used for databases are now available for data managed within Terracotta.
- The Terracotta server can have any number of standby servers that take over if a server instance goes down. All JVMs automatically connect to the new instance and continue working.
- The Terracotta Cluster Membership APIs provide information on nodes joining the cluster, nodes leaving the cluster, and when cluster operations are disabled (e.g., during a failover event) and re-enabled.

Run your Terracotta cluster on Sun servers

It just makes sense to run highly-threaded distributed Java platform-based applications on Sun servers and the Solaris 10 OS. Both the hardware and the OS are designed from the start with goal of running these types of applications with exceptional price/performance, high availability, and eco-responsibility.



Terracotta Operations Center.

To illustrate the inherent suitability of Sun servers, storage, and the Solaris OS for Terracotta clustered applications, Sun recently performed tests comparing the Solaris 10 OS and Linux on Sun Fire™ X4200 servers and Sun SPARC Enterprise® T5240 servers. The results, listed in the table below, are quite impressive.

Scale-out with x64 servers

The test results showed nearly double the performance with Solaris 10 on the Sun Fire X4200 server compared to Linux on the same server. In addition, the Sun Fire server's CPU was more highly utilized, providing better ROI for distributed Java platform-based applications. Sun's versatile new Flash-ready x64 servers deliver extreme efficiency,

intelligent scale, reliability, and performance. Optimized for the Solaris OS, the Intel® Core Microarchitecture, and AMD Opteron™, Sun's x64 portfolio also includes Integrated Lights Out Management (iLOM). iLOM handles server management and integrates easily with Sun™ xVM Ops Center and third-party monitoring tools, so you can easily manage and provision your distributed application infrastructure.

The Sun Fire X4240 server, powered by the six-core AMD Opteron processor, is the latest generation of the Sun Fire X4200 series, offering up to twice the memory and storage capacity of any system in its class.

System	OS	Total TPS	Threads	%CPU
Sun Fire X4200 server	Solaris 10 OS	8264.54	96	95%
Sun Fire X4200 server	Red Hat Enterprise Linux	4795.03	96	45-50%
Sun SPARC Enterprise T5240 server	Solaris 10 OS	7229.93	192	9%

4 Easily Scale Your Web Applications



It's the first and only two-socket AMD Opteron system with 16 hard drive slots in a 2 RU form factor, providing up to 12 cores and 128 GB of memory. With the Sun Fire X4240 server, you can start by specifying just enough processing power as well as memory, storage, and I/O for your current needs. But when fast growth creates high demand, you can increase capacity without removing the chassis. No changes to the OS, software, rack, network infrastructure, or storage connections are needed.

Innovative power-management — provided by AMD HyperTransport 3, HyperTransport assist, and more power-efficient components — result in overall energy efficiency. The servers remain in idle states longer, have better granularity, and are more responsive to changes in demand, reducing energy costs.

Scale-up with Sun SPARC Enterprise® servers
The results on the Sun SPARC Enterprise T5240 server were outstanding and utilized only 9% of the CPU capacity. Using no-cost Solaris Containers, you can run multiple instances of Terracotta on this system, providing near instant, on-demand, scale-up scalability.

Powered by the UltraSPARC® T2 Plus processor, the Sun SPARC Enterprise T5240 server combines the scalability and reliability of traditional enterprise systems with breakthrough performance, eco-efficiency, and cost effectiveness in a 2 RU form factor that redefines midrange computing. The UltraSPARC T2 Plus processor is a *System on a Chip* for multisolet systems. It integrates 8 cores, 64 threads, 8 floating point units, cryptographic acceleration, and PCIe I/O directly onto each processor. High speed processor interconnects help ensure that your multithreaded Web applications scale, enabling you to maximize throughput and ROI, while saving money and the planet!

UltraSPARC T2 and T2 Plus based servers are the first to incorporate Sun's unique power management features at both core and memory levels of the processor. They actively reduce instruction issue rates, park idle threads, disable cores, and turn off clocks in cores and memory to reduce power consumption by up to 35%, so when demand is slow, you're not paying as much for idle resources.

The Solaris™ OS — making the difference
Once again, the Solaris OS outperforms the competition. Terracotta automatically takes advantage of the turbocharged TCP/IP stack, a radically improved kernel, and special optimizations for memory allocation and chip multithreading — achieving faster performance without requiring changes to existing applications. Solaris Containers enable you to run multiple Terracotta instances on one server for better utilization and throughput, resulting in fewer servers to manage.

While Terracotta makes your data fault tolerant, the Solaris OS helps make the application and system more reliable and available. Solaris Predictive Self Healing works transparently in the background to make sure your business-critical applications and essential system services continue to run uninterrupted in the event of software and hardware failures. Additional Solaris features also help drive near-zero downtime.

The OpenSolaris™ community is where the next generation of the Solaris OS is under development. The OpenSolaris 2009.06 operating system delivers industry-leading features from Sun and the OpenSolaris community in networking, storage, and virtualization, along with significant performance enhancements and developer productivity updates. The new release includes Project Crossbow, the most

Learn More

sun.com/third-party/global/opensource.jsp
www.terracotta.org
Storage Simulator: sun.com/storage/disk_systems/unified_storage/resource.jsp

significant networking technology development to Solaris in this decade. This preview of the next generation of Solaris demonstrates Sun has the leading platform designed for the latest hardware technologies that power scalable and secure multithreaded applications in a virtualized and networked world.

More performance with Flash technology
Sun's Flash solid state disks (SSDs) provide the speed, scalability, and efficiency you need to improve application response times by up to 70x. Hot-swappable SSDs dramatically improve performance without disrupting your applications, and are 70x more cost-efficient than traditional HDD deployments. In fact, SSDs consume 38% less power per GB and 98% less power per I/O compared to spinning disks, making SSDs the most cost-effective way to boost server performance.

Simple scalability
With Terracotta running on Sun systems and the Solaris OS, it's simple to scale-up, scale-out, or scale-down your Java platform-based Web and cloud applications and take advantage of open-source innovation with commercial support. Terracotta makes in-memory session state and Web workflow durable, highly available, and faster with minimal developer effort. And with Sun platforms running these distributed applications, you know you're getting the benefit of extremely high performance, cost-effectiveness, reliability, manageability, and the capacity and tools to scale your applications on-demand.

