TERRACOTTA

Terracotta Technical Whitepaper: Enterprise Ehcache

Abstract

Application scale requirements range from single-machine installations to very large, multi-datacenter and cloud deployments. As application usage grows, and more users send larger and larger waves of transactions through, the architects and operators of that application must find ways to increase capacity to meet that escalating demand. Achieving high-performance scalability for enterprise applications, however, is a costly problem and a complex challenge. Typical approaches require development-intensive application redesign, expensive database licenses and high-end hardware.

Terracotta's Enterprise Ehcache is an easy-to-deploy, software-only solution for hard-to-solve performance and scalability problems at any point along the scale continuum.

Contents

Terracotta, Inc.
575 Florida St. Suite 100
San Francisco, CA 94110

Contact us to learn more about Terracotta products, services and support.
Please visit www.terracottatech.com, e-mail us at info@terracottatech.com or
contact one of our sales professionals at +1 (415) 738-4000.

Enterprise Ehcache Whitepaper v2
December 23, 2010

Overview

Application scale requirements range from single-machine installations to very large, multi-datacenter and cloud deployments. As application usage grows, and more users send larger and larger waves of transactions through, the architects and operators of that application must find ways to increase capacity to meet that escalating demand. Achieving high-performance scalability for enterprise applications, however, is a costly problem and a complex challenge. Typical approaches require development-intensive application redesign, expensive database licenses and high-end hardware.

Terracotta's Enterprise Ehcache is an easy-to-deploy, software-only solution for hard-to-solve performance and scalability problems at any point along the scale continuum.

This whitepaper will discuss how the unique architecture of Enterprise Ehcache delivers unparalleled benefits to enterprise applications including:

- Immediate performance gains
- Efficient access to large data sets
- Maximum performance at every scale
- Linear scale-out
- High availability

Snap in Performance

At one end of the scalability continuum—i.e., applications that run on a single machine—adding capacity means maximizing raw performance. Caching is usually the easiest and most effective way to reduce latency and increase throughput. A cache stores results that are relatively expensive or time-consuming to retrieve or compute so that subsequent work that relies on those results may complete without incurring the cost of repeated operations. At a stroke, adding effective caching can improve application performance by orders of magnitude with minimal code impact.

Ehcache is the de facto caching standard for enterprise Java, with over 250,000 enterprise deployments, including 70% of Sun's Java customers, and the majority of the Global 2000. It ships as the default cache of many popular applications, containers and frameworks including Atlassian, ColdFusion, Grails, Hibernate, Liferay, Salesforce and Spring, among others. Using Ehcache is generally as simple as a change to two lines of configuration, to turn on caching in your application. If it isn't already in your application, you can download the library, snap it in and configure it in a matter of hours to remove performance bottlenecks and improve application response times.

In recent tests performed by a customer, Ehcache delivered 90% database load reduction and orders-of-magnitude lower latency at completely linear throughput from 10 threads to 100 in a single application server. At 10 worker threads, the application performed one million transactions/second. At 50 threads, the application performed six million transactions/second. This linear scale illustrates the internal efficiency and inherent parallelism of the Ehcache library

Scale Up with BigMemory

For applications that cache large amounts of data, traditional in-memory caching sets off a time bomb in the Java Virtual Machine in the form of long garbage collection pauses. Storing more data in cache requires a larger Java heap. As Java's heap grows, so do the demands on Java's garbage collector.

The unpredictable nature of garbage collection makes it especially hard to manage; it is impossible to predict when garbage collection will occur and how long it will last. Our work with many enterprise customers has shown that garbage collection is manageable up to an occupied heap in the range of 2-4GB. In fact, even Oracle suggests a 4-8GB heap maximum to avoid long pauses.

To solve the problem of long garbage collection pauses at larger heap sizes, Terracotta offers a plug-in to Enterprise Ehcache called BigMemory. BigMemory uses Java's direct buffer API and an innovative, high performance memory manager to store cache data in memory but off the Java heap where it is invisible to the garbage collector. By keeping large caches off the heap, BigMemory eliminates the need to tune Java's garbage collector and its associated performance bottlenecks. The result is an easy-to-deploy, 100% Java software-only solution for all data-related performance and scalability problems that can easily be used with your data-intensive application today. BigMemory gives Java applications instant access to a large memory footprint in a single JVM, but without the garbage collection cost.

BigMemory has proven to deliver consistent performance for caches ranging from megabytes to hundreds of gigabytes. For applications that run in a single JVM, BigMemory snaps into Enterprise Ehcache with a simple configuration change, giving that JVM instant access to all the memory on the machine without the performance bottlenecks of garbage collection.

For applications using the Terracotta Server Array as a distributed cache, BigMemory maximizes the memory available to each node in the server array (see the next section for an introduction to the Terracotta Server Array). With more memory at thedisposal of each Terracotta server node, a terabyte scale distributed cache is delivered with a fraction of the number of nodes.

BigMemory enables the following for your data-centric applications:

- Maximize the memory usage of your application
- Reduce or eliminate garbage collection pauses without costly tuning effort
- Predictably meet your service-level agreements (SLAs) in terms of maximum latency and throughput even with very large data cache sizes

- Simplify your deployment by consolidating the number of JVMs needed to run your application and/or the Terracotta Server Array

A detailed discussion of the architecture and technical benefits of BigMemory for Enterprise Ehcache may be found in the BigMemory whitepaper on Terracotta.org.

Scale Out with the Terracotta Server Array

While Enterprise Ehcache provides a simple and powerful way to maximize application performance on a single machine, it also delivers seamless migration to a powerful distributed caching architecture when the time comes to scale out.

Any application using the Ehcache library can be configured to use the Terracotta Server Array as a snap-in distributed cache. With two lines of configuration, the performance gains achieved by Ehcache in a single machine can be multiplied by adding more application servers as needed for massive headroom to meet load spikes and future demand.

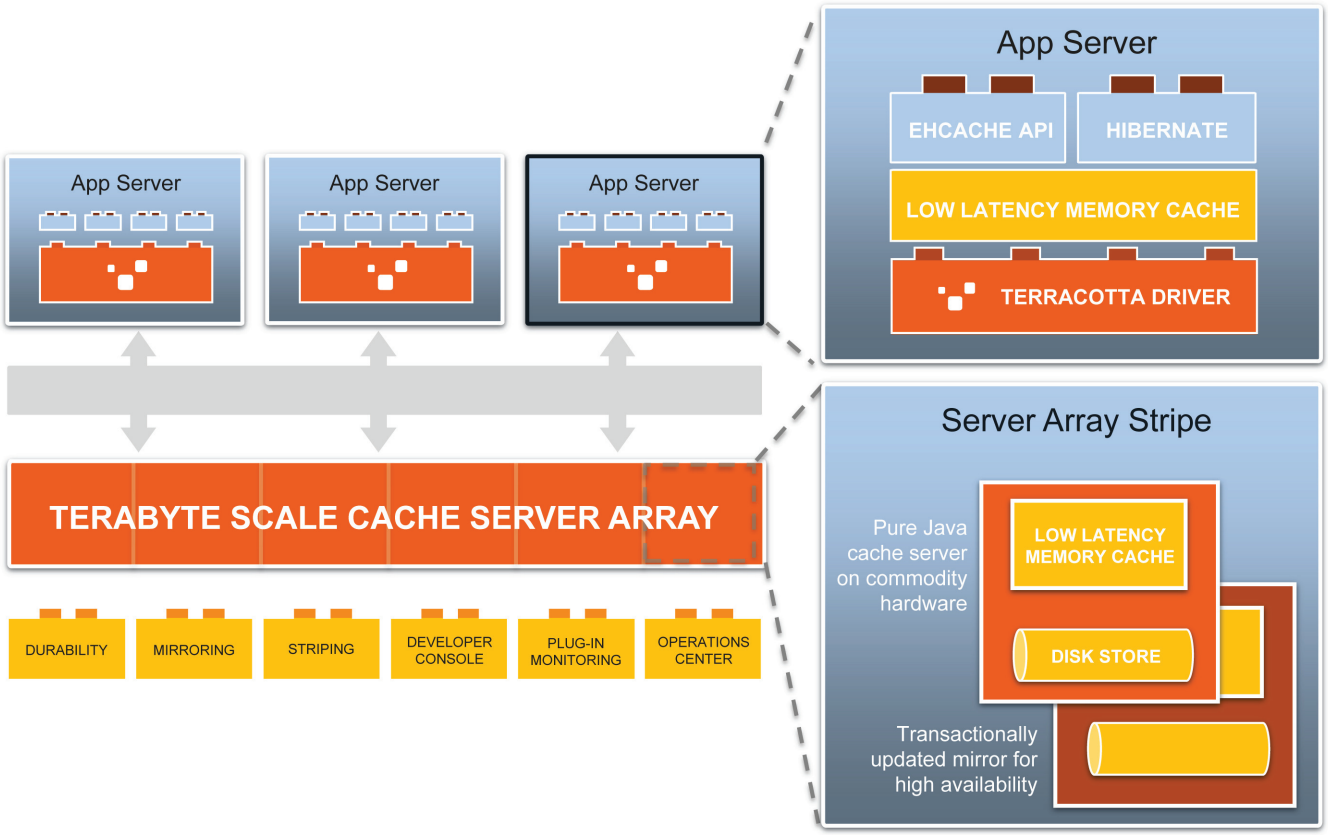
Consider the following sample Ehcache configuration:

```
<ehcache>
  <terracottaConfig url="someserver:9510"/>
  <defaultCache
            maxElementsInMemory="10000"
            eternal="false"
            timeToLiveSeconds="120"
            />
  <cache name="com.company.domain.Pets"
           maxElementsInMemory="10000"
           timeToLiveSeconds="3000">
    <terracotta clustered="true" coherent="true"/>
  </cache>
</ehcache>
```

The addition of the “terracottaConfig” and “terracotta” elements are the only changes needed to enable distributed caching in Ehcache. The “terracottConfig” element directs Ehcache to the location of the Terracotta server array. The “terracotta” element in the “cache” stanza directs Ehcache to make that cache an instance of a distributed cache.

Architecture

The power of Enterprise Ehcache to deliver enterprise-grade performance and scale across the entire scale continuum comes from its unique architecture. This architecture consists of Ehcache in memory in the application tier connected via a standard TCP communications layer to a scalable array of cache servers running on commodity hardware.



The application tier has multiple application servers to distribute application workload, and more can be added on demand to handle greater loads. Ehcache is designed to work with all common application servers and containers – such as Apache Tomcat, IBM WebSphere. JBoss and Oracle WebLogic. Ehcache is also commonly used within standalone Java server applications. (We use the term "application server" interchangeably here for servers and server applications). Each application server has an in-memory cache behind the Ehcache interface that responds to cache lookups in microseconds.

Lookups for cache entries not present in the memory cache are automatically sent through the TCP communications layer to the Terracotta Server Array. The server array responds to cache lookups in milliseconds. All writes to the cache in the application layer are sent to the server array which coordinates acknowledging the write, persisting it to disk, and making the cache update available with configurable consistency guarantees as needed to the rest of the servers in the application tier.

The Terracotta Server Array is an independently scalable set of cache servers that run on commodity hardware. This array delivers enterprise-grade data management to Ehcache in the application tier. Each cache server has an in-memory cache and a disk-backed permanent store. Similar to RAID, the array is configured into groups of servers to form mirrored stripes. The data in the distributed cache is partitioned across the existing stripes. More stripes can be added on-demand to increase the total addressable cache size and I/O throughput. For high availability, each stripe is transactionally mirrored. Should a server node in a stripe be restarted or fail, one of the mirrors will automatically take its place, ensuring maximum uptime and data reliability.

We've seen that the unique architecture of the Terracotta Server Array provides high-performance access to very large data sets, linear scale, a spectrum of data consistency guarantees, and greater reliability than a typical RDBMS. Next we'll take it one click down, describing in detail how the architecture of Enterprise Ehcache delivers each of these benefits.

Large Data Sets: More Data on Less Hardware

The tiered combination of configurable in-memory caches backed by durable on-disk storage allows high-performance access to very large caches without requiring hundreds of servers to fit all of the cache in memory.

In the application layer, the in-process Ehcache cache in each application server uses a configurable amount of memory to provide very low latency access to the most used cache data. What doesn't fit in memory is automatically retrieved from the Terracotta Server Array as needed.

Likewise, the Terracotta Server Array has a configurable disk-backed memory cache. The memory cache and the number of stripes in the Terracotta Server Array can be sized to fit as much data in memory as required. This flexibility allows terabyte scale caches to fit in manageable and cost-effective server arrays of two to a dozen commodity servers.

Applications can retrieve any cache entry from the same Ehcache interface, regardless of whether that entry is in local memory or in the Terracotta Server Array. If the cache entry is stored in memory on the application server, the cache read will return in microseconds. If the cache entry is not in local memory, Enterprise Ehcache will automatically retrieve it from the Terracotta Server Array in milliseconds.

Because all of the cache data is stored on disk, any part of the distributed cache—from application servers in the application layer to nodes in the Terracotta Server Array—can go offline at any time with no application downtime and no loss of data. Enterprise Ehcache is the only distributed cache that offers access to terabyte caches at this level of availability and operational flexibility. Even if your application doesn't use terabytes of data, the massive headroom available will ensure maximum cache performance and reliability on the smallest hardware footprint possible.

Applications can retrieve any cache entry from the same Ehcache interface, regardless of whether that entry is in local memory or in the Terracotta Server Array. If the cache entry is stored in memory on the application server, the cache read will return in microseconds. If the cache entry is not in local memory, Enterprise Ehcache will automatically retrieve it from the Terracotta Server Array in milliseconds.

Because all of the cache data is stored on disk, any part of the distributed cache—from application servers in the application layer to nodes in the Terracotta Server Array—can go offline at any time with no application downtime and no loss of data. Enterprise Ehcache is the only distributed cache that offers access to terabyte caches at this level of availability and operational flexibility. Even if your application doesn't use terabytes of data, the massive headroom available will ensure maximum cache performance and reliability on the smallest hardware footprint possible.

Performance: A Balance of Throughput, Latency, and Consistency

To maximize the performance of an enterprise application requires balancing three related aspects according to their

relative impact on meeting the requirements of the application:

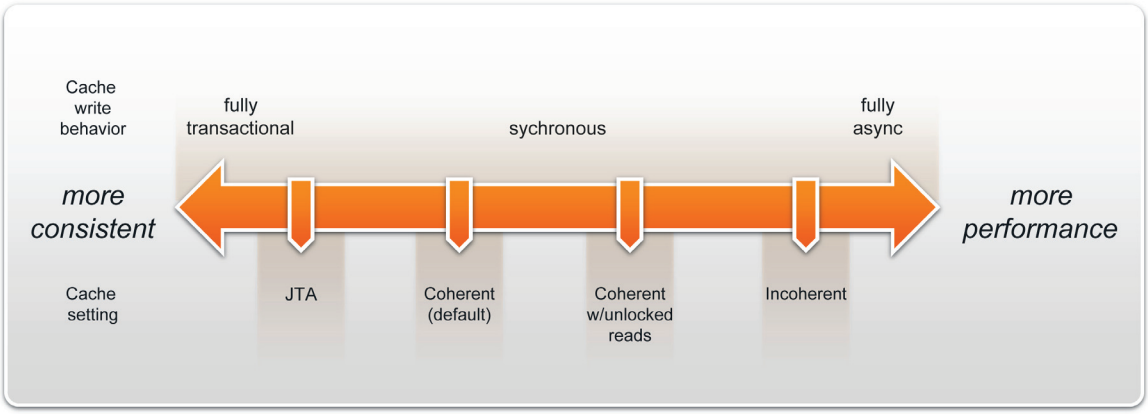
- throughput—commonly measured as the rate of transactions the application is capable of handling;
- latency—the time it takes for individual transactions to complete;
- consistency—the level of predictability, coherency, and correctness of data on which the application operates.

The Enterprise Ehcache architecture is uniquely capable of delivering the appropriate balance of these three aspects of performance according to the specific business needs of your application across the entire scale continuum. To increase throughput, add more stripes to the Terracotta Server Array. To reduce latency and increase available CPU for application operations, add more application servers in the application tier.

The Consistency Spectrum

Across the enterprise, there are typically requirements to support data access along a spectrum of consistency guarantees. This spectrum ranges from purely asynchronous operations suitable for read-only access to fully transactional access to business-critical data. Because the level of consistency affects throughput and latency and is dependent on the business rules of the application, Enterprise Ehcache offers configurable consistency guarantees to different data sets in the same application.

At one end of the spectrum, Ehcache allows fully asynchronous access to cached data. This yields the highest throughput and lowest latency, but the lowest consistency guarantees. In the middle of the spectrum, Ehcache enforces synchronous access to cached data. This yields a balance between fast access to data while reading and a coherent, stable view of the data as it changes. At the far end of the consistency spectrum, Ehcache enforces fully transactional, XA compliant data access.



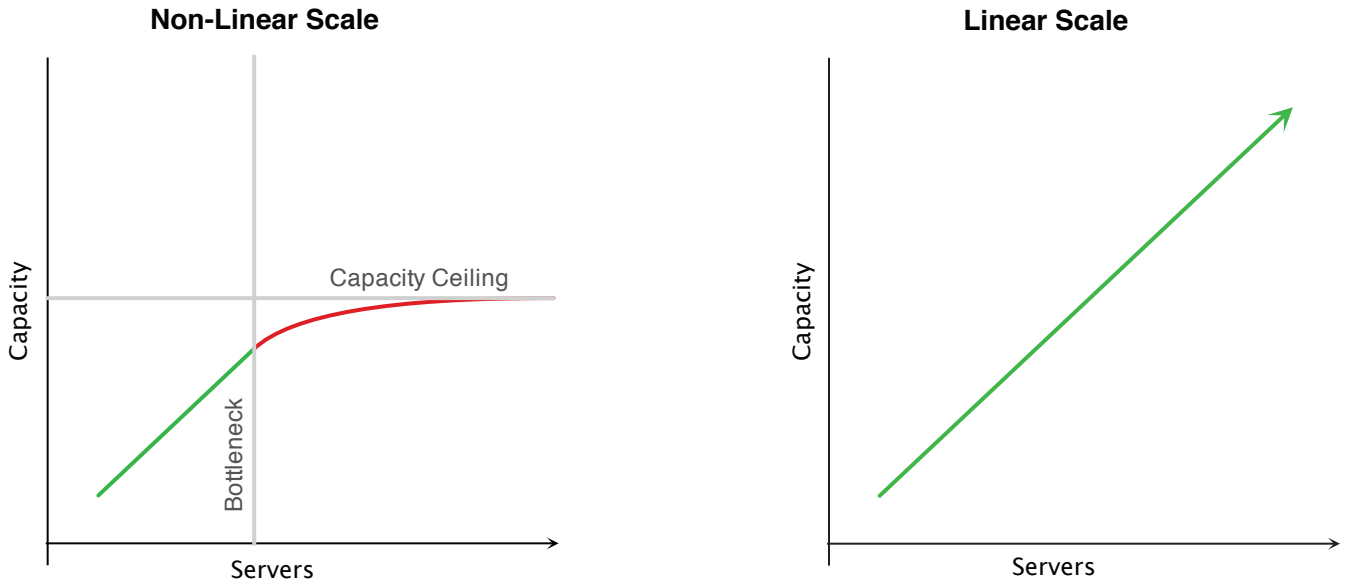
Consistency Spectrum

Enterprise Ehcache ships with a default consistency setting that offers a balance of high consistency and high performance, but is easily configurable to suit the specific requirements of the application. No other caching solution supports the full range of the consistency spectrum on the same architecture and deployment topology and within the

same application using the same API. Enterprise Ehcache is a single solution that delivers predictable and cost-effective performance at all levels of scale and consistency.

Linear Scale-out: Add-a-Brick Scalability on Commodity Hardware

Scaling out on commodity hardware promises a cost-effective and flexible means to quickly adapt to changing capacity requirements and workload. It only works, however, as long as there are no bottlenecks in the scale-out architecture. The scaled-out application must exhibit linear scale along the entire scale continuum, meaning that adding more computing resources yields a corresponding increase in capacity with no diminishing returns.



Bottlenecks—such as a monolithic database server with a maximum capacity that can't be cost-effectively scaled—impose a practical capacity ceiling on the entire system. Beyond this point, adding more application servers yields no benefit.

The combination of the tiered architecture of Enterprise Ehcache and the capability to scale both the application tier and the Terracotta Server Array eliminates bottlenecks and affords linear scale along all dimensions. Adding application servers at the application tier increases the total computing power available for application functions. Adding commodity servers in the Terracotta Server Array increases the total addressable data set size and I/O throughput capacity on tap to the application. Both tiers are independently scalable according to the computation and I/O profile of the application providing unparalleled operational flexibility.

Scale Out the Application Tier

The separation of application logic in the application tier from cache management logic in the Terracotta Server Array allows each to be optimized according to its specific task. The in-process Ehcache cache present in the application tier is optimized for high concurrency and low thread contention that maximizes the performance of each application node.

Because the hardware operating in the application tier is not overloaded with cache server tasks, all of its resources are devoted to application business logic. The application JVM heap can be configured to be relatively small and, therefore, immune to garbage collection operations that cause long service interruptions in peer-to-peer caches.

No other technology offers this separation of concerns and independently scalable application tier in the same deployment architecture and API.

Scale Out the Terracotta Server Array Tier

The dedicated cache server functions of the Terracotta Server Array provide a central authority that enables a number of runtime optimizations not available to other cache technologies. Transactions can be batched, folded, and reordered at runtime to increase throughput. Latency is minimized because no cross-node acknowledgements are required.

The server array can be scaled elastically on demand with no downtime. Each stripe in the Terracotta Server Array is a share-nothing partition of the cache data which is spread across the stripes using a round-robin partitioning algorithm. As a result, new stripes can be added with no additional overhead.

Other approaches to data partitioning use static hashing algorithms to distribute cache data across the grid based on the number of grid nodes. This approach makes it costly to grow the grid to fit more data or handle more work. Such a change in grid topology requires the data to be rehashed and rebalanced across the grid cluster. The work of rehashing and migrating the data to the new grid topology while the application is running can overload the hot grid nodes, causing latency spikes that pull those nodes out of acceptable operating tolerance, thus dropping them out of service. This can lead to cascading failure as more workload is dumped to fewer servers that are also in the process of rebalancing, in turn causing them to fail. As a result, rehashing and rebalancing must be done offline during a scheduled maintenance window.

In contrast to a static partitioning scheme, the round robin partitioning used by the Terracotta Server Array allows new stripes to be added without rehashing all of the stripes. As a result, new stripes can be brought online instantly.

The Terracotta Server Array offers a number of features that allow instant-on cache server deployment:

- A bulk-loading mechanism warms up new cache servers before adding them to the array, protecting the application from the runtime computational overhead and latency of cache loading.
- A kick-start function makes new server array topology configurations instantly available to the application cluster. This means that when new servers are ready for service, the application can make use of their extra

capacity immediately.

No other technology offers the operational flexibility to independently scale the application tier and the data management tier by adding commodity hardware on the fly.

High Availability: Guaranteed Uptime and Data Access

To ensure maximum uptime and cache reliability, Enterprise Ehcache runs in a highly available configuration with no single point of failure. All writes to the cache from the application layer to the Terracotta Server Array are internally transactional and guaranteed. Any application server may be restarted or fail with no data loss.

Similar to RAID, the Terracotta Server Array is striped and mirrored so that, should a server instance in a stripe be restarted or fail, a fully up-to-date mirror will automatically take its place with no data loss and no operator intervention.

No other caching technology offers 100% high availability built into its architecture at all levels of scale with no compromises in performance.

Conclusion

Enterprise Ehcache breaks scalability barriers and performance bottlenecks without application redesign. Based on the de facto caching standard for enterprise Java, Ehcache snaps into enterprise applications for instant, unparalleled speed-up and on-demand, unlimited scale-out.

Compared to other caching technologies, the unique architecture of Enterprise Ehcache delivers access to large datasets; high performance with configurable consistency; elastic, linear scale-out; and high availability with no single points of failure and more built-in reliability than a typical RDBMS.

Enterprise Ehcache is an easy-to-deploy solution for hard-to-solve scale and throughput problems. Through simple configuration changes alone, Enterprise Ehcache gives business-critical applications the capacity for non-disruptive, cost-efficient growth.

About Terracotta

Terracotta's software products provide snap-in performance and scale for enterprise applications. Our flagship product, Enterprise Ehcache, extends the capabilities of Ehcache, the de facto caching standard for enterprise Java and the default cache for Hibernate, Spring, Grails and other leading frameworks.

With more than 250,000 enterprise deployments, including the majority of the Fortune 2000, Terracotta is behind some of the most widely used software for application scalability, availability and performance. For more information, see www.terracotta.org.

Terracotta, Inc.
575 Florida St. Suite 100
San Francisco, CA 94110
USA

Product, Support, Training and Sales Information
sales@terracottatech.com

USA Toll Free
+1-888-30-TERRA

International
+1 415 738-4000

Terracotta China
china@terracottatech.com
+1 415 738-4088