March 19, 2010

For iPhone Launch, Razorfish Uses An Elastic Application Platform

by Noel Yuhanna and Mike Gualtieri
with Mike Gilpin and Adam Knoll

EXECUTIVE SUMMARY

The breakneck adoption of the Apple iPhone posed big challenges for Razorfish, the firm that developed the iPhone activation application for the UK. To support the activation of 12,000 iPhones per hour, Razorfish needed an application platform that delivered performance, scalability, reliability, and quick development. Razorfish chose GigaSpaces eXtreme Application Platform (XAP) for the job. Application development professionals can learn from Razorfish's experience as an example of how to use elastic application platforms such as GigaSpaces XAP to develop and deploy mission-critical, high-volume transactional applications.

SITUATION: IPHONE'S ACTIVATION APPLICATION MUST KEEP UP WITH HIGH DEMAND

Although Apple had a very successful iPhone launch in the US in 2007, there were some initial activation issues due to the unexpected volume of new customers, causing the iPhone activation application to be down for three hours during the first 30 hours of its initial launch. Apple did not want to risk a repeat of this performance for its UK launch. The European telco introducing the iPhone approached Razorfish to help design and develop an iPhone activation application that:

- **Handles up to 12,000 iPhone activations per hour.** The activation platform had to support spikes in iPhone activations during peak shopping hours and in response to special promotions. Each activation involved several transactions, access to data from multiple sources, and systems in two separate companies.

- **Operates continuously.** If you can't activate iPhones, you can't sell them. Razorfish's solution needed to be highly resilient to avoid any interruption in iPhone activations. This means that the application could not have any single point of failure in the software, hardware, or network.

- **Secures sensitive customer and activation data.** A key requirement was to ensure the security of the application and of customer credit card numbers, customer addresses, account information, and other personally identifiable information.

- **Integrates processes and services across diverse systems.** Razorfish needed to execute and orchestrate business processes and services involving Apple's iTunes platform and the European telco's heterogeneous data sources, including its customer relationship management (CRM) application.[1] iPhone activations integrate separate systems for authentication, authorization, device activation, billing fulfillment, credit checking, and network provisioning.

These requirements are challenging enough, but in addition, Razorfish had only 12 weeks to develop and deploy a complete iPhone activation solution.

RAZORFISH USED GIGASPACES XAP TO QUICKLY DELIVER A RELIABLE, SCALABLE APP

Razorfish considered but rejected a traditional Java application server approach because the iPhone activation application needed above all else to be resilient to any single point of failure. A key characteristic of the elastic application platform approach is the distribution of both data and work to two or more nodes.[2] If one node fails, any work running on that node seamlessly fails over to another node. The same level of resiliency can be achieved using a cluster of traditional Java application servers, but at the price of greater code complexity and higher server cost. Using XAP makes this transparent. Razorfish considered various technologies and architectures to integrate Apple's iTunes platform with the European telco's data services to support the iPhone activation application, including technologies such as JBoss, memcached, and Oracle Coherence, but chose GigaSpaces largely because it offered:

- **Tight integration with Java.** With only 12 weeks to develop a working application, Razorfish wanted to use a platform that would leverage its existing Java and Spring development skills. GigaSpaces XAP, written in Java, also offers the OpenSpaces development framework, which provides application program interfaces (APIs) that Razorfish could easily use in its code. OpenSpaces integrates with the Spring framework, easing integration with heterogeneous data sources including mainframe and distributed sources.[3]

- **Performance and scalability.** GigaSpaces XAP's in-memory architecture can accommodate additional order volumes simply by adding more cache servers without requiring additional database servers. Testing showed that the activation application is capable of supporting more than twice the peak processing volume.

- **High availability.** Razorfish deployed GigaSpaces XAP across multiple redundant servers and storage devices to eliminate any single point of failure. If a server node fails, XAP automatically switches processing to a backup with no interruption in service.

- **Security baked into the application.** The iPhone activation application required that the system protect data — including personally identifiable information — at all times. GigaSpaces XAP easily integrates with existing authentication and authorization models including LDAP and middleware security.

When Considering Distributed Caching, Consider Options Beyond GigaSpaces XAP

In addition to GigaSpaces, several other vendors offer comparable elastic application platform solutions for Java, including IBM WebSphere eXtreme Scale, GemStone Systems GemFire Enterprise, Hazelcast Software (open source), and Oracle Coherence.

RECOMMENDATIONS

NEED PERFORMANCE AND SCALE? CONSIDER ELASTIC APPLICATION PLATFORMS

The traditional approach to scaling applications is to configure a load-balancing appliance to spray a bank of application servers with incoming requests from clients. This approach scales effectively as long as the other resources (such as databases or other applications) the application code uses do not become a bottleneck or a single point of failure. Application development professionals building high-volume, highly available applications should consider elastic application platforms when:

- **The ability to scale must be completely transparent.** Unlike conventional application server programming models, elastic application platforms do not require application developers to do anything different in their code in order to scale. The developer consumes a simple API that provides a vast key-value data store that looks like a large shared memory. Behind the scenes, the distributed caching features of the elastic application platform spread the data across multiple servers using a sophisticated hash algorithm. The application developer is blissfully ignorant of all the machinations that go on behind the scenes to distribute the data across the servers.
- **The application requires resiliency by design.** Distributed data caches are resilient by design because they automatically replicate data stored in the cache to one or more backup servers, guided by the policies defined by an administrator. If one server fails, then another server provides the data. The more replicas, the more resilient the cache. Distributed data caches can be vulnerable to data center outages if all the servers are in the same data center. To address this weakness, some distributed caching solutions offer special WAN features to replicate and recover data across multiple data centers.
- **The application has especially demanding performance requirements.** A key performance characteristic of distributed caches is that they store data in fast-access memory rather than on disk, although backing store on disk may be an option. Since this data spans multiple servers, there is no bottleneck or single point of failure. More-advanced elastic application platforms also provide means to ensure that cached data will tend to be on the same server where application code is processing, reducing network latency. Platforms do this either by implementing a "near-cache" concept that places data on the server running the application using that data or by directly managing application code execution in the platform, placing adjacent code and data in cache nodes that are on the same server.
- **The application needs to integrate with other data sources.** Most distributed caching platforms offer read-through, write-through, and write-behind features to synchronize data in the cache with external data sources. Rather than the developer having to write the code that does this, an administrator configures the cache to automatically read or write to a

database or other external data source whenever an application performs a data operation in the cache.

- **The application workload is by nature distributed.** For elastic application platforms offering distributed code execution, developers should also consider the nature of the workload their application will present to servers. If they can divide the workload into units that naturally fit into the distribution schemes these platforms offer, and if current application server architectures are constraining application performance and scalability, then the greater sophistication of these platforms' distributed code execution capabilities can be just what's needed to turn a troublesome application into a screamer.

ENDNOTES

[1] iTunes is one of the applications used for iPhone activation, as well as for loading or updating software on the iPhone, although apps can also be purchased or updated "over the air." iTunes also backs up the data off the iPhone onto the PC on which it is installed.

[2] For more on elastic cache platforms, read: "Elastic Caching Platforms Balance Performance, Scalability, And Fault Tolerance," *Mike Gualtieri's Blog For Application Development & Program Management Professionals,* March 18, 2010 (http://blogs.forrester.com/mike_gualtieri/10-03-18-elastic_caching_platforms_balance_performance_scalability_and_fault_toleranc).

[3] OpenSpaces is an open source project sponsored by GigaSpaces. It "enables scaling out of stateful applications — using Spring. It is built around GigaSpaces' core product — the eXtreme Application Platform (XAP)." For more information, see www.openspaces.org.