How Priceline.com Rid In-Memory Cache Of Java Jitters[i]

Priceline.com is one of the early Internet startups, and as such it has built a lot of its own technology for gathering up pricing information on airfares, hotel rooms, and car rentals. In recent years, it has increased the distribution of data. So it not only needs to gather pricing information quickly from myriad sources, but it also needs to aggregate that data and push it out with a certain amount of gusto.

To speed up the gathering and dissemination of pricing and availability information for the hotel booking section of its business, priceline.com recently built its own in-memory caching system – called Cache 22, of course. This cache is built in Java, like most of priceline.com's front end and all of its back end software is today. Tuning up this in-memory data cache is not trivial, given some of the limitations of Java virtual machines themselves, but Michael Diliberto, chief information officer at priceline.com, and Nasreen Ali, director of engineering, walked EnterpriseTech through how Cache 22 is designed, how it fits into the company's business, and how its use may be expanded throughout the company.

The Cache 22 project, says Diliberto, was one of the biggest development efforts the company has done in recent years, and it solves a few problems inherent in the hotel reservation business.

"You see the evolution of sites like Kayak or TripAdvisor or Google Hotel Finder that are providing access to data – in this case hotel prices and inventory," explains Diliberto. "The change on our side is that instead of just producing a website that people visit to look up prices and book hotel rooms, now we need to also send that information out to other parties for display. Then, they can come back and complete the booking here or complete the booking through an XML interface elsewhere. This has been a large shift, from the production of a website to the production of a platform."

Given the short attention span of Internet surfers, particularly when they are shopping, latencies matter. Priceline gathers up hotel room and pricing information from suppliers such as global reservation systems, like TravelPort, or hotel switches, like Pegasus. These in turn connect to the individual suppliers, which would be the Wyndhams, the Marriotts, the Intercontinentals – all of the major hotel chains and also all of the independents who want priceline.com to sell for them.

The trouble is, it can take up to 2.5 seconds for a query to come back from one of these remote systems, and it takes even longer from the systems of some of the independent hotel operators.

"That's just too long to wait," says Diliberto. "So we had to put together a system where we can collect the inventory and pricing information, all of the details, and keep it accurate and fresh, keep it as up-to-date as possible without burying the remote systems. In most cases, these remote systems are not state of the art systems. They are hotel reservation systems that have been around for years and years. We want to be fresh, but not bury them, and we want to distribute out a very large amount of data at an incredible speed."

Enter Cache 22.

Rather than query each hotel individually as customers surf the priceline.com site, the cache relies on shopping robots that are programmed to query the reservation systems for hotel room availability and price at periodic intervals. This information is put into a cluster of sharded MySQL databases and fed out of in-memory data stores. This in-memory data store is not a commercial product or open source project, but a homegrown tool. To be technical, it uses the Java virtual machine to send data into a hash map in memory in the server cluster, allowing for fast access to information from memory that might otherwise be served from a drive using flash or disk media.

The shopping robots work behind the scenes, and they follow rules set by priceline.com about how frequently data must be updated – the closer to the current time, the more the information is updated, generally speaking, and if there is a big event in a particular city, these robots are tweaked to keep this information as fresh as possible. The robots gather up hotel room inventory from as many as six different sources, but priceline.com usually has multiple sources and tries to find the cheapest rates available for each room.

"We added a lot of heuristics in," says Ali. "We also look at customer demand. So as customers come in and click on things on our Website or from our distribution partners like Google, Kayak and TripAdvisor, we know what the customers are interested in and we try to shop those more and make sure those have the freshest rates. We strive for, and maintain, a very high accuracy rate."

Priceline.com set a goal of serving up hotel price information in around 200 milliseconds for 100 hotel properties. If some of priceline.com's downstream partners ask for more data about hotel rooms, then the response time could get longer as this information is transmitted.

"Our caching infrastructure is currently holding something over 7.5 billion rates and it is updating at a rate of about 42,000 rates per second and we are serving out roughly 28,000 rates per second," says Diliberto. Priceline.com does not divulge its transaction rates or concurrent customer counts.

The Cache 22 system runs on a two-tier network, which is replicated in its two datacenters. The company's primary datacenter since 2000 is an AT&T co-location facility in mid-town New York. For the past four years, the company has operated a second datacenter in an Equinix co-location facility in Ashburn, Virginia. (This is where a lot of companies, including Amazon Web Services, host their servers.)

It doesn't take as much iron to run the Cache 22 software as you might think. For the shopping tier of the Cache 22 program, each datacenter has a blade chassis from Hewlett-Packard with ten brand-new ProLiant BL460c Gen8 blade servers. These blades have two eight-core "Ivy Bridge" Xeon E5 processors, 256 GB of main memory, and two disk drives. They run VMware's ESXi 5.5 server virtualization hypervisor and each server slice on the box is equipped with Red Hat Enterprise Linux and, up until recently, the commercial-grade, 64-bit HotSpot Java virtual machine from Oracle. (More on that in a moment.) Each server in this tier of Cache 22 has three partitions, with a 50 percent oversubscription on the threads in the machine. So each priceline.com datacenter has 30 virtual machines that are used to shop hotel rates or to serve up data from the in-memory cache stored in the JVMs.

The second tier in the Cache 22 setup is the back-end database that stores all of the rates. A collection of older BL460 G6 and G7 blades with quad-core Xeons, each with 96 GB of main memory, two 1.2 TB solid state disks, and a couple of disk drives with a total of 1.2 TB of capacity host a dozen MySQL instances. There is a spare database blade, and each blade hosts two database shards. The data is shared by a key made from several data elements, including check-in, check-out, advanced purchase, and geography. The sharding technique for the MySQL databases was created by priceline.com itself. That is because when the company started building the in-memory cache for hotel information a year and a half ago, there was no commercial option available. The sharding is designed such that the odds that all of the information that you might need about a hotel booking is located in one shard and therefore can be fed rapidly out of the cluster.

The Cache 22 setup is front-ended by eight web servers, four per data center, which feed out to priceline.com's partners and its own web site. The regular priceline.com web site is comprised of around 150 web servers in each data center, according to Diliberto.

One big headache with the Cache 22 setup is one that affects all Java workloads: Garbage collection. The memory given to a JVM is called a heap, and Java allocates and deallocates objects in this memory. So periodically, a garbage collection process is run inside the JVM to get rid of unused objects in memory.

"We love Java as a language and it has served us so very well over the years," says Diliberto. "But we got into a situation that we were allocating so much memory per session that it became a garbage collection nightmare. The garbage collection would need to do full sweeps and it

would pause the system for 30 seconds when we were working on a 200 millisecond response time goal. We tried to shrink down the size of the individual JVMs, but we also don't want the hardware footprint to explode. We wanted to find a way to get through these garbage collection events."

The problem is that the Internet doesn't stay constant. Some traffic might increase at one trading partner, and that would change the memory footprint in JVM, and then the application would have to be retuned.

So last summer, the company investigated the Zing Java virtual machine from Azul Systems. That company got its start making appliances for accelerating Java, and one of the things that Azul created was a very clever pauseless garbage collection routine for the JVM running on its proprietary hardware. Azul has ported its JVM to run on X86 hardware in a virtual instance, with a heap size of up to 1 TB. That's a lot in a world where JVMs choke on their garbage collection with 2 GB or 3 GB heap sizes. Zing is packaged up to run inside of VMware's ESXi or Red Hat's KVM hypervisors.

Priceline.com went live with Zing over the July 4th holiday last summer after a few months of testing, and Ali said that this was the first time that software engineers had not been called in over a holiday weekend to deal with memory issues in the JVMs as conditions in the hotel market changed. The two software engineers that were fussing around with garbage collection and heap memory are now focusing on application development.

Ali says that priceline.com is contemplating what parts of the company's business can be cached next in the manner that the hotel pricing was. The car rental portion of the business will get its own cache, perhaps this year. Caching the airfare part of the business is more problematic.

"Airfares could be accelerated," concedes Diliberto. "We would need to spend some time studying it and working on it. Hotel really lends itself to caching. You don't check in at the Marriott and check out at the Hilton. At the airlines, the complexity of journey logic and individual legs being mixed by carriers and across carriers is more complex and more problematic than the hotels. And the hotels have a bountiful ecosystem supporting it for distribution."

There is a possibility that Cache 22 could be picked up by other operating units in The Priceline Group, including the Booking.com and Agoda.com hotel booking systems (which are separate from those run by priceline.com) or the Kayak metasearch engine for travel and lodging. But these units all run their IT operations autonomously. So it is up to them.

For more information on Zing: **azul.com/zing**
Follow Azul on Twitter: **@AzulSystems**

---

[i] Reprinted with permission, Copyright© 2014 Enterprise Tech. All Rights Reserved.