

BAT Keeps Orders Flowing with GigaSpaces

“When a user triggers a batch process, for example to define routes for the next day, that would have taken two hours before. The grid takes about 10 minutes to do the same job.”



In Argentina, when a consumer products company tried to replace its monolithic field order management system with one based on a three-tier relational model, it found the costs prohibitive. So the company switched to grid computing — with technology from GigaSpaces. Now 15 entry-level servers do the same work as the much larger computers they replaced — for the same capital investment — but twelve times faster!

The company is Nobleza Piccardo, part of British American Tobacco. The problem: The existing system was drowning under the weight of supplying 47,000 retailers. On average that equals 17,000 retailers per day at 30-40 database transactions each — creating orders, fulfilling orders, billing retailers, defining sales routes, defining delivery

routes, determining product availability, updating retailers’ records, and doing other tasks typical in field order applications.

The grid performance boost meant that other applications — at times virtually shut out by the field order application — once again were highly

available. Those included order processing, customer management, stocktaking, pricing, and accounting.

Alejandro Ramallo, who manages both IT and customer relationships for Nobleza Piccardo, discusses their situation: “We were moving from a COBOL flat file system with a very monolithic architecture — where everything ran on mid-range servers. We could have scaled by replicating the system. But the application wasn’t ready for that — without any relational database underneath. “I considered using a typical three-tier architecture with a relational database like Oracle and building the system in Java and using a

web-based interface. But we couldn’t size the architecture. We were moving from a COBOL file system and just adding Oracle would affect performance, and therefore the architecture, radically.

“We were also under strong pressure to minimize capital expenditure and total cost of ownership — mainly because we had a 300% devaluation of the currency.

“So, we decided to buy low-end servers. Typically, we might use a 12-processor machine to accommodate the entire application. But a 12-processor machine is very expensive. So, with the same money we bought 33 low-end machines. These are \$3,000 dual processor Intel Linux servers. We’re using roughly 15 in production and the others as a test environment.”

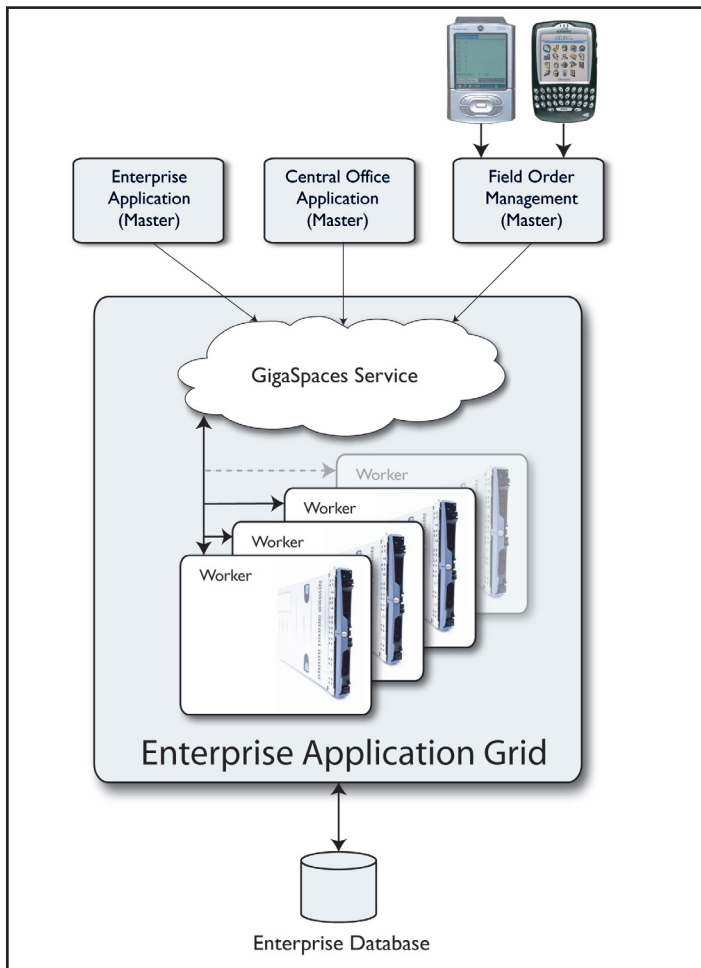
Not Very Hard

But changing hardware was only a partial solution, Alejandro says. “We couldn’t simply replicate the existing batch process on each of these low-end machines. That would impose a ceiling on the processing capability of the software.”

So, Nobleza Piccardo implemented a grid using GigaSpaces. “The software splits the processes into tasks and places them in a task repository. The low-end servers grab the tasks and execute them in parallel. When a user triggers a batch process — for example, to define routes for the next day — that is a huge job that would have taken two hours before. We split that into tasks and deploy them to the grid. The grid takes about 10 minutes to do the same job with five machines working in parallel.

Alejandro explains how it works: “The grid is built on a master-worker model. We have a process called the master process and the master process coordinates other processes, or workers, to execute the various tasks in parallel. That coordination is handled by GigaSpaces middleware.”

“Triggering a batch process would have taken two hours before. The grid takes about 10 minutes to do the same job.”



Building the grid took two weeks Alejandro reports, using two full-time employees. “It was not very hard, and that is the main point of the JavaSpaces — and the GigaSpaces product. It’s a very powerful concept, yet it’s pretty simple. You can learn how to do it in just one day.”

In fact, Alejandro says, if he had it all to do over again, he could implement the grid even faster today — thanks to Rio, the new open source resource management software GigaSpaces supports.

“In our implementation, the assignment of workers is not generic. We tell them where to get the Java code for each task. With Rio that isn’t necessary. Workers don’t know anything about the world. They just know they have to execute a task. Based on the task, workers load the correct code from a central code server, execute the task, and then de-load the code.” Those services are provided to the application via a simple GigaSpaces

application programming interface, Alejandro says. But support for APIs beyond those for JavaSpaces made GigaSpaces more appealing — even if Noblez Piccardo may not need those other APIs right away.

“Yes, I bought a JavaSpaces implementation, but in reality what I got was a whole grid integration middleware,” Alejandro says. “That’s because JavaSpaces is not the only application programming interface GigaSpaces provides. It provides a .Net interface, a C++ interface, an XML interface and everything is integrated in middleware. I considered that in my selection because it tells me where GigaSpaces is going and that it will be supporting the other things I will be doing in the future.”

Enterprise Class

This concern for the user’s broader middleware requirements is one reason GigaSpaces is truly enterprise class, Alejandro says. Another is GigaSpaces’ persistency model. “You are putting your Java objects in the JavaSpaces repository — but if the JavaSpaces server fails, you need a store somewhere to put those objects so you can recover them from the failure. GigaSpaces provides a JDBC interface, so you can plug in any database where you want GigaSpaces to store these objects.”

Other enterprise class features include built-in support for transactions and clusters. “GigaSpaces has one-click cluster configuration,” Alejandro says. “I deploy GigaSpaces on two different machines in a cluster, so if one machine fails, the other machine will handle the work. Regarding transactions, he says, the key is transactional scope.

Alejandro explains: “If the worker fails but has already taken a task out of the repository, then the task could be lost. But GigaSpaces will roll back the full operation and the task will then be put back in the repository and not lost.”

One final reason Alejandro calls the solution enterprise class is technical support. “GigaSpaces was even more proactive than we were,” he says. “They would actually call us and say: ‘You didn’t call us. Let us know if you need help to do this.’”

He mentions a couple of leading IT industry names. “I have support contracts with them and it doesn’t usually work. You can wait a couple of months to receive a response to your ticket — even if you’re paying millions of dollars in software licenses. GigaSpaces is very, very proactive.”



sales@gigaspaces.com
www.gigaspaces.com

USA
 257 Park Avenue South
 8th fl.
 New York, NY 10010
 Tel: 646-862-6935
 Fax: 212-308-7886

Europe
 30 Borough High Street
 London SE1 - 1XX
 United Kingdom
 Tel: +44 709 286 3096
 Fax: +44 709 286 3097

International
 8 Maskit Street
 P.O. Box 4063
 Herzliya, 46140, Israel
 Tel: +972-9-952-6751
 Fax: +972-9-957-6780