Re-make or Re-model?

As developers consider moving applications to the cloud, to what extent will they have to re-think familiar go-to languages to make apps work there?

Languages for Cloud Computing

Distributed Caching

Scaling Horizontally

Changing Data Architectures Midcloud

Re-make or Re-model?

By Jack Vaughan and Rob Barry

As developers consider moving applications to the cloud, to what extent will they have to re-think familiar go-to languages to make apps work there?

THE UPSURGE IN cloud computing presents development teams with challenges that may require them to rethink established ways of building applications.

One school of thought maintains that the best bet is to upload applications as they are to the cloud, doing as little rework as possible. Another school contends that applications should be radically re-cast if they are to fully exploit cloud architecture.

For developers, the issues play out on several levels. Naturally, language is essential to programming. So cloud developers may ask, "Can I use my familiar language tools, or should I learn a new language?"

Software architectures will be reviewed in terms of memory: how development teams allot processes in memory, how they call to relational databases and whether they should use relational databases at all.

NO SHORTAGE OF LANGUAGES FOR CLOUD COMPUTING

Of course, languages are intrinsic to what developers do, and the selection of language can influence the future course of a development effort. To date, .NET, Python, Apex and Java, along with Web services architecture, are the primary languages of the cloud.

But what comes to pass for cloud can vary and carry different names. In certain cloud settings that could best be described as "hosted services," developers merely upload what they have already created and tested on a traditional platform.

While Amazon.com had a significant hand in getting cloud computing rolling, the Amazon Web Services

(AWS) offering, which requires developers to work with Web Services Description Language (WSDL) contracts and Amazon services application programming interfaces (APIs), has been somewhat overshadowed by Amazon's Elastic Compute Cloud (EC2) offerings, which allow users to send applications to the cloud on an "as is" basis.

"It starts at one end with just raw storage and computing capacity. That is what Amazon is," said John Rymer, an analyst at Forrester Research Inc. (who adds that, in fact, Amazon can field its provisions via Web services). With Amazon, said Rymer, you don't have a program model and database.

Languages for Cloud Computing

Distributed Caching

Scaling Horizontally

Changing Data Architectures Midcloud

The Languages of the Cloud

- **APEX:** Salesforce.com's Force.com uses this proprietary language. Apex is a low-level language and is said to be built for use by both professional developers and end users. Visualforce, Force.com's graphics programming interface, is built on Apex.
- **JAVA:** One mainstream language making recent inroads in the cloud is Java. This spring, Python proponent Google opened up its AppEngine platform to include Java support as well. Meanwhile, Schakra Inc. built a Java software development kit, or SDK, for Microsoft .NET Services.
- **.NET:** At its launch, Windows Azure was said to support any .NET language, including Iron Python and Iron Ruby. The emerging consensus has most Azure development undertaken with C#.
- **PYTHON:** Early iterations of Google's cloud approach relied on Python. This venerable object-oriented language was instrumental in the company's rise to prominence. But it is far from mainstream.
- **WEB SERVICES:** Early cloud interest coincided with the growth of high-level Web services that employ Web Services Description Language contracts. While lately Amazon has promoted EC2 as a cloud platform supporting as-is porting of existing apps built using any language, its Amazon Web Services is an option for cloud computing development.
- **YOU NAME IT!:** A broadly popular approach to cloud computing seeks to port applications to the cloud on an as-is basis—thus not requiring rewrites or special languages. —SOATALKBLOG

Languages for Cloud Computing
Distributed Caching
Scaling Horizontally
Changing Data Architectures Midcloud

"You have to do that on your own," he said.

Google is different in that it has the App Engine and the BigTable data store, Rymer noted.

> Low-level programming likely invites vendor lock-in, but it also means programmatic power.

In its first iteration, Google's cloud approach relied heavily on the Python language, which, though popular within the developer army at Google and elsewhere, is not widely known. In recent months, Google has begun to test a Java software development kit (SDK) for cloud development.

"Google provides a program model. That is fundamentally different than Amazon. It provides more of what a developer needs," said Rymer.

But the benefit of providing what the developer needs also invites the threat of vendor lock-in.

Rymer cited Salesforce.com and its Force.com interface. "You have Force.com, where you have a proprietary language: Apex. It is low level," he said.

Some say that graphical tools have come a long way and can streamline development for nondevelopers. As such, it is part of a larger trend that, where possible, seeks to insert graphical modeling interfaces and thus take development out of the hands of developers.

Programmers and architects alike know low-level programming likely invites vendor lock-in too. But it also means programmatic power. "With Force[.com], you have richer tools, and also a kind of access to an application model. It's good for managing business entities," Rymer said.

Google's application model has benefits, supporting Python (and, lately, Java) and the BigTable architecture. Rymer emphasized that the Google scheme is not built for transactional purposes. "It's designed for search," he said. (For more on Google's architecture, see "[Is this the dawning of the age of hosted providers?]" on the SOA Talk Blog.)

A later arrival to the cloud development fray, Microsoft has pledged to support any .NET language in the cloud. Such languages include VB.NET, IronRuby and IronPython, but the most likely first choice will be C#, the flagship language of .NET. Recently, a Java SDK for Azure, Microsoft's cloud platform, has even arrived.

If you like many flavors in developer languages, the cloud does not disappoint.

DISTRIBUTED CACHING

Ever-faster processors and cheaper blade computers set the stage for the cloud era. They also insulate developers from some performance and scal-

Languages for Cloud Computing

Distributed Caching

Scaling Horizontally

Changing Data Architectures Midcloud

ability issues. But with memory architecture in particular, would-be cloud architects may want to re-visit their designs if they want to see continued improvement gains based on parallel operations (a form of computing in which many calculations are carried out simultaneously).

While many factors have combined to make parallel cloud computing widely attainable, parallelism is still difficult. In recent years, distributed caching and machine-level parallelism, especially in the Java space, have matured quite a bit, and the diverse offerings of assorted vendors may indicate how some cloud applications will evolve.

JavaSpaces, for example, is a service specification that arose out of a smorgasbord of Java community standards. Based on the notion of a Tuple space (a means for dividing associative memory into units that can be accessed concurrently), the JavaSpaces framework enables scalability in parallel processing with distributed object caching.

In recent years, technologists have worked—somewhat under the radar—to commercialize this and other parallel schemes, especially for applications targeting traders on Wall Street given the real-time nature and high-computing requirements of these applications.

On Wall Street, Enterprise JavaBeans (EJBs) and associated proxies were used to effectively abstract the location of objects, but this added many layers to system architecture. That meant inefficiencies and latency. "Garbage collection"—a problem that in many ways Java solved—still created performance issues where ultra-high performance and efficiency were concerned.

With memory architecture in particular, would-be cloud architects may want to revisit their designs to see continued improvement gains.

Standard J2EE servers needed improvements, many of which came from specialty vendors of distributed caching schemes, persistent object stores, JVM accelerators and failover architectures.

Nati Shalom, the chief technology officer at GigaSpaces Technologies—which has a commercial version of JavaSpaces—offered a view of how development issues in the cloud could pan out. In July 2009, GigaSpaces rolled out its eXtreme Application Platform (XAP) 7.0 application server, which can react to loads in real time and automatically provision additional resources as required.

"If an application was originally built

Languages for Cloud Computing

Distributed Caching

Scaling Horizontally

Changing Data Architectures Midcloud

in a static way," Shalom said, "moving that application unchanged to the cloud means you have simply placed it in another hosting environment."

That is a waste in terms of scalability and performance benefits by his measure. In some cases, it can also create application failure.

"For us, cloud is a very natural fit, because we've been dealing with cloud before the name cloud came into the world," said Shalom. "It brings a lot of challenges that we faced in a niche market. Now that cloud is getting more affordable, more people are migrating. And more people are getting into the issues."

SCALING HORIZONTALLY

Practices that fared well in simple Java apps will not play well in the cloud. As a result, many developers will be forced to rethink familiar practices such as data calls and storage. Programmers versed in high-performance transactional Java have already discovered this hurdle.

"If you are using the database to store application-specific data, transient data or data related to a user session, it is not going to scale on the cloud," said Jeff Hartley, the vice president of marketing at Terracotta Inc., a manufacturer of high-availability Java infrastructure software.

At this year's JavaOne conference, the company said it was working with VMware Inc. to reduce Java application complexity for cloud computing.

"Databases don't scale well 'horizontally' [that is, simply adding similar machines to handle work] as a general rule," Hartley said. They scale vertically, he noted, but only if you "buy bigger hardware."

Hartley said the material in a database can constrain the scalability of clouds. "You can't just add another machine for a database as you can with something like Teracotta," he said.

In fast, distributed systems, it is preferable to keep user session information out of the database, and Terracotta software helps in this regard. You can place session data in distributed memory rather than database memory, and "it will scale horizontally quite easily," according to Hartley.

Distributed caches and high-availability application clusters have become bona fide best practices for some types of Java and SOA applications, said James Staten, a principal analyst at Forrester Research Inc.

"They are very characteristic of the applications we see going in the cloud," he said.

"One of the key problems of cloud-based implementations is latency between the components of the application, which is something GigaSpaces and others address."

Among the larger players in a small legion of data caching providers is Oracle Corp. When it purchased Tangosol Inc., the maker of the Coherence caching software line, the database company delved into the data caching

Languages for Cloud Computing

Distributed Caching

Scaling Horizontally

Changing Data Architectures

Midcloud

space.

Cameron Purdy, Tangosol founder and now Oracle vice president of development, explained what he described as the "distributed caching space," and its relationship to the cloud. When e-commerce became a hot application early in this decade, session management was immediately a problem. Coherence addressed the problem with caching and failover.

Among other tasks, Coherence manages sessions. The technology can be used for caching highly used catalog information and so on. By offloading elements from persistent storage, users can improve scalability and reduce queuing.

Released in mid-July 2009, Oracle's newest version of Coherence demonstrates further improvements in session handling.

"Some of the new options include a no-locking option for very high-concurrency systems, where applications are load-balanced over many servers," Purdy said.

In its latest version, the Oracle Coherence software provides terascale support to in-memory data grids.

Optimizing Data Resources

ALTHOUGH PRIVATE NETWORKS have featured distributed caching for a while, moving the model into the cloud presents several security risks.

"It was designed for private networks so that if you send a message to a single IP, it goes out to everyone in the network listening to that IP," said Emil Ong, the chief evangelist and a lead developer at Caucho Technology Inc.

"We're trying to make it so developers can write their code on one machine and migrate it to the cloud without having to change any of their code," Ong said.

When starting a new application instance in the cloud, developers often have to write complex custom scripts to push out an application.

"We take care of that by putting that deployment mechanism in there," Ong said.

To build applications that are fully optimized for distributed resources, Ong said being meticulous is a must.

"It's easily the most difficult problem in computing, where you have multiple actors working in tandem to make use of certain resources."

In June 2009, Caucho Technology announced a new release of its Resin 4.0 application server, which features distributed caching. Caucho has traditionally made speed its primary concern, Ong said. Distributed caching lets it boost Java server performance in the cloud. It allows the results of complex computational processes to be stored for future use to cut back on repetition. —J.V.

Languages for Cloud Computing

Distributed Caching

Scaling Horizontally

Changing Data Architectures Midcloud

A safeguard agent can automatically detect and correct service disruptions.

CHANGING DATA ARCHITECTURES MIDCLOUD

Perhaps because relational database calls pose issues in highly distributed architectures, Google took a new approach when building out its flagship search site, which has always been somewhat of a poster child for cloud computing. Although enterprise developers tried to recreate relational database-heavy transactional apps on the Web, Google kept things flat—for example, went without a SQL database. The question is whether this architecture, which serves Google cloud apps such as text search, can also apply effectively to classic enterprise apps.

As developers look to move applications to the cloud, they will discover a mixed landscape. They will find Amazon with its SimpleDB. They will find Google with Hadoop and MapReduce. They will find out how prevalent stateless representational state transfer (REST) and nonrelational architectures have become. Do they have to shun relational methods that have been at the heart of enterprise development for a long time?

Microsoft, for example, has had to contend with this issue as it tries to fashion its Azure cloud architecture for use by established applications and newbie applications alike.

The company's original cloud architecture took an approach to data similar to first-generation clouds from Amazon and Google. At the time, that meant forgoing SQL, the familiar workhorse of modern IT.

Microsoft pumped up its research and development on REST architectures. But in July 2009, the company shifted gears and announced an additional data option: a sort of "SQL Server in the cloud" for developers that did not want to start from scratch with data in the cloud.

Clearly, for Microsoft and its Azure cloud, the complexity of moving to a new platform is no simpler than it is for others. One wonders, what do you have to learn or unlearn to program for Azure?

"You won't have to unlearn anything," said Roger Jennings, an analyst at OakLeaf Systems. "The data-oriented developer had to wait for SQL data services. That is where Microsoft did a midcourse correction. Now they are porting SQL server to the Azure fabric," Jennings said.

Up to that point, he said, Microsoft had a version of SQL Server 2005 that had no relational features—that used an entity-attribute style DB. "Developers didn't like it at all," Jennings said. "It didn't have the ability to do joins.

They could not leverage their basic skills. The truth was [Microsoft was] trying to emulate Amazon's SimpleDB."

As Hadoop and other data architectures augur some disruption in conventional development circles, one

Languages for Cloud Computing

Distributed Caching

Scaling Horizontally

Changing Data Architectures Midcloud

can count on vendors to enter with products to smooth the bumps. Cloudera Inc., for example, sees its mission to further commercialize Hadoop, adding ease of use for developers.

"Our flagship product is a distribution that is in a convenient package," said Christophe Bisciglia, Cloudera's founder. This package includes, for example, Amazon machine images.

According to Bisciglia, one can think of Hadoop as "a large-scale batch data processing system." The main components include a distributed file system and a process or computation engine (i.e., MapReduce).

Formerly at Google, Bisciglia discovered the difficulty early on in getting people to think about how to work with large amounts of data, a real concern in the cloud. As a result, he began teaching a course on the subject at the University of Washington.

"People had no exposure to dealing with data at a scale of Google or Yahoo," he recounted. "What is challenging is to have people re-frame and rethink existing problems in terms of MapReduce. It is a new way of thinking about data."

Various high-level languages, such as Hive, can impose data structure on data files, said Bisciglia, "so you can do SQL" while working with MapReduce.

"It's often the first thing people try and do," he said. "The real thing to understand is that it is a batch database system, not a real-time system."

> Various high-level languages, such as Hive, can impose data structure on data files, so you can use SQL with MapReduce.

For his part, while admitting to some scaling advantages in Hadoop-style computing, analyst Roger Jennings chose to emphasize the importance of SQL.

"The people who are saying SQL is dead for the Internet are wrong," he said, pointing to Facebook as an example of a very large, very modern, cloudlike website that uses a SQL database for its store. ■

Jack Vaughan is the editor in chief of SearchSOA.com. Write to him at jvaughan@techtarget.com. **Rob Barry** is a news writer for SearchSOA.com. Write to him at rbarry@techtarget.com.

Editors' note: *This chapter on application development for the cloud is the second part of an e-book on cloud computing that also includes chapters on CIO strategies for the cloud, Software as a Service and Infrastructure as a Service.*