

PROCESSOR

Cover Focus Articles

 [Click To Print](#)

General Information

May 21, 2004 • Vol.26 Issue 21

Page(s) 1 in print issue

Share & Enjoy

Utility Computing Promises Flexible Resources For The Enterprise



nt>

Imagine being able to distribute computing and storage resources throughout a company and access them from any PC or server transparently, whenever you need them. That's the promise that utility computing offers. But like most broad enterprise initiatives, the devil of utility computing lies in the details.

To begin with, it can be a chore to even get a succinct definition of what utility computing really is. It helps to start with a technology that is better defined and understood: grid computing. In a grid computing environment, multiple hardware resources are virtualized together to form a single entity that can be applied to a problem. The classic example of a grid computing solution is the SETI@Home project, which has massed together thousands of home computers around the world to analyze signals from radio telescopes.

James Kobielus, a senior analyst with Burton Group, defines grid this way: "It's a virtualized computing environment, meaning that all of these grid environments allow you as a client to basically access and request and bid for resources—execution resources and data storage resources—within some cloud of resources. You access, as a client application or user, all this capacity through an API—an application programming interface provided by the grid environment."

Moving up the ladder from grid to utility, you enter the world of marketing. According to Kobielus, utility computing is less a specific technology than a sales strategy. "Utility computing is a utopian concept, it's a paradigm; it's driving a lot of development in the industry. When IBM and HP and these others talk about utility computing, they're often using it to refer to on-demand computing, generally, and also to use it to encompass their take on grid computing and autonomic computing."

■ What Can It Offer?

So just how can a business leverage utility computing for its applications? To begin with, you need to make your applications available. This means wrapping existing legacy applications, using technologies such as Web services, or engineering your applications the right way from the start using SOAP or other distributed computing methodologies. For businesses already leveraging J2EE or .NET, many of their resources may already be potentially sharable once they have had Web Service Descriptor Language entries created for them and have been placed into a UDDI registry.

The big benefit of this approach is that it offers the freedom to locate and share computing resources and software throughout the enterprise and to outsource resources where it makes sense. "If you're a small to midsized business, you don't have as big of a budget as the Charles

Schwabs of the world to put a team on this,” says Kobielus. “What you want to do is . . . acquire software services and leverage them all to the hilt so you don’t have to acquire more capacity until you absolutely need it because you can make maximum use of existing computing capacity.”

“Likewise, you don’t want to [buy, license, or develop] software that provides redundant functionality with your existing software,” Kobielus adds. “Say you have procurement software that you’ve already licensed or developed. Well, if somebody else in your organization feels a procurement-related need or any kind of application that requires some sort of access to procurement functionality, they don’t need to rebuild redundant software to do that. They can reuse the existing modules in the software that you’ve already deployed.”

■ Not Quite Ready For Prime Time?

However, as with most new technologies, the hype seems to be leading the capabilities, at least at the moment. Kobielus expects it to be a long, slow slog to bring utility computing into regular use.

“The product of Web Services and SOA and utility computing in a standardized way has not yet been realized because the standards are not completely there and the vendors haven’t yet brought their platform even up to speed with the latest,” Kobielus says, noting that it could take two to five years for the industry to bring the Web Services stack up to speed. “Eventually by the end of this decade, we’ll have a mature WS stack,” he adds.

“The thing is these older approaches have been implemented; billions and billions of dollars have been spent bringing up the legacy systems. Then they’re going to still stick around for 10, 20 years in many organizations. I mean, companies are loath to obsolete those investments overnight,” says Kobielus. “Middleware, evolutionarily, then becomes like a coral reef, where newer approaches accrete on top of older approaches and they have a coral reef that builds up. And eventually the older stuff begins to pedal into the subsoil, and then it disappears—it’s obsolete and [it] decommissions. And the newer stuff just keeps accreting on top of the older stuff.” ■

by James Turner

[View the chart that accompanies this article.](#)

(NOTE: These pages are PDF (Portable Document Format) files. You will need Adobe Acrobat to view these pages. [Download Adobe Acrobat Reader](#))