

Deep Blue

A Basal Approach to Grid Computing: Let's Start With Baby Steps



ANURA GURUGÉ

The problem that grid computing intends to solve has, for the last decade or so, been the ugly little secret of IT that we really don't want senior management, especially the "bean counters," to know about. It's our very own "PCgate"; though one has to also immediately defend much of the IT community by pointing out that it really was a problem we never gave much thought to.

The problem I'm referring to is the fact that most corporations today have a surfeit of unused computer resources—whether it be processing cycles, memory or disk storage—despite the perennial gripe as to how we need more computing power. The discrepancy being that much of this unused capacity is at the desktop and departmental levels; i.e., PCs and local servers, whereas the additional capacity is invariably required at the data center.

Yes, at this point, us "ol' mainframers" from the '70s, who took so much abuse about our antiquated, monolithic computing model, find it hard not to smirk. But PC-centric distributed computing is here to stay, indubitably, and it's up to grid computing to help us get around this embarrassment. So, to that end, I like to define grid computing as a means for better utilizing all the computer resources belonging to a specific community—whether that be a corporation or a Wi-Fi-equipped neighborhood. To this end, if perchance, you haven't encountered Berkeley's SETI, please visit www.setiathome.ssl.berkeley.edu and check out this collaborative project, which pre-dates grid computing by at least a decade.

Let's Not Complicate It Too Much

All the major server vendors, ably led by IBM, now have grid computing-related solutions. As to be expected, the emphasis is on better resource sharing at the server level—and I am slightly miffed that IBM's latest definition on this topic starts off talking about clusters of servers. Well, we all know that grid computing at the server is relatively easy to solve, and is getting easier as IBM (and others) roll out additional server virtualization technology that makes server clustering that much easier.

The challenge is that of gainfully harvesting the desktop resources. Adobe, with its After Effects 6.0 Professional, started to pave the way, and IBM is helping quite a few companies (e.g., Bowne & Co. at www.bowne.com) extend the reach of grid computing beyond the data center—albeit with customized projects. The goal of this column is to trumpet the need for grid computing across the board, and at the same time, advocate that we try to tackle this using baby steps.

Complexity is killing us at every turn. I look at XML Web services and shudder to think what a great job we've done of complicating it to the point that many are no longer sure

where to start. Realizing grid computing is obviously also not a simple task, we need to try and keep it as simple as possible. There are many variables (e.g., platforms, languages) and issues (e.g., security, workload partitioning) that have to be cogently addressed. And, to that end, we have some increasingly sophisticated architectures such as Open Grid Services Architecture (OGSA). But, I also strongly believe that we can tackle this problem as small, easy tasks to begin with—rather than feeling compelled to tackle the entire problem using the latest and greatest technology.

Two Technologies: One New, One Old

In my opinion, Web services and Terminal Servers (e.g., Citrix, Windows 2003, and Linux) provide us with a decent enough basis to start tackling the PC waste problem. Given that I've already said that Web services are getting too complicated, some of you are no doubt wondering why I'm now touting them as a way to solve the problem. In many ways, grid computing is the much awaited killer application for Web services. Conversely, Web services is what will save grid computing—and this is already acknowledged by OGSA.

With grid computing, we need a standardized, platform-independent way to farm out "units of work" to all these sparsely used PCs. Grid services is tailor-made for that. If we're only going to use it on a corporate basis, behind the firewall, we could, if we refuse to get sidetracked, keep many of the security and validation-related issues in perspective and not get too bogged down, adding so many layers that the whole thing becomes unwieldy. But Web services, with all its auxiliary standards, are already here, and we need to avoid trying to invent another scheme when Web services is more than adequate.

The same with terminal servers. Don't scoff. They do address the issue, especially if we use them as a way to avoid having to upgrade desktops yet again. There is another hidden benefit: With a terminal server approach, we can sneak Linux on to the desktop, while still letting users have access to Microsoft Office.

So, the bottom line is that we have to embrace grid computing with gusto, and immediately start better utilizing all this unused PC capacity. Terminal servers give us a quick fix, while Web services is the longer-term solution. But let's not wait anymore. Let's tackle PCgate before it becomes a scandal. **Z**

About the Author

Anura Gurugé, an ex-IBMer (1974-1980) at Hursley, U.K., was "Mr. SNA" for more than two decades. Visit his Website for other details of his books and work.
e-Mail: anu@wownh.com • Website: www.guruge.com