THE CLIPPER GROUP
Navigator™
Navigating Information Technology Horizons℠
Published Since 1993
Report #TCG2003012
March 31, 2003

Using *zSeries* as a Grid Server — Many Unexplored Possibilities for the Enterprise

Analysts: Anne MacFarland and Mike Kahn

Management Summary

In a restaurant kitchen, many ingredients and dishes are prepared to a point where the demand of patrons can be satisfied in a reasonable time. Bad estimations of demand waste costly ingredients, so dishes are prepared only to points where they may be kept on “hold” – or finished off for service to the patron. These points of no return are calculated carefully as part of menu planning. Undue quantities of unrecoverable capacity (i.e., what ends up in the garbage) can wreck the bottom line.

Many enterprises are, to some extent, in the same on-demand mode as a restaurant. Lean inventories, trackable delivery, and just-in-time purchasing are crucial to minimizing overhead. The current business climate gives still more reasons to look for ways to do more with less – and enterprise executives are now eying the IT infrastructure. Over-provisioning for flexibility is like a restaurant precooking all the food it hopes its patrons will order. In lean times, it does not make sense. **The *demand business model* requires agility – the ability to dedicate more assets to the needs of the moment. Unused capacity, or “white space,” must have the ability to be easily redeployed to meet business needs.**

Some of you might remember when application workloads (called programs, back then) were brought to servers in the form of decks of punch cards. With software, networked servers and storage, and adequate bandwidth, that portable approach to delivery can be used again, in the form of grid architecture, which frees up infrastructure from the bondage of permanent configuration. **If workloads, or parts of workloads, are transportable between servers, business agility increases. You also can get better utilization of your resources. These abilities - to do more with less - are why enterprises want to use grid architecture.**

This is a good step but there can be more. If you don’t constrain yourself to thinking about the grid in terms of simple servers, like rack-mounted PCs or blade computers, there are many possibilities for further efficiency. *What if you broaden your thinking to include computers capable of executing many applications simultaneously, like a SMP server? What if you could partition that environment to look like many identical smaller computing environments to the applications that are deployed on them (and they don’t even know that they are running in a virtualized environment)? What if you could manage these virtual servers as part of a grid?*

All of this, and more, is possible if you add a IBM eServer *zSeries* mainframe to your grid. Once the shock wears off, read on to learn how zSeries can be one or more nodes in a grid, a grid by itself, or a grid that is part of a larger grid.

IN THIS ISSUE

The Elastic Grid

In the name of cost-cutting and consolidation, enterprises need to manage IT assets (software and hardware) in the same way that they handle logistics, i.e., assets must be delivered "just in time" and "only when you need them." **Grid protocols provide a way to do this – if they can be used for the core applications that support the enterprise's operations.**

The deployment of grids has been mostly confined to homogeneous herds of PCs or small servers and to limited, divisible, often scientific workloads. Their goal has been to accelerate throughput. The few variables were great for gaining knowledge about how grids can work, but **the real benefit is in extending the concept to meet the needs of the enterprise, where workloads are not always divisible and are frequently real-time urgent.**

The challenge is to take the grid's ability to utilize resources more efficiently through scheduling and staging, and stretch it to be broadly useful in an enterprise with many workloads, many IT platforms, little time and less money. If you enrich the variety of resources in the grid, it becomes more enterprise-useful. If you take larger nodes – mega nodes[1], and use their partitioning, virtualization, and automated provisioning capabilities to deal out applications and data deftly, you get a lot more work out of less infrastructure – perhaps a lot less. **You now have an elastic grid with some areas of multi-dimensional stretch.**

What If?

Consider the following:

- *What if* you could have a node that had the manageability features and flexible capacity to be invoked as needed independently of what else it was doing?
- *What if* you could have many nodes in a single enclosure that could be allocated and managed in a uniform way for localized optimization?
- *What if* you could configure a small sub-grid within a box, and link it to other grids, or use it as a node of a larger grid?
- *What if* a grid node could behave as if it were hundreds of logical nodes? What if there was software and automation to manage these images?
- *What if* you needed your grid nodes close to each other and close to particular data repositories but still wanted these assets to be part of a larger grid?
- *What if* you could use flexible memory and internal communications links to provide optimized transactional or database processing on a grid node?
- *What if* you could have a node that was internally redundant and self-restarting, and could be a repository for grid administration services?
- *What if* you could use processor virtualization to emulate grid architecture for the complex testing and development that is needed to quickly bring new grid-based solutions to market?
- *What if* this could be done with a platform that you may already have in place, or could easily acquire?

zSeries Can Be Many Kinds of Nodes

With IBM's *eServer zSeries*, you can have those what-ifs, using a node that can do a wide variety of things, depending on the role you assign it. The zSeries is IBM's latest generation of mainframe servers, the workhorses of many large enterprises. zSeries has been endowed with the capability to run many Linux applications simultaneously. Since many grid implementations are based on Linux, the many possible roles of zSeries in a grid are considered.[2]

[1] For an exploration of this term, see **The Clipper Group Explorer** dated March 30. 2003, entitled *All Nodes Are Not Created Equal — Thinking Differently About the Grid Nodes* at http://www.clipper.com/research/TCG2003011.pdf.

[2] If you require *Windows* applications to run on in a grid, the zSeries is not the answer. However, zSeries can still manage Windows-based servers in a grid; see discussion later.

zSeries as a Node

A mainframe as a single node? This is hard to imagine. How about a single zSeries processor dedicated as a grid node? That's a lot more plausible and doable. (And, yes, you could have a zSeries with only one active processor[3].) Of course, this is a lot of processing power but if you want it, zSeries can deliver it by running a single Linux image on a whole processor.

zSeries as a Multi-Node

Since you can have more than one processor in a zSeries, you could configure two or more as whole-node Linux systems, as just described. This is one way to for zSeries to deliver multi-node capabilities.

Another way, and more likely, is to configure a single processor to run a varying number of Linux images under z/VM[4]. The processor is divided so that each image thinks it is a standalone. The number and size parameters of the divisions is variable, up to the capacity of the processor. This would provide a single tier of grid.

zSeries as a Mega-Node

What if you combined both of the ways that a zSeries can become a multi-node?

1. Dedicating a number of processors to Linux workloads, either individually or as a pooled resource
2. Using z/VM to partition each of them into many virtual images of Linux?

Well, that will make zSeries into one-or-more tiered pools of Linux partitions that Clipper calls a *mega-node*.[5]

Clipper's Glossary of Grid Terminology

In **The Clipper Group Explorer** dated March 30, 2003, entitled *All Nodes Are Not Created Equal — Thinking Differently About the Grid Nodes* (see footnote #1 on the previous page), we introduced our own terminology for *gridspeak* because we wanted you to think differently about the grid. Here are a few useful terms.

- **Node** – an independent (think "Stand-alone") computing system, capable of executing a single instance of an operating environment across one or more processors
- **Multi-node** – a set of independent nodes that exist in the same rack or cabinet (think "blade server" or SMP server), possibly with some shared architecture and/or management, which is transparent to each internal node
- **Mega-node** – a computer system capable of hosting several-to-many Multi Nodes simultaneously
- **Grid** – a collection of nodes interlinked by a network over which workloads are deployed, optimizing access to available resources
- **Mixed-Grid** – a grid that is not homogeneous, i.e., it contains different kinds of operating environments (think "combinations of chip and operating system")
- **Multi-Grid** – a network of connected grids
- **Grid manager** – a piece of software that facilitates aggregation, service arbitration, reservation, scheduling, and use of grid resources.

zSeries Nodes in a Grid

The possibilities are many for a mega-node in a grid. Each is a different way of delivering partitioned processing capacity to Linux applications. But isn't that what many enterprises want to do in their grid? So what makes it a grid?

[3] Mainframers call these *engines*, because there is more to an engine than a single processor, but we will refer to a singular processor complex as a *processor*.

[4] z/VM is a special operating system called a *hypervisor* that serves as a *host* for other operating systems (such as Linux) that run under it as a *guest*. z/VM does its magic in two ways. It can amalgamate or pool a number of mainframe processors into a single virtual server. Because the number can be less than one (i.e., a fraction of a processor), z/VM can also be used to partition a single processor into several (or many) virtual servers. It is the range of possible fractional-to-many virtual servers that give z/VM its unique ability to partition the right-sized virtual server for the application's requirements.

[5] If you want to run Linux, or Unix Applications that can easily be ported to Linux, zSeries may be the best mega-node.

Grid architecture gives the ability to deploy workloads, for a duration of time, across a collection of networked nodes to optimize access and/or throughput. This is more than just connectivity. The nodes need to be managed as a pool of available assets at the application (workload), logical (data access) and physical (device) level. **It is the coordinated grid management scheme, under the control of software that we call the *grid manager*, which makes it all happen.**

zSeries Node in a Grid

As long as a zSeries node (a.k.a, one or more Linux images running on a zSeries processor) can be defined as a resource available for management by the grid manager, **any of the incarnations of a zNode[6] described above can participate in a grid,** whether all within a zSeries, or more widely connected, and distributed.

zSeries as a Grid

Since interconnected nodes under common management define a grid, **zSeries can host a grid within a single processor or across one or more processors in a single mainframe**. The grid manager could reside within this grid, and *everything grid* could be done within the zSeries mainframe. Of course, most environments will have other assets in the grid.

zSeries as a Multi-grid

Since multiple virtual nodes can turn a single zSeries processor into a grid, and because there can be a number of these within a single mainframe, the **zSeries can itself host a *multi-grid***. Further, it could connect multi-grids across several zSeries[7] via the LAN/WAN network. In addition, **zSeries can be part of a larger heterogeneous external grid**.

Further Roles for zSeries in a Grid

Since it is easy to carve out a grid node on the zSeries, it is easy to define separate development environments with safety. The adaptation of computing grids to accommodate interactive workloads is something that requires extensive testing, all made easier in a virtual environment. Because zSeries can deliver resources according to policy, less-critical workloads can be cut back or deferred with ease during unplanned peak periods.

Since most of the larger zSeries have traditional (non-grid) application workloads running z/OS, including those workloads most critical to the enterprise, there may be some significant advantages to having the grid applications very close to the enterprise applications and databases. zSeries can provide speedy access between application partitions on the same mainframe using *HiperSockets*, including access from Linux applications to enterprise data in IMS, CICS, Oracle and DB2 databases and other applications.

zSeries' internal LANs (called HiperSockets) and flexible memory and processor allocation offers many ways to spot optimize. They are key to its superb performance in transactional and database environments. They are important to consider as applications become more interactive (whether through Web Services or through Grid protocols).

In addition, all of the mainframe's innate high-availability characteristics also pertain to work being done on Linux partitions under z/VM. And **zNodes benefit from the rock-solid security that exists between zSeries partitions**.

Don't forget that **zSeries has had many generations of dynamic storage virtualization**, with strong security built-in, as well as other storage advantages (such as mature hierarchical storage management). **ZSeries brings these superior capabilities to the grid.**

zSeries as Host to Grid Management Components

A fully-distributed large grid faces certain organizational challenges. The larger the grid, the more that messaging is needed to arbitrate the workloads, and even more if file locking and cache coherence must be maintained. The messaging of distributed management schemes can get unwieldy, much like the cacophony in a meeting of active participants.

[6] Short for *zSeries node.*

[7] *Parallel Sysplex*, a vehicle for clustering zSeries together and with storage, is not required to tie multiple zSeries servers together in a grid. This can be done through the network.

In a grid, particularly a data grid[8], a resilient point of aggregation would be useful, as a repository of images for restore (or a cache to deploy), for cache and file system coordination, and for a simpler approach to pervasive security issues. This is not a point of centralization, but rather like the recording secretary of an overly large loud committee, who exists to remind and inform, not to run things. Of course, any point of aggregation must be extremely resilient and responsive to unexpected workloads. **zSeries may be the best-of-breed server for hosting these mission-critical grid management functions.**

Is Any of This Real?

IBM is offering a range of ways to deploy zSeries as part of a Grid. It will offer zSeries for Grid either as *Linux on zSeries*, or using Linux as a gateway for z/OS on zSeries, depending on what you have and what you want. The *Globus 2.0 and 2.2 Toolkit for zSeries* are available on both IBM and Globus.org web sites for those who want a do-it-yourself approach. The *IBM Grid Toolbox* will include a Globus Toolkit binary for Linux on zSeries with its other *eServer* binaries, which offers a more streamlined deployment. *WebSphere*, the broadly useful application server and infrastructure tool is, of course, available for zSeries and Linux on zSeries, and will be a useful tool. **A more complete-ready-to-deploy solution approach to deployment is afforded by partners such as Platform Computing[9] and DataSynapse[10], who have ported their grid managers to zSeries.**

The inclusion by IBM of a zSeries node is an obvious early step in the direction of node variety. IBM has consistently espoused heterogeneity in its server product lines and in its discussions of grid architecture, and not just as a possibility in some vague future incarnation. By using Linux as a gateway to connect grid nodes with specific capabilities, and by building and partnering for a core set of applications that can run similarly on all of its platforms, IBM is building a comprehensive space for commercial grid computing.

Farther down the road, the leveraging of zSeries Parallel Sysplex in a grid environment might allow multithreading across a grid domain. It may allow management of grid workloads, not just to available processing, but also to dynamic business priorities. This is beyond anything that IBM has announced, but stimulating to consider.

Early Adopter of zSeries-as-a-Grid

The University of Florida has chosen a Linux-only *z800* and *z/VM* to use in its research on grid computing. It will act as a virtual grid to simulate different types of grid environments. While this kind of ever-changing deployment may be an extreme situation, it proves that zSeries can act as a *grid-in-a-box*, as well as many of the other modes described earlier.

Economic Ramifications

If your view of a mainframe is that of an expensive computer you can't afford, take another look. **The (albeit short) history of Linux on zSeries has produced hundreds of satisfied customers, who have found the right value for their enterprise.** Remember, Linux can run on the same box as z/OS, with the Integrated Facility for Linux (IFL), without being subject to software licensing fees on traditional partitions (e.g., the costs of running under z/OS). Customers will be able to start small by partitioning a Linux image on an existing zSeries[11], or by buying or leasing one of the small, Linux-only *zSeries 800*.

Most enterprises know where they would like to go with technology, if the economic environment were different. It is the cost, the unavoidable risks (both known and unknown), the need for highly-specific process definitions, and the lack of clarity of future market directions that keeps many businesses prudent to the point of pause. This prudence is important for controlling costs quarter by quarter, but sacrifices long-term evolution in the process.

Grid computing offers a flexible, evolvable architecture, and zSeries brings rich functionality to a grid. With the use of larger, partitionable nodes and z/VM virtual-

[8] A data grid distributes content, not workloads.

[9] http://www.platform.com

[10] http://www.datasynapse.com

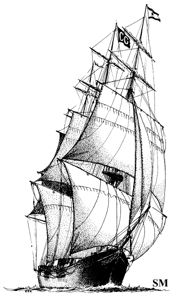
[11] Without the benefit of avoiding the software pricing advantage of an all-Linux (IFL) processor.

ization, how you determine the "cost" of computing services changes. Particularly with Linux, zSeries may be an affordable alternative to herds of dedicated grid resources. **Keep your mind open to the possibilities. Don't abandon the ability to evolve – or limit how you think your enterprise can use a grid.**

Conclusion

IT operations are a long *campaign*, not a *camping* experience. **It is not about how little equipment you can deploy in the short term, but about determining what you need to fully achieve your objectives, and planning to deploy the assets that achieve this most effectively, at the least total cost.**

If you need the capabilities of a zSeries, leveraging the ways you use its virtualization with grid architecture can decrease its TCO to meet current budgetary stringencies. If you do not have a zSeries, what it can do for an enterprise grid may be reason to consider buying or leasing one. **If zSeries and grid architecture together make your IT environment do all of what your enterprise needs it to do, measurable and immeasurable benefits will more than justify its use.**



About The Clipper Group, Inc.

The Clipper Group, Inc., is an independent consulting firm specializing in acquisition decisions and strategic advice regarding complex, enterprise-class information technologies. Our team of industry professionals averages more than 25 years of real-world experience. A team of staff consultants augments our capabilities, with significant experience across a broad spectrum of applications and environments.

- ***The Clipper Group can be reached at 781- 235-0085 and found on the web at www.clipper.com.***

About the Authors

Mike Kahn **is Managing Director and a cofounder of The Clipper Group.** Mr. Kahn is a thirty-year veteran of the computer industry. For the vendor community, Mr. Kahn specializes on strategic marketing issues, especially for new and costly technologies and services, competitive analysis and sales support. For the end-user community, he focuses on mission-critical information management decisions. Prior positions held by Mr. Kahn include: at International Data Corporation — Director of the Competitive Resource Center, Director of Consulting for the Software Research Group, and Director of the Systems Integration Program; President of Power Factor Corporation, a Boston-based electronics firm; at Honeywell Bull — Director of International Marketing and Support; at Honeywell Information Systems — Director of Marketing and Director of Strategy, Technology and Research; with Arthur D. Little, Inc. — a consultant specializing in database management systems and information resource management; and, for Intel Corporation, Mr. Kahn served in a variety of field and home office marketing management positions. Earlier, he founded and managed PRISM Associates of Ann Arbor, Michigan, a systems consulting firm specializing in data management products and applications. Mr. Kahn also managed a relational DBMS development group at The University of Michigan where he earned B.S.E. and M.S.E. degrees in industrial engineering.

- ***Reach Mike Kahn via e-mail at Mike.Kahn@clipper.com*** **or via phone at** ***781-235-0085 Ext. 21. (Dial "1-21" when you hear the automated attendant.)***

Anne MacFarland **is Director of Enterprise Architectures and Infrastructure Solutions for The Clipper Group.** Ms. MacFarland specializes in strategic business solutions offered by enterprise systems, software, and storage vendors, in trends in enterprise systems and networks, and in explaining these trends and the underlying technologies in simple business terms. She joined The Clipper Group after a long career in library systems, business archives, consulting, research, and freelance writing. Ms. MacFarland earned a Bachelor of Arts degree from Cornell University, where she was a College Scholar, and a Masters of Library Science from Southern Connecticut State University.

- ***Reach Anne MacFarland via e-mail at Anne.MacFarland@clipper.com or at (781) 235-0085 Ext. 28. (Dial "1-28" when you hear the automated attendant.)***

Regarding Trademarks and Service Marks

The Clipper Group Navigator, **The Clipper Group Explorer**, **The Clipper Group Observer**, **The Clipper Group** *Captain's Log*, and *"clipper.com"* are trademarks of The Clipper Group, Inc., and the clipper ship drawings, *"Navigating Information Technology Horizons"*, and *"teraproductivity"* are service marks of The Clipper Group, Inc. The Clipper Group, Inc., reserves all rights regarding its trademarks and service marks. All other trademarks, etc., belong to their respective owners.

Disclosure

Officers and/or employees of The Clipper Group may own as individuals, directly or indirectly, shares in one or more companies discussed in this bulletin. Company policy prohibits any officer or employee from holding more than one percent of the outstanding shares of any company covered by The Clipper Group. The Clipper Group, Inc., has no such equity holdings.

Regarding the Information in this Issue

The Clipper Group believes the information included in this report to be accurate. Data has been received from a variety of sources, which we believe to be reliable, including manufacturers, distributors, or users of the products discussed herein. The Clipper Group, Inc., cannot be held responsible for any consequential damages resulting from the application of information or opinions contained in this report.