THE CLIPPER GROUP
Navigator™
Navigating Information Technology Horizons℠
Published Since 1993 Report #TCG2009022RLI April 23, 2009

System z as a Cloud for Business Services — Available Today to Meet the Needs of Tomorrow

Analyst: Anne MacFarland

Management Summary

The costs of leveraging technology have continued to fall for the individual. For a business of more than modest size and ambitions, they have not. New ways to leverage technology for better relationships with customers and partners abound, but none is inexpensive on a large organizational scale. None is simple. Cloud computing, as a form of contracted sourcing, offers a way for both vendors and customers to leverage technology in an incremental way.

Most clouds have deployed *share nothing architectures*, which are a good match for micro services and subscription business models. They are a less good match for the way most businesses of size do computing, which has often been described, from a network point of view, as a *hairball*. To recreate this interaction, businesses are seeking clouds that can be federated, secured, and aggregated into a more useful whole. As CEO Paul Maritz of VMware recently put it, "*We want to build a software mainframe.*"

For IBM *System z* owners, this should be a wake-up call to take a fresh look at their mainframes as the place to host enterprise clouds. Many customers have been adding more workloads onto System z, leveraging both the control systems that arbitrate its *share-all architecture*. Recent mainframe models, designed to provide twice the compute power as their predecessors with no increase in energy draw, underlie the new economics of long-term mainframe computing. **At the large scale that is the new reality for many enterprises, the *share-all* of a zCloud becomes less expensive, per service supported, than the commodity server scale out of most clouds, whether internal or external.** This is not a competitive statement, but a comparative statement. A different mode of computing is involved.

Like the NASCAR racer that tests the limits of what is possible, **the IBM mainframe is able to offer cloud capabilities far beyond that of a basic, share-nothing cloud**. It is a good test-bed for more interactive-process clusters, just as a racer is a good test bed for anti-lock brakes that now are standard in ordinary cars. The granularity of the mainframe's virtualization, its shared everything architecture, its memory controls, its internal communications structure, its specialty engines, its energy efficiency and its accommodation of Linux and Java natively are all cloud-relevant, and enable System z to be a natural *zCloud*. **If you have a mainframe for back-end interactive (even, maybe, collaborative) workloads, it makes eminent financial sense to put your more isolated and more prosaic workloads on zCloud as well.**

Additionally, for those who have not considered a mainframe in the past, this may be the time to do so. The way that you assess computing options should not be mired in last-generation notions of what was possible or appropriate. For a look at zClouds, and why you should want one, please read on.

IN THIS ISSUE

- The Cloud Dilemma and the Mainframe Solution 2
- Basic Cloud Requirements 2
- The *More* that Business Wants in Clouds 5
- zCloud Use Cases 7
- Conclusion 9

The Cloud Dilemma

The dilemma that clouds face is how to achieve operational simplicity while adequately meeting the rapidly-escalating needs of users. The flip side to the advantages of short-term commitments is the ease of disengagement. In highly dynamic markets, businesses represent a large, stable, and attractive customer base – but they come with rules and regulations that impose new requirements on the clouds they patronize. The challenge is most obvious in the area of security. By its nature, a cloud architecture introduces many more handoffs, each of which is a point of vulnerability.

The share-nothing deployments of today's self-proclaimed clouds are comparatively simple. Think of family members, each with their own TV, getting exactly what they like – but never sharing the experience. Businesses may have many role-specific needs that can be satisfied by share-nothing approaches, but they also have a need to leverage information about business operations more widely – particularly as a business or operation grows beyond its modest beginnings.

Since business services seldom stay simple, serialization (the litany of process) can introduce single points of failure unless redundancy is built into every layer in the hardware and software stack. This serialization is reduced when services are more loosely coupled and collaboration is done by arbitration. A failed service falls out of the pool and is replaced by another instance, but that kind of collaboration also comes with the challenge of how to manage and secure the resulting mish-mash of applications and data, without building out an infrastructure that is either unmanageably complex, inflexibly policy-bound, or both.

System z is architected for *share-everything multi-tenancy*. This capability, built over years, and based on knowledge of and collaboration between mainframe components, can meet these and other challenges. As a platform, it is, by its nature, highly redundant, with built-in resilience. It has more dimensions of extensibility, and supports both high utilization and high throughput by tight coordination of processors and memory (jointly evolved, over decades, to work together) connected by a very fast internal network. This support for multitenancy has persisted as the mainframe has evolved to meet contemporary needs because it corresponds with how work is done at the larger businesses that tend to use System z. **While this design was dictated by the resource constraints of when it was invented, isolation, prioritization, and control never go out of style. It is a style that clouds need. System z has it today.** (See Exhibit 1 on the next page.)

The challenge to potential cloud users is to figure out what qualities underlie the services they would like to purchase and use via a cloud. Businesses must differentiate between the *good enough* that is *truly satisfactory* and the *good enough* that is *short-lived*. Short-lived commitments are the bane of business, for organizational change, by its nature, is beset by ramifications that take time and effort to resolve.

Basic Cloud Requirements

By now, people are becoming familiar with basic cloud attributes[1]. They have seen them in action, and they have seen some clouds falter, usually because they were insufficiently architected. They have recognized precursors in what was called *grid computing*, not too long ago.[2]

Cloud's basic requirements boil down to five.

1. *Scalability* involves the ability to grow large while functionality remains undiminished.
2. *Resilience* is the ability to keep going when infrastructure elements fail.
3. *Elasticity* is the ability to add shares that support a service without disruption operations and the ability to add resources to support a larger service.
4. The ability to support a service from onset to termination is often called *support of the service lifecycle.*
5. Finally, a cloud must support *basic security*.

Each will be discussed.

Scalability

A cloud must be able to scale to large proportions quickly to meet customer needs. Since cloud customer demand usually is not specified by a long-term contract, it is not predictable, particularly in the early months of operation. This is particularly true for so-called *public clouds*, where enrollment is unrestricted. Of course, even public clouds could restrict the availability of additional capacity, if they had to, due to extraordinary

[1] For a discussion of four different perspectives, or dimensions of cloud computing, see also the issue of Clipper Notes entitled *Understanding the Dimensions of Cloud Infrastructure in Order to Harvest the Benefits*, dated April 20, 2009 and available at http://www.clipper.com/publications/TCG2009021.pdf.

[2] For a look "back to the future" on grid computing as a precursor to cloud computing, see the issue of Clipper Notes dated May 17, 2007, entitled *All Nodes Are Not Equal - Thinking Differently About Grid (as We Used to Know It)*, and available at http://www.clipper.com/research/TCG2007064.pdf and **The Clipper Group Navigator** dated march 31, 2003, entitled *Using zSeries as a Grid Server - Many Unexplored Possibilities for the Enterprise* and available at http://www.clipper.com/research/TCG2003012.pdf.

Exhibit 1 — Why zCloud Makes Sense — A Summary

A More Intimate Kind of Resource Sharing

***Share more* is inherently a better use of resources than *sequester more*.** System z supports more kinds of more generous sharing than other computing approaches.

The Universality of Linux

Linux provides a commonality that is supported across all server platforms. It is the lingua franca of opportunistic computing. Unlike applications running under z/OS, Linux, even on the mainframe, must be clustered to achieve high availability. Because of the universality of native Linux implementations, Linux on z can cluster with Linux on other platforms, like x86-based servers. So, in a zCloud, sharing is also extensible.

More Granular, Pervasive and Integrated Virtualization

Both LPARs (logical partitions) and z/VM provide isolation of images and support server consolidation. The fine granularity provided by z/VM[1] and the vast extensibility and sharing provided by the Integrated Coupling Facility (ICF) lets z support the share-nothing of traditional scale-out open system cloud operations as well as the integration needed to support complex back-end business processes with fewer resources, less floor space, and less energy.

zEconomics

Mainframe acquisition is more complex because what it will cost depends on how you plan to use it, what you already have installed and operational, and on the nature of the aggregate of all the workloads you plan to run on it. This is a complex calculation.

It is important to look also at the costs that can be reduced by the use of a mainframe. These benefits come in an aggregative litany.

- The granularity of virtualization and management elements built into the System z platform allow more workloads to be placed on fewer servers. The advantage over virtualization on x86 is in the range of up to a hundred times as many Linux instances on a processor/engine. The very full utilization that the mainframe delivers via a myriad of firmware features and service processor functionalities is far higher than the utilization on other platforms – lets be generous and call non-z utilization 33% utilization. That adds another factor of three to the cost amortization. These capabilities produce tangible cost reductions – a mainframe takes 1/25th of the floor space of comparable workload support on x86 servers. The co-location of functionality within a box also reduces the cabling and switches that must be deployed. (As the configurations of scale-out environments vary, the savings here are left as an exercise for the user.)
- In recent years, mainframe design has doubled processing capacity without increasing energy use. Customers track mainframe energy use at one twentieth of the energy used by x86 systems for an equivalent aggregation of workloads.
- Mainframe specialty engines, the engines that are priced lower than general purpose engines, also have changed mainframe economics, particularly for Java applications (zAAP offloads all the Java processing) or information intense workloads (zIIP receives and executes offloaded database and XML parsing) and Linux applications using IFL (Integrated Facility for Linux).. The icing on the cake is that, for existing customers that upgrade their mainframes to a new model, these specialty engines are upgraded (with additional capacity) at no additional cost. The last few generations of engines have more than doubled their processing capacity. A no-cost upgrade becomes a highly attractive value proposition, sort of a technology dividend that keeps on paying.
- Because of the control systems built into the mainframe, and because of the fewer elements that have to be managed due to the extreme consolidation the mainframe supports, customers say a mainframe can be managed with one fifth of the staff that is needed for equivalent workloads running on x86 servers.

The technology platform that supports cloud computing is a major operational expense. Operational overhead costs are, if not death, debilitation by a thousand recurring cuts. Purchase cost is but once, and can be amortized over the operational life of the equipment – which, in the case of the mainframe, can be anticipated to be many years. Even if you take these customer-reported benefits with a grain of salt, or several, the aggregate savings are something to consider.

circumstances, but it would diminish their attractiveness.

Today, there is a variety of scale-out cloud platforms. Some are tuned for particular workloads, like analysis of vast data, or support for two-stage commit transactions, or specialties, such as post-production video editing.

Scalability may be limited by how an application is written. It also may be limited by application dependencies. Service-oriented architecture (SOA) practices address these limitations by breaking cumbersome applications into self-contained, self-describing components that may be replicated, as needed. Large applications are now written this way.

Enemies of scale also include *latency* and *chattiness.* In distributed computing, the latency comes, not anymore from limited bandwidth, but from simple physics. Chattiness comes from the serialization mentioned earlier.

The mainframe is, in essence, a self-contained cloud, with full redundancy and hot-swap capabilities. It is scalable in many ways. The ability to support more workload instances on a basic set of resources satisfies the need to support unimpeded growth. System z options for capacity on demand support another kind of scalability. The options include such niceties as capacity for a planned outage – something that can make major upgrades not only transparent to end users but also CapEx-neutral. Unlike some other capacity-on-demand schemes, with the mainframe, you can turn the capacity off when it is not needed, decreasing the energy draw and operational costs. This contributes to its high qualities in the next basic category – resilience.

Resilience

A cloud must be resilient both in the eyes of the consumer of services and of the cloud purveyor. In the past, resilience would be achieved by clustering physical servers. In scale-outs of commodity boxes, clustered cache (in-memory) resilience keeps the user experience of the application intact when boxes fail, and keeps deployment costs for large scale-out deployments down.

In deployment of inexpensive hardware, often shorn of redundancies to keep costs down, resilience is achieved by quick retargeting of spare instances of application elements, facilitated by virtualization. This strategy jibes with the American passion for convenience – which is exactly what cloud consumers are looking for. It can also come with an overhead of more change to be managed and the by-product of more failed technology elements to be removed and responsibly recycled.

The business side of cloud, and its funders, see things differently. For them, resiliency is best provided transparently and in a manner that will sustain the cloud over the long term at low cost. Only then can margins for services be kept adequate to support profitability in a highly competitive market. In most business situations, revenue from ads will not be tolerated as a means of economic support.

System z provides resilience at all levels, from its operating environments, to firmware to highly-redundant hardware, to its Parallel-Sysplex enhanced architecture. It is built into the system – the platform – at several levels, and, for the most part, this does not incur separate licensing charges. z/OS does add stringent controls that reinforce and augment system resilience. Of course, that comes at an extra charge[3] – but in business situations involving sensitive data, risk avoidance must be part of standard operating procedures.

Elasticity

Cloud elasticity is a matter of management finesse. The primary tool for this is virtualization, whereby resource capacities are amassed into pools that flexibly and quickly satisfy the needs of the workloads that draw on them. Additional tools, like orchestration and change management, are needed to manage cloud operations, and rules are needed to automate their management. If the elasticity is supported by niche-bred tools, the licensing and staffing costs may be higher than if, as **on the System z mainframe, the virtualization and resource allocation and automation are built into the platform, executing according to the QoS priorities of the workloads.**

Service Lifecycle Management

A significant virtue of a cloud approach to computing is its support for evolution. Everything is presented in terms of a service for a finite term. With this declarative service orientation, frequent changes, both as the deployment of new services and as service retirement,[4] becomes easier to support.

---

[3] The expense of z/OS is due to its complexity and the costs of evolving it to meet current needs. Such costs were the reason Sun chose to open source a basic version of Solaris. The skill sets needed to evolve a fully mature and complex software product like z/OS would make the education of a crowd to the point that they could participate in its evolution a large challenge. It would also not be in the best interests of IBM or its z/OS customers. IBM's Academic Initiative today is focused on teaching the skills to leverage the controls and capabilities of the mainframe to application developers and system administrators.

[4] The process of launching a service is the basis of developing a process for its retirement. The connection between two services is, itself, a fully documented service.

IT service management is a well-honed element of IT management, but it was created in an era of greater predictability, a narrower range of requirements, and a focus on keeping the hardware infrastructure up and running.

Redefining the Scope of the Lifecycle

The concept of cloud extends traditional service lifecycle management beyond physical infrastructure to include applications and access to them. It extends the domain of concern beyond the data center out to the user receiving the service. This extension requires the integration of business process modeling as the basis of coordination of application services, in order to add more deft anticipation of demand. Recasting IT into armadas of small ACID[5] services also puts the burden of efficiency and effectiveness on their well-timed orchestration. You have a choice.

Consider the Scope of the Service

Orchestration is easier when the scope of cloud services is extremely limited. Early cloud efforts have supported a singular, targeted service such as storage, content management, or support for remote workers or desktops. **For business use, a limited-scope cloud may be adequate in a number of situations, but not sufficient for avoidance of operational risk. Some kind of federation of services will be necessary.** This may be an exchange of heartbeats, or exchange of data, or tighter coordination of full-scale clustering. Adequacy of federation is often more than just optimized communication or data transmission. **The share-everything approach of the mainframe welcomes managed federation.**

Service Collaboration

Collaboration between application services is a key part of service-oriented architectures (SOAs). It also complicates security and begs for end-to-end encryption. It can expose a higher-level risk by exposing the cadence of business operations, unless the entire domain of operations is securable as a unit. **As a self-contained cloud, System z has an edge in addressing all of the risks added by deconstruction.**

Secure Service Support

Basic cloud security must protect each service, and each element supporting the service, from unauthorized access, corruption, and undue exposure. It must similarly protect the data that the service uses and generates. One of the challenges of federating services is to keep security coverage seamless without complicating the business model and changing the economics. Anyone who has grown a business has seen that moment where the negative ramifications of growth become a significant burden, and yearning for the simplicity of the good old days starts. **Nowhere is this more painful than in the area of operational security and nowhere does the mainframe stand more above other platforms.**

The *More* That Business Wants in Clouds

So far, we have discussed the characteristics of a cloud that is both *deliverable* and *deployable*. Deliverable means that it makes resources available to a group of people. Deployable means that it supports services that make it easy for people to do what they want to do. What more could you want? Plenty!

A business may have additional requirements to make clouds more usable, in a business context. Business life is seldom *as usual* for long, and more is punctuated by a litany of crises and unexpected events. Whatever is finally going well often will cease to be a source of differentiation.

Think of the cell phone. Originally, it was all about making telephone calls and picking the best plan. Now, it is more about the quality of e-mail service and the additional applications that can be hosted on a particular device. Therefore, to be truly useful to business, there are more cloud requirements, in addition to the basics.

Sharable Resources

There is a quality difference between *apportioning resources* and *sharing*. Running Linux natively on System z illustrates the point well. It can be done in three ways.

- Linux can run natively on an engine, which seldom is done because the resources of a mainframe engine are not consumed by a single application.
- It can run on a logical partition or *LPAR* of an engine, which carves out a set of resources for use by an application. LPARs are mostly used for large applications with a predictable need for resources.
- Linux can also be run under *z/VM*, as a guest.

z/VM[6] can support thousands of guests on an engine – or across several engines – including the mainframe's specialty engines. In installations

[5] ACID properties are *A*tomicity, *C*onsistency, *I*solation, and *D*urability.

[6] For still more detail about z/VM, see **The Clipper Group Navigator** dated February 28, 2007, entitled *Oh, the Things You Can Do with z/VM 5.3!*, which is available at http://www.clipper.com/research/TCG2007030.pdf.

where two geographically-remote mainframes are linked by an *Integrated Coupling Facility*[7], the z/VM can span geographies. Unlike other hypervisors, z/VM can also nest guests within a single machine[8]. Because z/VM is licensed by the engine, the cost of each guest is lower than schemes where each guest requires a separate license (a license that also must be managed).

A significant difference in running Linux on z versus other platforms is in the initial resource allocation. On servers based on x86 architectures, it is usual to assign as much memory as the application will ever need – to err on the side of generosity, so to speak. On the IBM mainframe, this is considered a bad practice. Mainframe memory management sees inactive memory as something to be paged out[9], and over-assigning memory can cause paging churn. Instead, the minimum memory should be assigned to Linux z/VM guests, because they can always access more when needed, using this same memory management.

The default on the Mainframe is share more, assign less. This is the same approach as most schemes to save our planet from ourselves. Share more, recycle more, and retain only what you need. It is familiar. It is borne out by IBM's decision to leverage the mainframe in its own data center consolidations, when it had other options available.

Network and Resource Optimization

More broadly, System z's reprioritization capabilities, in the ability to offload workloads to other engines, allow the *more* that is occasionally needed by applications not to stress the system, nor require redundancy for failover.

Because application guests on a zCloud can use the internal *Hipersockets* network, dramatically reducing the use of external networks for communication among applications and databases and cutting the security risks that come with such network externalization, traffic and risks can be minimized. **Because system z's internal resources have been developed over decades to work together, they can meet the needs of high-priority workloads with less duplication and arbitration.**

Process Prioritization

System z juggles workloads by priority. When new workloads are added, they come with specified priorities. That allows all workloads to get the share of resources that they need and deserve. The specialty engines that System z supports, *zIIP*[10], *zAAP*[11], IFL (Integrated Facility for Linux), and *ICF (Integrated Coupling Facility)*, host Linux (IFL), and orchestrate offload (in the case of zIIP and zAAP) or handle sharing databases among mainframes (ICF).

Managed Quality-of-Service Tiers

Tiered offerings underlie the profitability of many business models, including open source software. Managed *quality-of-service tiers* bring better satisfaction to any customer base with a wide spectrum of budgetary constraints.

Managing quality-of-service tiers is different from process prioritization, in that is focused not on the *process* but on the *service* that a process supports. It is a different granularity. It often may include an overlay of enrichment or customization.

Once again, **this is inherently done more easily by a system-in-a-box than by a system of boxes. It is part of the System z heritage.** With farms of commodity servers, tiers of service can be supported, with effort.

Not Merely Integrated but Optimized

Many of the System z capabilities described above, and the way they communicate and coordinate with each other, support optimized application and business operations. The growing number of software vendors providing SOA on mainframe[12] or SOA-including-a-mainframe is indicative of the role System z can play in the optimization that SOA enables. System z can also

---

[7] And Integrated Coupling Facility is a specially-characterized processor more familiarly called an ICF.

[8] This allows a kind of scale-up that is unique. It allows easy deployment and management of larger clumps of composite services that are still integrated by standards-based runtimes. The individual components can still be modified as desired, and deployment of the new composites can be more efficient. Most organizations are not at this point in their use of virtual machines and service oriented architectures, but that future may be coming sooner than you think.

[9] z/VM pages out memory that is not being accessed to a separate swap space, ensuring that it is not corrupted by another application. This permits oversubscription of memory space. It is a different approach than memory management in x86 processors.

[10] For more on zIIP, see **The Clipper Group Navigator** entitled *System z9 Adds zIIP to Ally with DB2 on z/OS to Better Serve the Onslaught of Business Data*, dated January 24, 2006, and available at http://www.clipper.com/research/TCG2006006.pdf.

[11] For more about zAAP, see **The Clipper Group Navigator** entitled *zSeries Zips Through Java with zAAP*, dated April 7, 2004, and available at http://www.clipper.com/research/TCG2004030.pdf.

[12] For specific examples, see **The Clipper Group Navigator** entitled *For a More Secure SOA Use System z And Data Direct*, dated December 3, 2007, and available at http://www.clipper.com/research/TCG2007102.pdf and also **The Clipper Group Navigator** entitled *Build a Super SOA with Sola*, dated May 28, 2008, and available at http://clipper.com/publications/TCG2008027.pdf.

Exhibit 2 — Minimal versus Better Business Cloud

	Minimal Business Cloud	Better Business Cloud
Scalablity	✓	✓
Resilience	✓	✓
Elasticity	✓	✓
Service Lifecycle Management	✓	✓
Secure Service Support	✓	✓
Fully Sharable Resources including Networks		✓
Network and Resource Optimization		✓
Process Prioritization		✓
Managed Quality of Service Tiers		✓
Optimizable, Not Merely Integrated		✓
Auditable Controls		✓
Risk-Oriented Security		✓

be a security hub for wider operations, catching exposures that fall between the cracks of local controls.

Auditable Controls

Even privileged users actions must comply with security policies – and all their actions are logged by *RACF* or equivalent software. IBM's *Consul zSecure Suite* can track resources or users (an application is a user) and can synthesize logs from different systems into a process dashboard that satisfies both compliance needs and those of a business seeking to understand and optimize diverse components of its operations. **System z's approach to managing the whole system serves well the need to manage compliance.**

Risk-Oriented Security

Available to all workloads on the mainframe are certain built-in enhanced security capabilities. The cryptographic co-processor does encryption in hardware, reducing the performance penalty and avoiding the need for a separate appliance that itself can be a source of risk or failure. End-to end-encryption is better than end-to-appliance or end-to-switch.

If there is an instance of z/OS on your mainframe, putting the LDAP directory under its control, not just for applications on System z but also for those on other platforms, via *IBM Tivoli Directory Server for z/OS*, adds more security. The z/OS *Integrated Cryptographic Service Facility* (ICSF) can hold keys on the mainframe but protect the much larger domain of all IT system elements they connect to, either directly or indirectly.

The Minimal–Better Spectrum

Finding your place on the Minimal-Better spectrum of Exhibit 2 (above) is not a snap decision. Many service requirements are not on a check-off list, but come as expectations, or conformance with unstated corporate policies. The penalties for contravening corporate policy may not be immediate or obvious. They may only be seen in the weakening of a system – but it is a system that often spells the difference between profitability and doom.

It should also be noted that not all of these advanced features in the right-hand, "Better Business Cloud" column are available on Linux-only mainframes. Some of them leverage capabilities and/or features on an instance of z/OS. This adds cost, but the cost, when administrative costs are factored in (as discussed in *zEconomics* in Exhibit 1) may be comparable or even less expensive for the software and appliance surround that would have to be added to the system to get the equivalent functionality piecemeal.

zCloud Use Cases

Marist College

Marist College has had IBM mainframes for thirty years. The recent upgrade to a four-engine[13]

[13] Marist College's z9 has one engine dedicated to z/OS, one zAAP. The rest of the engines are IFLs.

System z9 was easy to justify, according to Bill Thirsk, Vice President of Information Technology and CIO, "because of its cost-effective operations and greenness, compared to the several servers that would be needed on other platforms."

Recently, Marist has expanded use of System z to support a variety of initiatives. As well as supporting College operations and administration, Marist supports similar functionality for several other colleges on z/VM guests. It supports college-wide email, the college's web presence, and numerous other applications. It gives a z/VM share to every student who requests one, not just the Computer Science majors, as heretofore. Students use their shares for personal projects, blogs, and wikis.

Most interesting is *Sakai*, an online educational environment that supports courses, collaboration, videos, wikis, podcasts, links to articles, and other organizational resources that complement traditional (academic) college activities at an institutional level. Many full degree programs, including Master of Business Administration (MBA) and master of Professional Accounting (MPA), are offered totally online. The System z supports all of the Sakai functionality.

These activities "have not really pushed his System z9's to full capacity," Thirsk remarked. He is thinking of offering mainframe shares to Marist alums, in the form of lifelong ePortfolios, to keep them in touch with leading edge technology.

A Large Financial Services Customer

A very large IBM customer was faced by the need to accelerate the complex development cycle that underlies their ability to offer new and compelling value propositions to their customers. The applications underlying these value propositions are complex. The development teams are a diverse lot, and orchestration of their efforts must be deft. Time is the enemy.

The project was championed by an individual who had little mainframe experience, but who knew a great deal about the constraints of rules-driven, regulation-constrained operations. He was looking for the test-and-qualification capacity of a cloud solution that also would be a repeatable solution that could accommodate of his future requirements.

It had to accommodate meticulous adherence to all of all the particulars (conventions, certificates, etc.) that underlie the applications that execute in highly regulated financial environments. It had to cope with the specifics that infest IT elements. Thus, adherence to standards and standard practices become the way to navigate the inevitable complexity.

It had to support an environment where almost everything came in multiples – operating systems, guests, instances, databases, application servers, HTTP servers. The only constants were authentication and security. Complexity is easy to test in isolation, although it can be time-consuming, if it cannot be parallelized and automated. The automation of many objects and processes is in itself challenging in many dimensions, said the customer spokesperson. Reuse was an obvious priority.

For example, the tools used to test had to be not just good, but also designed in a way that could be discovered and examined in order to assure that the testing would be entirely satisfactory (i.e., sufficient and complete). In a complex, rules-driven environment, there is no tolerance for the waste of one-off solutions or incidental, badly-documented capabilities.

The customer decided to use a platform and tools "that were derived from experience." In IBM Tivoli, they found a partner who had the experience to give them the right high quality building blocks to work with. **In System z IFLs running Linux under z/VM, they found an expansive source of instant capacity that did not itself become part of the problem.** Even z/VM was transparent to the testing operations, being one of the hypervisors that virtualizes itself in addition to the resources it controls.

This story of System z, used as a capable, well-appointed basic cloud in a very complex scenario, is very different from the opportunistic and open-ended zCloud at Marist College. For this financial institution, System z met the time objectives, supported complex testing without latency, and, according to the corporate charge-back gnomes, was a cost-effective solution as well.

Hoplon

Hoplon Infotainment confounded the world by running its *Taikodom* massively multiplayer online game (MMOG) on IBM System z and IBM Cell processors.[14] Hoplon now offers a development platform, *Bitverse World Builder*, the *BitVerse Runtime*, and Eclipse-based tools and libraries, for design and software development.

Hoplon's business model has expanded, driven by customer demand and its investor funding. It will use its *Bitverse* environment, still running on IBM System z and Cell processors, for more

[14] For more information on Hoplon's deployment see **The Clipper Group Observer** entitled *A Hybrid Solution for Xtreme Information Use – Hoplon Leverages IBM's Cell/B.E. and System z*, dated May 22, 2007, and available at http://www.clipper.com/research/TCG2007065.pdf.

than gaming. Initial pilots have proved the value of this expansion. Hoplon plans to use *Bitverse* to provide 3D environments for training, collaboration, and social interaction, as well as for simulations of product use and business situations.

Hoplon's experience offers yet another, quite different view of a zCloud, as the enabler of new forms of business success.

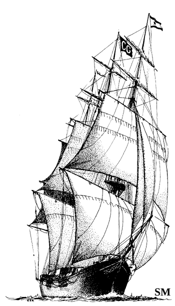
Three Very Different zClouds

All three customer examples are compelling in their own contexts. All leverage both heritage and innovation to provide compellingly attractive services that both drive and support business innovation. **Just as there is no singular definition of cloud that fits all uses, there is no limit to the nature of clouds on System z mainframes.**

Conclusion

The value of cloud computing, in the eye of the beholder, is the availability of technological support that is always available, comprehensively useful, and attractively priced. If you are charged with providing that kind of technological support, it is time to be thoughtful and prudent. It is time to consider all you options. Strong market forces are at work. Competitors are only a click away and more findable than ever. **Choose the solution that best fits what you need to do.**

By the modern definition of cloud computing, System z has been an internalized cloud for decades. It has groomed its control functions to support sophisticated sharing that supports complexity in a way that is lean (in resource consumption) if not simple. **When considering cloud computing, remember to consider the IBM System z mainframe. If you already have one at the center of your mission-critical enterprise, you already know the advantages.**



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 Author

Anne MacFarland **is Director of Data Strategies and Information 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. 128. (Please dial "128" 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*, **The Clipper Group Voyager**, **Clipper Notes**, 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.

ZSL03018-USEN-00