Title: The New IBM zEnterprise Systems and Testing

Mike Duron
IBM Corp
Systems Bringup and Integration
Austin, TX, USA
e-mail: duron@us.ibm.com

Nancy Finn
IBM Corp
System z Core Technologies
Poughkeepsie, NY, USA
e-mail: duron@us.ibm.com

Paul A. Wojciak
IBM Corp
RAS Test Architect Hardware Test
Poughkeepsie, NY, USA
e-mail: wojciak@us.ibm.com

Abstract:

zEnterprise™ and zEnterprise BladeCenter® Extension solve enterprise customer business problems through innovative product capabilities. Many challenges were faced and overcome during zEnterprise integration and testing. A structured yet flexible test approach was central to that success. IBM® labs and client facing organizations throughout the world collaborated on zEnterprise from test through deployment in customer datacenters. The result is a high quality enterprise ready offering.

Part 1: Introduction and Background

zEnterprise™ systems represent an integration of technology components from the enterprise and distributed computing worlds. Enterprise Information Technology departments have wrestled with the challenges of data center management across data silos for years [1] [2]. Keeping up with technology options and effectively integrating them into a 24x7 operation requires powerful yet flexible tooling. Bringing together enterprise leadership characteristics such as availability and security with distributed compute resources in the zEnterprise BladeCenter® Extension (zBX), a tightly integrated heterogeneous compute platform is realized. The scope of platform resources being managed is described as an ensemble[1]. See Figure 1 zEnterprise ensemble with one Central Processing Complex and zEnterprise BladeCenter Extension [7] for an ensemble supporting enterprise applications. Under Unified Resource Manager (zManager) control, the total solution enables management of System x® and POWER7® compute resources through the system z® management interfaces. For customers deploying multi-tier enterprise solutions with z hosting one or more tiers, zEnterprise brings all compute tiers under the z management purview.

zManager integrates network and storage resource deployment for POWER7 AIX® and System x Linux® and Windows® hosts within the zEnterprise systems. Inclusion of Websphere® DataPower® Integration Appliance XI50 for zEnterprise into the solution becomes an integrated step of the deployment process. Through zManager, definition, configuration, monitoring and maintenance tasks of zBX are brought together under a common user interface. A number of views are provided to visualize the physical and logical resources of all compute tier components in the solution. These views support operations management of associated infrastructure components alongside system z resources. The concept of workloads provides visualization for resources aligned with business objectives.

[1] An ensemble defines the scope of platform management and consists of a collection of one or more members. Each member is a IBM zEnterprise 196 (z196) or IBM zEnterprise 114 (z114) with an attached IBM zEnterprise BladeCenter Extension (zBX). zBX attachment is optional.

zManager enables policy driven performance monitoring and management of those workloads. Taken together, the rock solid foundation of zEnterprise and flexible management capabilities of zManager extended to include distributed compute resources results in more efficient and better performing operations in the data center.

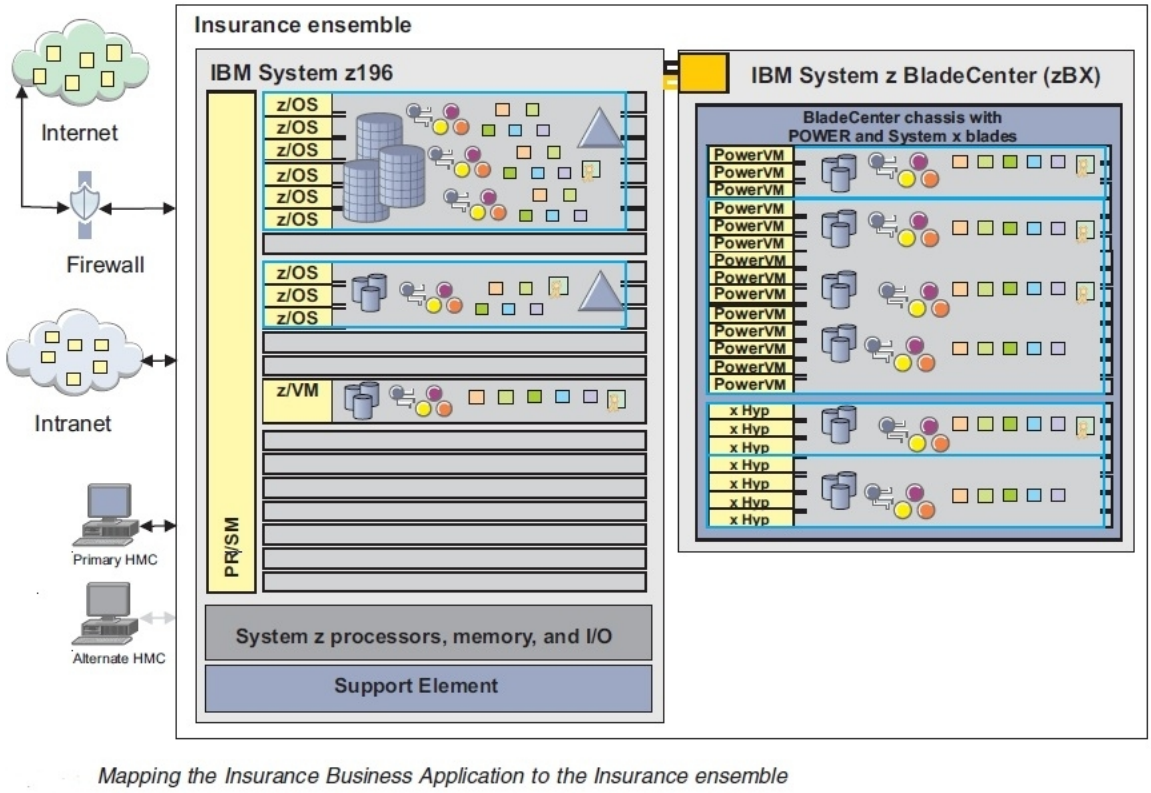


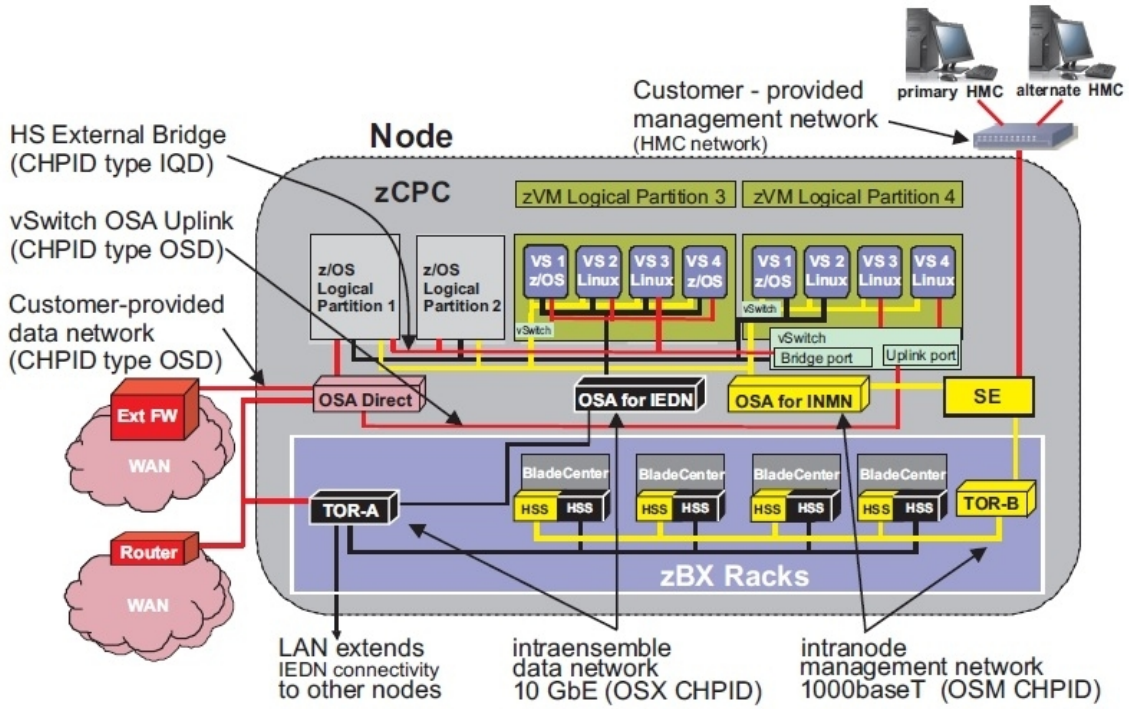
Figure 1 zEnterprise ensemble with one Central Processing Complex and zEnterprise BladeCenter Extension [7]

From a system test perspective, bringing disparate hardware and software components together under a common management umbrella to meet enterprise class expectations presents a unique challenge for the test team. However, that was the test requirement for zEnterprise. BladeCenter is built from commodity hardware to meet affordability objectives. Reliability of commodity components will be less than components engineered to meet higher mean time between failures[2]. Servers constructed from commodity components with lower reliability depend on redundancy in order to have higher availability. In the case of BladeServers®, this redundancy for availability often involves having extra blades configured for workload continuity in the event of a failure. In the BladeCenter Extension, critical chassis components are redundant such as power supplies, cooling, network and Storage Area Network (SAN) switches, and service processors. This improves the overall availability of BladeServers in the BladeCenter Extension.

Extensive testing of the BladeCenter Extension networking infrastructure and capability occurred within the IBM labs. Several different test environments were required to cover the possibilities given their complexity. See Figure 2 zEnterprise ensemble scoped networks [7]

[2] Mean Time Between Failures for repairable and nonrepairable items is calculated from the item's failure rate, λ, where λ = Number of Failures / Cumulative Operating Time. MTBF = 1/λ = Cumulative Operating Time / Number of Failures [3].

for a general view of the networks. A private management network[3] serves to interconnect the BladeCenter Extension to the zEnterprise management infrastructure. This creates a control path for operations and life cycle management as well as a monitoring and reporting path for any error arising from the attached BladeCenter Extension hardware. Availability of Ethernet networks in the BladeCenter Extension is achieved through redundancy as in other areas. Through the Open Systems Adapter (OSA) INMN interface, BladeCenter Extension errors are recognized, isolated then called home. Network communication between zEnterprise hosts and virtual servers within the BladeCenter Extension occurs over the Intra-Ensemble Data Network (IEDN). All data network connections are configured with zManager and constantly monitored by the zEnterprise maintenance infrastructure. By attaching an external router to the IEDN switches or by using z/OS routing services, connectivity from outside the ensemble to the secured network environment of the zEnterprise BladeCenter Extension is achieved. Either method affords the customer zEnterprise monitoring and error handling of those connections. This centralized approach to network error handling improves upon traditional per chassis distributed error reporting in stand alone BladeCenter environments. Orchestration of network set up and monitoring through zManager and service via zEnterprise provides enterprise class capability for the BladeCenter Extension networks.



Networks in an ensemble with an vSwitch Uplink to a customer-provided data network

Figure 2 zEnterprise ensemble scoped networks [7]

Storage attachment to the BladeCenter Extension occurs through redundant 8 Gigabit fiber channel switches. See Figure 3 Example of storage attachment to zEnterprise and BladeCenter Extension [7] for a SAN example. zManager facilitates the configuration of those storage resources and their assignment to hosts within System x and POWER7 BladeServers. The SAN connectivity health is constantly monitored and errors are reported. Error reporting from SAN resources occurs via the host operating systems running on

[3] An intranode management network (INMN), sometimes referred to as a private service network, is required for platform management within an ensemble.

BladeServers or via direct detection by the zEnterprise maintenance infrastructure. SAN environments typically consist of products from several vendors. The BladeCenter Extension supports interconnection of many different storage products from IBM and non-IBM vendors. In most cases, the existing storage infrastructure within the datacenter can be connected to and exploited by the different BladeServer System x and POWER7 hosts. Within the IBM labs testing was performed with a number of different attached storage devices.

SAN environments may sometimes yield surprises when certain products are interconnected. There can be firmware level dependencies for proper interoperation and these must be matched with software device driver levels. Attachment solutions for such cases have consistently been found for BladeCenter Extension SANs. Use of SAN Volume Controller to serve as the attachment point between devices and the System x and POWER7 hosts can overcome situations were vendor interconnectivity questions arise. The scope of zManager control extends from the operating systems in zEnterprise partitions across all hosts within the BladeCenter Extension. By defining ensembles, the scope of zManager control is established. CECs (Central Electronics Complex) within the ensemble get the benefits of zManager management and monitoring. Those with the BladeCenter Extension attached get additional capability for those BladeCenter hosts. zManager enables a rich set of performance monitoring and control capability. Further, energy management functions can be used to optimize power consumption and performance. POWER7 and z/VM (z/Virtual Machine) guests hosts take the greatest advantage of performance controls through goal based profile settings[4]. zManager Platform Performance Manager (PPM) orchestrates the dynamic allocation of POWER7 and z/VM host resources based on those goals. The more extensive the virtualization of hosts, the more valuable is PPM and the functions provided by zManager.

zManager integrates zEnterprise CEC virtualized resource definition, operation, and error handling of guests and their resources under z/VM. An integrated view of all guest resources across z/VM is provided by zManager. Through this view all members of the shared ensemble infrastructure can be visualized whether they are primary hosts and guests on z or whether they are AIX, Linux or Windows hosts on BladeServers. Thus, with the zEnterprise BladeCenter Extension, the management and coordination of hardware and firmware installation, configuration, change management, and maintenance are integrated with the rigorous system z enterprise approach for all ensemble members. Further, the visualization and control of all virtual hosts across that ensemble is made possible through the single zManager user interface.

[4] Performance policies describe how virtual server resources should be allocated to meet the compute demands of the customer's workloads. Policy goals define the relative importance of software applications across the different parts of a customer's business. Each policy goal specifies velocity, how fast work should be dispatched when compute time is available, and business importance, the importance of that application with respect to all others. Unified Resource Manager uses these values to govern dispatching of work across the virtual servers.

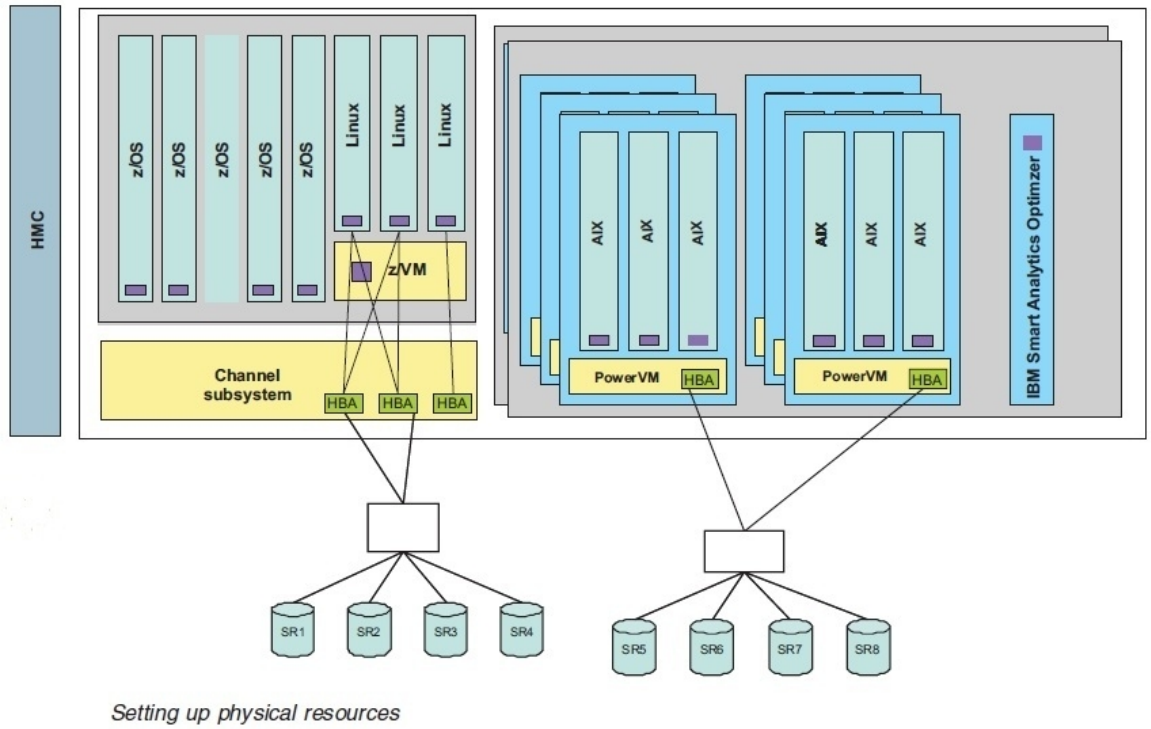


Figure 3 Example of storage attachment to zEnterprise and BladeCenter Extension [7]

Part 2: Meet the Test Approach

zEnterprise with the BladeCenter Extension joins together enterprise class and cost optimized computing. The test approach applied enterprise level test expectations to all components of the zEnterprise solution. System z customers expect the highest availability from their compute solution investments. Typical mean time between failure of system z hardware is measured in decades [6]. System z is designed for 24x7 operations providing the highest availability to our customers' workloads. When designing a test to validate 24x7 availability, one needs to understand all the types of errors that can occur. The system design for error handling must also be understood since this can depend on system configuration and state at the time of error. A great deal of emphasis is placed on maintenance concurrency on system z as well[5]. Tests of serviceability must be included to verify that system availability is upheld by concurrency of maintenance actions. Taken together these actions provide a high level view of Reliability, Availability, and Serviceability (RAS) testing.

One common way to select tests is to derive them from the specification. Test literature will refer to this as a functional test selection method or black box testing [4]. For the zEnterprise BladeCenter Extension, test requirements were derived from high level design documentation. Test plans were designed to meet the test requirements and articulate the criteria for each test case's success. The tests covered all aspects of the solution from basic BladeCenter chassis and infrastructure components to zManager ensemble operations and virtual server lifecycle support to ensembles with realistic customer workloads running across

[5] Concurrent hardware maintenance refers to the ability for a servicer to repair a part in the computer while that computer and associated operating systems and workloads remain at runtime and remain operational.

several zEnterprise and BladeCenter Extension systems. Testing availability across all these components required inclusion of error handling tests from all areas of the solution. See table Table 1 zEnterprise BladeCenter Extension Availability Test Areas for an example of some tests derived from the specification. Test techniques are based on system z RAS test experience. One can infer from the table that to verify availability a number of test techniques are needed to inject errors.

Table 1 zEnterprise BladeCenter Extension Availability Test Areas

Design Area	Specification Derived Availability Tests	Test Technique
IEMN, IEDN, SAN	Verify redundancy keeps networks alive during errors, code update, and repair.	Cable pulls, soft error injection for error simulation, use defective devices
Chassis Power and Cooling	Verify redundancy keeps BladeServers alive with no thermal events. Verify that repairs are concurrent.	Stop fan rotation, disconnect AC input, use defective components
BladeServer	Verify error handling, reporting, and repair. Verify availability provided by redundant BladeServers within a workload.	Soft error injection for error simulation, use defective components
Support Processors and ensemble Hardware Master Console (eHMC)	Verify redundancy provides availability during error handling and code updates.	Manually reset support processor, create loss of connectivity

While product specification derived tests served to produce the majority of the test plan content, there was a need to include additional testing based on the actual implementation. Such tests are called structural tests or white box tests [4]. BladeServer installation, change management, and the communication interface between service processors were areas that needed implementation cognizant testing. Installation of BladeServers in BladeCenter Extension is accomplished with the entitlement process[6] [7]. The reliability of entitlement was dependent on many factors. BladeServer hardware reliability, communication path integrity between the BladeServer and the zEnterprise Support Element (SE), and specific load on the SE from other tasks during entitlement being the main contributors to entitlement failures. Observation of these results led the test team to look at how the entitlement code was written to handle unexpected events. Additional tests were written that covered specific code paths related to failure handling. The reliability of entitlement was greatly improved even in cases of intermittent errors from the hardware or temporary communication loss between the SE and BladeServers.

The same error events occurring during BladeServer entitlement affected change management of the firmware within BladeCenter chassis components and BladeServers. The change management code on the Support Element orchestrates firmware updates over the IEMN. During testing in the lab, intermittent errors occurred that caused firmware

[6] Entitlement is performed from the Support Element using the Perform Model Conversion task. BladeServer hypervisor and management firmware are updated during entitlement resulting in an updated, operating BladeServer ready for use.

updates not to complete. Reliable exchange of status updates between the SE and the component being loaded proved to be at fault. The test team made a deeper examination of the code controlling firmware updates. A set of stress tests were devised targeting the intermittent error cases and their handling by the change management code. Improvements soon resulted producing a more robust change management process for BladeCenter Extension firmware.

With the knowledge of both the communication path intermittent errors and the zEnterprise code improvements for entitlement and change management, the test team designed additional coverage for other functions. Tests were introduced targeting status acquisition for visualizing ensemble resources on the user interface since intermittent errors existed with object status rendering. Another set of functions on BladeCenter Extension needing implementation centric tests related to energy management. Periodically, commands from the SE to query or control the energy management state of a BladeServer would fail. SE code was examined and tests were created to cover different failure and state handling of energy management code. This eliminated the energy management unreliability. The preceding examples demonstrate that sometimes testing must take into account the actual code implementation to assure sufficient reliability. In addition, the plan for test must accommodate changes for areas that are discovered to be defect rich [9]. More tests may need to be added to the plan as a result of this discovery. The limitations of cost optimized commodity components in an enterprise compute space are also evident.

There are a number of views of testing that some refer to as schools. Such schools have intellectual affinity and exhibit exemplar techniques, hierarchies of values, and standards of criticism [8]. Our testing approach drew from multiple exemplar techniques. Even with tests based on the specification and tests based on the implementation, additional exploratory testing was required in some instances [4]. Such exploratory testing involves test experts. Interactive exploration tests are performed with the intent of rapidly uncovering defect dense or trouble spots in the system. Past experience rather than test documentation guides the experts to find implementation or even design weaknesses. For the zEnterprise BladeCenter Extension, exploratory testing was performed. Exploratory testing was applied to two areas, z/VM and DataPower.

The z/VM operating system is sufficiently unique and most people with any familiarity at all happen to also be experienced users. The z/VM test team members fit the description of subject matter experts. Members of this test team undertook exploratory testing in addition to executing a functionally derived test plan for z/VM on zEnterprise and managed by zManager. The exploratory testing found that the panels on the zManager interface to defining and configuring the z/VM guest resources needed improvement. Without an extensive specification for those panels available, the testers simply used there expertise at z/VM guest definition and management to guide their improvement suggestions. The result was largely a more intuitive look and feel to those guest definition panels.

Another area that the z/VM test experts explored related to the Systems Management Application Programming Interface (SMAPI) commands supporting zManager. These experts used their experience to configure networking interfaces, virtual switches, and disk resources for z/VM guests. By defining and manipulating many combinations of guests and their associated resources, the experts exposed a number of concerns. Some SMAPI commands simply did not work as intended. Other command sequences were prone to

failure when the guest resources were changed while many guests were operational. The messaging back to the user in failure cases was not at all clear regarding problem cause and resolution as well. The z/VM support in these areas was greatly improved making them usable and reliable. Given incomplete design documentation in some cases, exploratory testing by expert testers proved successful at achieving z/VM quality objectives.

The Websphere DataPower Integration Blade as part of zEnterprise also underwent exploratory testing by the test team. A wide breadth of functions for Service-Oriented Architecture (SOA) applications, Enterprise Service Bus (ESB) capability, data transformation, wirespeed intercommunication and many others are supported by DataPower[7]. DataPower appliances include a rich graphical user interface for configuration and management. A specification was written describing how those functions were to be offered through the zManager user interface. The test team needed to exploit those capabilities in our zEnterprise ensembles. A solution level test environment was constructed that included DataPower Blades. The solution test workloads were capable of functionally driving a broad set of SOA and ESB functions. Test experts exhausted the limits of the specification derived test and began exploring what was possible with DataPower in BladeCenter Extension under zManager. The experts found that user role definition through the user interface was problematic and confusing. Exhaustive tests were applied to bound and properly document the user roles for DataPower. Security being a central premise to DataPower value necessitated that users who were not authorized for certain capability would not inadvertently be granted access. This was found to be the case as zManager mapped all of the prior DataPower user interface functions to the eHMC. The expert testers worked to close these gaps. The result was a secure, well behaved interface with documentation clearly describing the roles of users accessing DataPower within BladeCenter Extension.

Test plan design by specification driven and implementation driven test cases combined with exploratory testing by subject matter experts serves to describe the test approach. There are some additional test techniques that were instrumental in delivery of the zEnterprise BladeCenter Extension. Some of these techniques were driven by necessity. An illustration of these techniques is next, presenting further details of specific challenges and how they were met by the test team.

Part 3: Specific Challenges

Fundamentally, zEnterprise with the BladeCenter Extension comprises multiple software stacks across several compute platforms. With zManager orchestration for all zEnterprise ensemble resources, the testing scope became broader than teams within our test organization had traditionally done. There were a number of systems test teams doing multi-tier work. The work for the zEnterprise BladeCenter Extension and zManager did not readily fit within the bounds of those teams' existing infrastructure, missions, or capacity. As a result, a new solution level test infrastructure and team was put in place. The zEnterprise and zManager solution test team was purposed with creating a customer environment for evaluating the end to end capabilities. This team took on the challenges of scaling the virtual environment towards the defined limits, evaluating the integration of all hardware and code

[7] Wirespeed implies data transfer at the maximum bandwidth supported by the physical interface. For instance, 10 Gigabit network adapters should be able to achieve 1.25 Gigabytes per second data transfer when operating at wirespeed.

components, deploying real workloads modeled after production environments, and doing so based on our relationships with system z enterprise customers.

The team was comprised of subject matter experts across a wide set of areas: networking, SAN, Power systems and AIX, performance management, System x environments, z/VM, z/OS, zEnterprise hardware, and others. The expectation was that the test environment constructed would allow most testers to do their work simultaneously. The breadth of capability being tested suggested this approach but the test schedule to get the work done prior to product availability demanded such sharing. The solution test team worked in concert with test teams from development and traditional system z hardware test to interlock their work. This was crucial in order that solution testing be able to progress while certain product building blocks were still coming to life. A classic sequential development model was followed [5]. This would imply that a solution test effort wait to get started until prior test phases were complete. In order to train the broader solution test team on the zEnterprise BladeCenter Extension and zManager and provide assistance to struggling functional test teams, solution test initially engaged in functional and then simple system level testing. This approach proved invaluable. Solution testers drove the zEnterprise BladeCenter Extension project's success through their engagement in all test phases.

The hardware infrastructure put in place for solution test was extensive. While there was a hardware system test BladeCenter Extension with more physical BladeServers than solution test, no other test environment constructed could rival the extent of solution test's virtualization. See for a view of the solution test environment. There were 1000's of virtual servers deployed across the POWER7, System x, and z/VM environments. Solution test scaling exposed a number of problems that needed resolution. Single BladeServer scaling limits were found and fixed. Testing of zManager definition and control for z/VM guests uncovered problems that had be resolved. The real value of solution test was returned when workloads were deployed.

Solution test had two main workloads that were based on customer related efforts. The workloads were referred to as DayTrader[8] and Bookstore[9]. These workloads model multi-tier heterogeneous compute environments. They both require DB2 backend on the mainframe and some kind of web serving application servers on BladeServer. By deploying these workloads into solution test, validation of the zEnterprise and zManager value propositions for ease, flexibility, and control was possible. While solution test was working through their challenges, the hardware system test team was facing struggles of their own.

Hardware system test team work was previously mentioned in the context of RAS testing, change management, installation and entitlement testing. A recurring theme that may be discerned is when you subject cost optimized hardware to enterprise class expectations your design may have to adapt. This was the case with zEnterprise BladeCenter Extension. Test presumed that components being assembled would come in with certain stability. This proved not to be the case with the BladeCenter hardware and firmware. The BladeCenter Extension development and test team worked closely with their counterparts in BladeCenter. Reliability of firmware updates for BladeCenter had been a problem before for their customers. The z team experienced this immediately when test of BladeCenter Extension

[8] DayTrader is a benchmark application simulating an online stock trading system. http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/index.jsp?topic=%2Fliaag%2Fdcss11%2Fl0wdcs00_dcs72.htm

[9] Bookstore is an internal IBM workload that simulates a multi-tier online Book Retail environment.

was begun. Entitlement of BladeServers was designed to synchronize the blades with latest available firmware on the SE during installation. However, BladeServer code updates proved unreliable resulting in installation failures across hardware test. The BladeServer development and test teams, separate from system z efforts, undertook a massive stabilization effort for firmware updates. The stabilization work focused in making BladeServer firmware updates robust and reliable. In parallel, the system z team improved the robustness of entitlement code since eliminating all failures was unlikely. Besides those system z robustness improvements and the reliability enhancements from system x, a recovery procedure was instituted to allow for intermittent BladeServer entitlement failures during installation to be handled. Code update reliability improved from the BladeServer team efforts and overall system z BladeCenter Extension installation success improved. Such collaboration between the server brands was essential in achieving zEnterprise BladeCenter Extension product objectives.

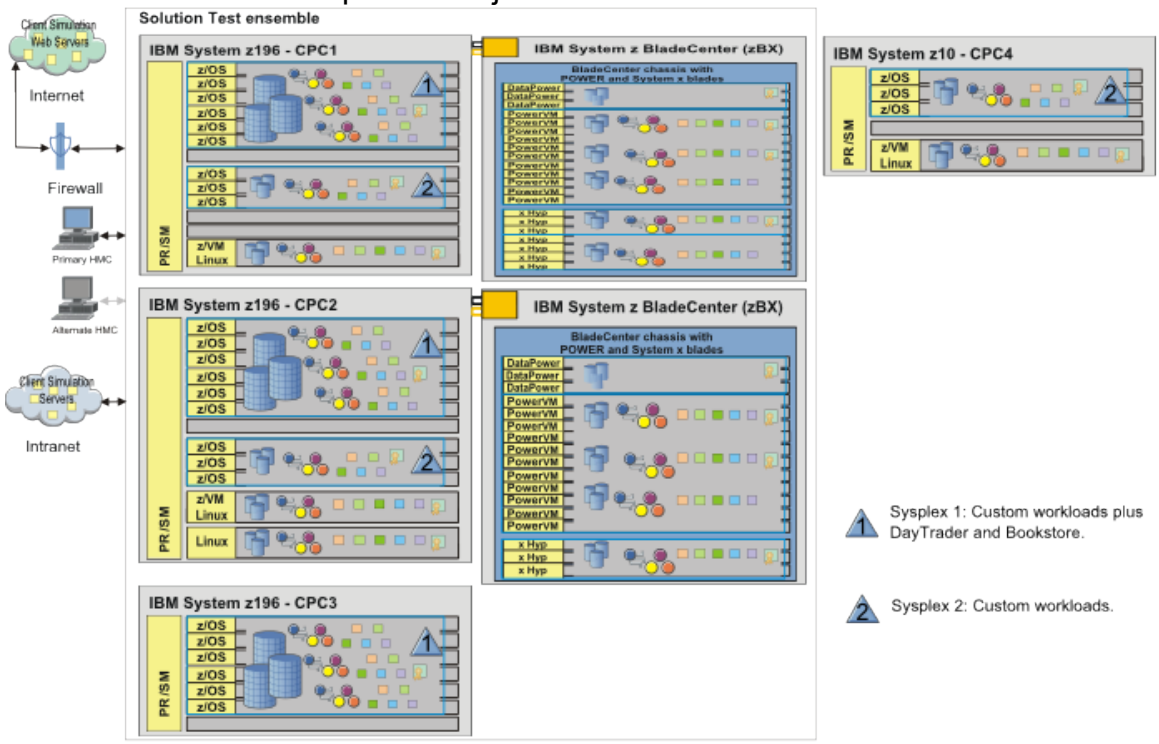


Figure 4 Solution Test Ensemble Diagram

Internal IBM accounts are traditionally used by system z to gain insight on new products and features. Early deployments of zEnterprise BladeCenter Extension at these production accounts provided valuable learning prior to formal product release. Regular product ordering and fulfillment processes are used by these internal account teams. This allowed the refit system z manufacturing processes for zEnterprise BladeCenter Extension to be tested. Installation instructions and product kits are developed and then verified by hardware system test. The end to end ordering, fulfillment, and installation processes were vetted during internal BladeCenter Extension installs. Additional learning from these efforts was fed back to development producing further improvements.

First customer deployments of zEnterprise BladeCenter Extension were afforded greater attention than standard zEnterprise installs. The lab services team was engaged to support zEnterprise BladeCenter Extension customer deployments. Lab services personnel worked with the test teams to learn product operation and recognize special circumstances requiring

attention. This training included installation recovery procedures. In conjunction with installation deployment experts, a number of offerings were created to ease customer workload migrations to zEnterprise BladeCenter Extension. These offerings are part of the Early Adopter Program. Two examples are Fit for Purpose workshops and Rapid Workload Optimization Assessments that have been instrumental in speeding customer acceptance. Taken together a rich set of offerings exist to open customers to the broad capabilities and possibilities presented by zEnterprise BladeCenter Extension.

Part 4: Conclusions

We have seen that enterprise class product expectations require 24x7 availability [10][11]. Meeting that availability demand with a solution comprised of mainframes connected to and managing cost optimized components requires thorough design. While zEnterprise is founded on the highest reliability and availability, cost optimized products are not. Thus, redundancy necessarily plays a central role in achieving overall availability objectives. However, the best designed features will never reach their potential without a test aligned to assure those objectives are met. zEnterprise BladeCenter Extension was designed with enterprise class capability in mind. The test teams set out to create a test that delivered enterprise class results. From the highly virtualized multi-tier workloads deployed in the solution test environment to the battery of mainframe caliber RAS tests aimed at assuring fault tolerance, the team operated with those goals in mind. Setbacks in such an undertaking are typical. Points existed where there was doubt that sufficient robustness could be achieved. Reliability gaps in components required extensive debug and creative solutions. zManager capability brought a new set of responsibility to system z. Integrating those functions into the base system z management structure was difficult. Again the test team worked with customer expectations foremost to achieve the desired results.

Besides being well versed in meeting what enterprise customers want, the system z test team had test environments with sufficient scale for zEnterprise. These test environments were already multi-tier and already had complex workloads running. Extending the test capability to include the tightly integrated zEnterprise BladeCenter Extension came naturally. The lab environments proved perfect training grounds for the world wide team of experts that would support BladeCenter Extension. These experts gained valuable experience participating in lab test efforts and then used that knowledge to drive product installations at internal production accounts. Experience from these first deployments provided lessons that guided further product improvements. Thus, when zEnterprise BladeCenter Extension launched, the product had already logged 1000's of production hours from internal accounts.

zEnterprise BladeCenter Extension provides a level of systems management integration that was not previously possible for system z products. Alternative management platforms do not offer these capabilities for system z. With the single zManager user interface visualizing and managing system z, Power systems, System x, and DataPower resources, BladeCenter Extension achieves what alternative products have long attempted to accomplish. Customer value of this integrated approach will be realized through more efficient datacenter operations management, better performing system z centric multi-tier workloads, and system z serviceability standards. These advantages are now being realized by enterprise customers around the world.

References and Acknowledgements

[1] PWR-Consulting, Plant-Wide Executive Series Report: Cloud Computing: Feds to Cut 800-Plus Data Centers by 2015, PWR-Consulting, 2011.

[2] C. Progler, G. Richardson, P. Lavely, ExecBlueprints: Overcoming Technology challenges when the Business is Always On, Books24x7, 2009.

[3] J. V. Jones, Integrated Logistics Support Handbook, Third Edition, McGraw-Hill, 2006, Chapter 4.

[4] P. Jorgensen, Software Testing A Craftsman's Approach, 3rd ed., Auerbach, 2008, pp.7-9, 353

[5] B. Beizer, Software Test Techniques, 2nd ed., Van Nostrand Reinhold, 1990.

[6] Clabby Analytics, IBM's New z114: A Replacement for SPARC/Itanium Servers? (July 2011), www.clabbyanalytics.com, Available at: ftp://public.dhe.ibm.com/common/ssi/ecm/en/zsl03146usen/ZSL03146USEN.PDF Accessed: April 27, 2012.

[7] zEnterprise System Ensemble Planning and Configuring Guide GC27-2608-04, version 4a, IBM, 2012, p.9,19,25,53

[8] B. Pettichord, Schools of Software Testing (2007), www.pettichord.com, Available at: http://www.prismnet.com/~wazmo/papers/four_schools.pdf Accessed: May 1, 2012.

[9] E. van Veenendaal. Standard glossary of terms used in Software Testing. Version 2.1. 'Glossary Working Party', International Software Testing Qualifications Board; 2010. Available at: http://istqb.org/download/attachments/2326555/ISTQB+Glossary+of+Testing+Terms+2+1.pdf Accessed May 7, 2012.

[10] N. Bajgoric, Always-On Enterprise Information systems for Business Continuance: Technologies for Reliable and Scalable Operations, IGI Global, 2010, pp.xvi

[11] G. Shroff, Enterprise Cloud Computing: Technology, Architecture, Applications, Cambridge University Press, 2010, pp.3-10