IBM CORPORATION

Moderator: Lynn Winkelbauer
November 3, 2005
9:30 a.m. CT

Operator: Subject of this lecture is EWLM, 24-hour turnaround.

Lynn, please go ahead whenever you're ready.

Lynn Winkelbauer: Hello. This presentation is called the IBM Virtualization Engine version two release one. It's specifically focused on the enterprise Workload Manager, or EWLM, as I'll be referring to it.

Several of us put this package together, Derrick, Holger, Rick, and myself. I'm Lynn Winkelbauer, and I'll be presenting this for the rest of today.

If you flip to the next chart, the fine print, this chart just is IBM copyright laws. If you go to the next chart, this is a high-level overview of what I'll be covering today. We'll talk about the preface. This will cover the objectives (for this) audience as well as the agenda.

We'll give an introduction to EWLM and dive harder down into the architecture and the technology. I'll talk about the sales and marketing as well as the EWLM values and conclusions. And we've also listed some of the information resources that you can use.

If you flip to the next chart called learning objectives, basically what we want to do is to prepare you so that you should be able to explain where EWLM fits into the overall VE strategy. You should be able to describe the architecture of EWLM, its capabilities and functions, and basically to explain the EWLM road map.

You should be able to explain and understand the client pain points that EWLM addresses and be able to position EWLM to the other VE components as well as WLM, PLM, and Tivoli. We'll differentiate the EWLM against the competitive offerings and articulate and justify the client benefits and the value of EWLM.

Now, if you flip to the chart, the target audience, you'll see that this presentation is aimed at those folks who are positioning and planning the solutions that are utilizing IBM's virtualization capabilities.

Specifically, we're looking at the (FTSs), the ATSs, tech line, system architects, as well as the non-IBM personnel that have similar job roles, such as our business partners.

If you flip the chart again, recommended pre-req knowledge, what we're looking at for this presentation is that the audiences have a knowledge about the IT infrastructure as it relates to automation and virtualization, and they should also have an overview of the concept of an on demand operating environment.

Now, if you flip the chart to the agenda, the beginning of the agenda, what I'm going to go through is an introduction to EWLM, basically talking about today's infrastructure challenges, how EWLM is a strategic platform as a cornerstone, I'll talk about the high-level overview of EWLM, its benefits, as well as some of the positioning of EWLM.

Following that, I'll get deeper into the architecture and the technology, the structure of EWLM, it's (apologies), service classes, as well as some reporting and management capabilities, and then just some examples to recap and show the usability of the EWLM capabilities.

I'll follow that with the competition, our ISV status, as well as the EWLM road map.

If you flip the chart to the second page of the agenda, we'll start with the sales and marketing, basically some customer and use cases for VE, including the EWLM product, as well as the packaging and fulfillment, and then talk about the EWLM values and conclusions, basically IT values as well as business values.

And then included in this material is the information resources that you can use to help you understand the EWLM product.

So, let's start off with the agenda. I'll now go into the introduction to EWLM.

If you flip the page again, you'll now see a picture of today's complex infrastructure which creates challenges for our customers. You can see here that customers have many different servers, firewalls, switches, appliances, various types of servers, as well as middleware products.

And the management of this complex heterogeneous environment is getting very complex and too hard to handle. Typically the servers are run at a lower utilization to accommodate peak workloads.

Customers have to take into account privacy and security as well as business continuity, and basically the proliferation of technology and platforms to support is getting a lot harder as well.

So, what we want to do is to have a management capability to manage this infrastructure seamlessly.

If you flip the chart again to the platform approach, here what we're doing is showing the further refinement of movement of the traditional approach to virtualization and the problems that we are trying to overcome, such as closed systems and unique tooling, dedicated resources, as well as limited capacity management.

The VE strategy is to use a set of guiding principles, such as open standards and interoperability to create an open ecosystem and to provide freedom of choice of which platforms the customers want to choose.

We do this through common skills and a comprehensive product set. As you look at this strategic platform, this is something that the WebSphere environment has been doing for a while. The VE strategic platform consists of the virtual access, virtual management, as well as virtual resources, and we'll talk about these three items further on in the presentation.

Now, if you flip the chart to the next page, what you'll see here is the virtualization within the on demand operating environment. Now, if you've worked in the on demand operating environment, you probably have seen this chart many times before. And what we're trying to do here is to position the VE strategic platform as it fits into the overall on demand environment.

You'll see there the virtual access, the virtual management, as well as the virtual resources, how they play into the on demand environment as they map to your infrastructure services, your resource virtualization services, as well as the service level automation and orchestration.

Now, if you flip forward to the next chart, IBM virtualization engines, a complete portfolio, here is the virtualization engine product. If you start at the top, you'll see the virtual access. And this is the programmatic interfaces, which is based on SOA and Web services.

You have a programmatic access as well as a virtual view. The products which fit into this category are the VE console as well as the IBM total storage productivity center. These map into your virtual management products.

The virtual management consists of workload and performance managers, resource managers, modelers, mapping. And this is where the enterprise Workload Manager, or the EWLM, sits, as well as other resource management products such as the resource dependency servers.

Below this, what you'll see is (a) virtual resources, or resource virtualizers. And these are the machines that you're running today, whether they're on a partition, a virtual machine, I/O network, virtual tape, as well as IBM server and storage devices.

On the left-hand side, what you see is virtualization planning tools. Now, these planning tools help customers plan for virtualization technology. They help define how many processes or LPARs that they'll need, the pre-requisite products, as well as the system requirements for those products.

And now, if you'll flip forward to the next chart, IBM virtualization ecosystems, what you see here is how we're leveraging open standards. The virtual managers support third party virtualizers through industry standards, such as arm, WSM, WSRF MUWS.

We support non-IBM devices and services while leveraging the functionality of IBM's devices to differentiate them.

Now, if you flip again to the chart called EWLM and other VE components, what you'll see is how the VE components, like EWLM, IBM director, the RDS and the VE consoles complement each other, and they fill in the strategic platform with these products.

The ongoing standards-based approaches ensure that these new components can easily be added or existing ones (extend) – existing ones (expanded), for example, the resource dependency services, or RDS and VE, too.

If you go to the next chart, we'll talk about the levels of workload management and resource balancing. What we want to do is get to the phase of describing what EWLM is and what it can do for you.

If you look at the bottom of the chart, we already have operating system level workload managers. And basically that's platform-specific. It works on multiple applications but within a single operating system.

You isolate important jobs using resources within that operating system. If you take that up a level, you'll look at the system level workload manager. Those are things like your partition load manager, or PLM, and your IRD (or your intelligent resource director). Again, this is serving a single server but with multiple partition optimization. So, you can now optimize things like your CPU and memory resources within those partitions on the box.

Now, the enterprise Workload Manager is platform agnostics. We can now monitor applications and transactions across a multi-tiered heterogeneous environment. We can now align those resources with your business priorities. So, you're now managing it, managing your resources, based on your businesses, not on IT.

So, as you look up the chart, what we're doing is we're increasing the scope and the functionality of workload managers starting at the operating system, going all the way up to EWLM .

On the next chart, enterprise Workload Manager overview, you see the main functions of the EWLM product, which helps IT infrastructure respond to the business needs. The first one is a dynamic detection of the servers and the applications running on IBM as well as on non-IBM platforms.

EWLM will automatically get notified of these systems and applications. It is based on a dynamic policy-based workload management. So, we are going to take full advantage of optimizing the resources in multi-tiered heterogeneous environments.

We're able to collect performance data on an end-to-end basis and report on that performance data. We'll get application performance that has actually happened compared against the pre-defined goals.

So, since you have goals, what I can do now is define my high priority goals and my low priority goals. So, if I have a service level agreement and I have high priority work, I can now take those goals and influence my network balancers so that the balancers will route more of higher priority workload to those systems that are performing better.

Once the workload is running on the systems and EWLM is monitoring that work, we could then dynamically adjust the CPU allocations within a Power 5 box. This will raise the system level workload management up to the enterprise level.

So, on the next chart what we have is what clients can benefit from EWLM . And what we saw is that customers who are running in a multi-vendored, multi-tiered distributing application environment is going to have a lot of benefit from EWLM .

Also, clients that have concerns about meeting business based performance goals. Customers migrating to an on demand environment, and those with requirements for reporting service in terms of business groups or business policies, such as your service level agreements.

Customers who value the ability to be nimble, reacting to change in demand, and those who favor the use of open technologies in managing their server (farms).

On the next chart, customer use scenario, you can see how EWLM allows users of IBM's Power 5 servers to automatically change the LPAR configurations based on your business goals defined in the EWLM policies.

On the left you see how the service level workload management works where you have multiple LPARs, maybe each has its own consoles, potentially its own tools for determining the performance of those boxes, and then you have an end user that interacts with the LPAR to move a CPU from one LPAR to another within that Power 5 box.

So, you potentially have multiple tools, multiple consoles, and multiple people that's working on this workload management function.

What EWLM does is raise the system level workload management up to the enterprise level. You now have EWLM monitoring your service level agreements, looking for exceptions, and based on the goals that you've defined can automatically interact with the LPARs to move CPUs around.

So, you see the LPARs 1, 2, and 3, EWLM has looked at its business goals, has determined that the workload running on LPAR 2 has higher priority and has a need for more CPUs and can dynamically move the CPUs from LPAR 1 and 3.

This information can then get proliferated up to the VE console so the VE console has an overall view of what's going on in the EWLM environment as well as other management environments.

If you flip the page to the value of EWLM, it's really in the IT productivity. It is an automated workload management facility for use in a distributed heterogeneous infrastructure. It can manage to the business service levels, improve the utilization IT resources because I no longer have to run my resources at a lower utilization to accommodate the peak in my workload.

I can see how much resources are being consumed by my workload, and I can move more resources to where the high priority workload is or route the work to where the resources are more available.

I can rapidly understand the quality of service delivery from a customer perspective, from a business perspective, not from an IT perspective. And I can identify the bottlenecks that links the server resource data to where the bottleneck is.

The performance expectations are the same as understood by the CIO, and basically that's because the performances are defined at the business level, not at the IT level. And so really what EWLM does is close the communication gap between the business and the IT.

On the next chart, monitoring goals and service level definitions, what you can see is the relationship between EWLM and another Tivoli product, TMTP. Now, Tivoli solutions complement the virtualized infrastructure since the virtualized environment increase the value of automation to the customer.

Customers can leverage both EWLM and Tivoli where EWLM manages the resources that serve the applications. TMTP manages the application and performs problem determination. EWLM can be integrated with TIO, or Tivoli Intelligent Orchestrater, for an optimized IT environment.

And this really provides a seamless integration. So, you see here, EWLM's focus is more on the hardware and on the operating system where the TMTP product is more focused on the applications and the middleware.

Now, EWLM can complement the TMTP product as well, to break down CPU usage of servers that contribute to customer applications and processes for useful and forecasting and capacity planning.

TMTP couldn't easily provide this breakdown of resources per server.

On the next chart EWLM, TMTP, and TPM, what you can see here is how the products are working together. Now, if you're in screen show mode, the first picture you'll see is EWLM, and here you see an IT administrator basically looking at the EWLM end user interface and he notes that there's a performance problem for a critical application using the EWLM panels.

If you scroll forward, number two says because of path application problems, the administrator can use TMTP to do root cause analysis, and basically, what he can determine from using the TMTP product is that this application needs more CPU.

And so if you scroll forward one more time, number three says basically that the administrator then uses Tivoli Provisioning Manager, or TPM, to execute a work flow that can take a free resource from a pool and move it into the HTP cluster, if that's where the problem resides.

Now, if you scroll forward two times, what happens there is that once the resource is provisioned, EWLM will dynamically discover this new server, include it into its (pause) management domain, and immediately begin leveraging it to rebalance the workload and to maintain the application performance.

On the next chart what you see is the relationships between automated orchestrations of future integration. And that is the integration between EWLM and Tivoli Intelligent Orchestrator.

On the first chart shows EWLM, again, detects the service level objective is going to be breached due to some limited server capacity. And so through the EWLM objective analyzer, we will then send a recommendation to the Tivoli Intelligent Orchestrator.

So, if you click forward once, what you see here is then TIO arbitrates these incoming requests, and it determines what appropriate action to take. In this case, because we need a new server, it will trigger TPM to execute the appropriate workflow to provision a new server into our environment.

If you click forward once, here you'll see the TPM product, and what it will do, automatically, is to execute a workflow, which will again move the free resource from a spare pool into the cluster. If you scroll forward, one, two more times, what you'll see then is automatically EWLM will discover the new server and its topology and again begin leveraging it to rebalancing the workload.

This is all done without human intervention. So based on the performance that EWLM is obtaining from all of this managed servers and based on the goals that you've defined, it's going to notify Tivoli who then is going to provision new resources to make sure that you maintain your business service level agreements, or your service goals.

Now, let's scroll forward and get deeper into the architecture and technology of the EWLM. Let's go to the next chart, VE version two release one EWLM.

In version two of EWLM, we've increased the number of platforms and middleware products. In addition to Windows, AIX, (I5OS), Solaris, and (DOS), we now support Linux as well as HPUX. As far as the middleware, we support WebSphere, DB2, (UZB), Apache, and (IAF).

Now, some of the core infrastructure changes for version two is serviceability, and basically that is providing platform API tracing. We've improved the scalability and performance. We've made several usability enhancements, including new reports, and also we have installation and configuration enhancements included in EWLM version two.

As far as the resource optimization, we have better algorithms. We have Power 5 LPAR CPU management, and we have ERD, or the EWLM resource director. And that's similar to IRD, if you're familiar with that.

Now, on the next chart, entitled EWLM architecture overview, this shows the architecture of EWLM. So, I'll start at the top left. This is the control center, and this is the EWLM end user interface into the environment.

We have a domain manager, which you can look at as the brains of the operation, and it communicates with the managed servers. Now, the managed servers are those servers that you're running today in your environment, and these managed servers are all part of a management domain.

So, I'm going to go into each one of these components now.

If you scroll forward, we'll now talk about the EWLM control center. This is the Web user interface. It is role-based. It has authentication and encryption, and this is where you define your business policies and you maintain your business policies.

You have a high level business-driven reporting. You can see topology from a business view of where the transactions entered into your management domain and around, as it went round trip, say, into your HTP server, your WebSphere and to your DB2.

You can now drill down into those applications and servers and get very detailed information, such as page faults, memory, CPU, how much am I consuming for a certain service. I get response time and resource data, resource consumption delays as well as transactions, and process-centric work.

I can rapidly understand the quality of service delivery from a customer business perspective. I can identify bottlenecks and link those bottlenecks to a server.

If you scroll forward, what you see here is the EWLM control center roles. There are basically three roles that you can set up to access the EWLM control center: the administrator, the operator, and the monitor.

Now, the administrator has access to all of the directories. If you look at the right hand side of this chart, you see here set up, manage, and monitor. You'll have access to all three of those. You have the capability to create and edit domain policies, deploy those policies, and activate the service policies.

You can also manage and monitor the environment.

The operator has access to the managed as well as the monitor subdirectories. This user has the capability to activate service policies as well as to view reporting data. The monitor role basically has access to the last subdirectory, called the monitor, and this user has the capability to view reporting data both at a high level as well as a detailed level.

If you scroll forward, this is a picture of the VE console. And basically the advantages of console consolidation are obvious, and mentioned on this slide here. From a practical and implementation perspective, it's worth noting how the EWLM works into the VE console.

Here you see the help center of the VE console providing consolidated help of the administrator's role. It eliminates the need for the administrator to look at multiple consoles just to determine if everything's running OK.

It's focus is to monitor the health of the system, identify any kind of problem area, and to take corrective action. It's going to get its input from the managed servers, such as the IBM director management server, IBM Tivoli's monitoring, as well as EWLM.

All that information gets set up through the VE console, and the VE console will be presented to the administrator, and he can then have eye-catchers that he can look to see if there's something going wrong in this system.

If you scroll forward, the VE console launch pad. The VE console has two things, the help center, which you saw on the previous slide, and the launch pad. The VE console will detect the EWLM control center and automatically link to it.

So you can just click on a (button) within the VE console and get to the EWLM control center. In version two, the EWLM has been more integrated into the VE console, so a lot of the charts that you'll see right from the EWLM control center will be presented from the VE console as well.

If you scroll forward to the EWLM domain manager, the domain manager basically has a global view of the entire management domain. It assimilates the data and learns the topology of the business applications.

So, as the transactions are traversing from one server to the next, that information is fed up to the domain manager. It keeps the topology and the state management. It has secure messaging, firewall, as well as being DMZ friendly, and it is the global coordinator of those business policies.

It allows you to have external APIs into the domain manager. So there are system management APIs that allow you to get to the data that the domain manager has. And there's a connectivity here to the hardening of the EWLM performance data (Redpaper).

There are APIs to let you get to the network. So if you have a network balancer, you can take advantage of these APIs to help balance work into the environment. And there's dynamic application workload balances as well.

So, you have three different types of APIs that you can use to get into the domain manager to get to its data and to help with its function.

On the next slide, the EWLM management domain, that is basically the green circle you'll see on the right-hand side. This is a collection of servers that support a business application, and this is regardless of the hardware or the operating system environment.

It could be individual servers, partition servers, or virtual servers. There's general support for all servers, for all platforms, and depending on the operating systems, there's specialized optimization for IBM platforms.

You could run a mixture of application technologies or middleware technologies, and basically the primary emphasis is really on the new workloads, for example, the WebSphere environment. Although we are giving high value to our legacy systems, such as (KIX) and (IMF). So they will be included into this management domain in the future.

We accept hybrid technologies, for example, DB2 and MQ, as well as non-transactional process-centric workloads, such as batch jobs.

Now, if you scroll forward to the EWLM managed server, what you see is these servers that you're running today running your workloads. There's a common component for all the operating systems. It's Java-based and has limited platform sensitivities, although there is a native interface into the platform-specific layers to get unique information from the various platforms.

It has coordination with the domain manager. It will extract all the local statistics from its server, and propagate that information up to the domain manager, and it imports the end-to-end global statistics.

There is an operating system coordination, basically the transactional data from instrumented applications, process-centric data from the operating system platforms, and it assimilates, correlates, and summarizes data and sends this data up to the domain manager, so it has the global view from each one of the managed servers.

If you scroll forward, you'll see the EWLM structure. Here you see three different operating systems. On top of the operating system is the EWLM managed server code. Once the EWLM managed server code is installed on the system, the domain manager can now get information on the operating system.

On top of the EWLM, there's arm-enabled applications. So as work comes into the Web server, goes across to the WebSphere and into a database, the information of that transaction is getting sent to the local EWLM managed server.

The information then is sent on to the EWLM domain manager, and you can view that information from the EWLM control center.

From the domain manager you have the policies, and from the EWLM control center, you will take these domain policies and deploy them to each one of the managed servers.

On the next chart, EWLM approach: middleware participation, here you see the end user coming in with a request. This request can touch one application, or multiple applications. These applications can run on one or more operating systems, and these operating systems can be run on one or more servers in a virtual environment.

What I have with the middleware participation is an end-to-end response time reporting. I can have multi-tiered work requests flowing across all of these environments, and I have a relationship of the server resources being consumed by each one of my transactions.

If we scroll to the next chart, correlation with Arm, this is really how we get the end-to-end collecting and reporting. As workloads come in or as a transaction comes in, the application will then classify it with an Arm start transaction API.

It will classify this transaction class, called “buy”. As the transaction goes from hop zero to hop one, this information is passed along with it. So as it goes from the Apache server to the WebSphere server over to a local application, the time that's spent in each one of these hops are passed along with it.

As it comes back, each one of these EWLM managed servers are going to tell the managed system’s domain manager of how much time is spent on each one of these hops. So, now I have a correlation of a transaction as it traverses across my environment, whether it's on the same system, different systems, different operating systems, and different platforms.

If you scroll forward, management of arm-instrumented workloads, or middleware, the workloads are running in an arm-instrumented middleware, and therefore we can get end-to-end transaction response times at a detailed level.

So, each hop in a multi-tiered application is managed based on its contribution to the overall response times. EWLM then will have an understanding of the application topology as it flows from one hop to the next, and it puts this in context of the end-to-end goals.

If there's other workload running on this system, it does not allow that workload to be hidden by the good performance running on the other workload. .

There are various performance goals that can be set such as velocity goals. If an arm-instrumented middleware is set, you can specify response-time goals, and they are the most intuitive performance goals.

If you look at the next chart, arm instrumentation, this is how the transaction is sent from one hop to the next (even when one of the applications are not arm-enabled). A user request comes into a product who has arm instrumentation. That product will then use an arm start transaction.

This information will then get sent to the next product, product B. So, the information is sent along with this correlator. Product B here has some business logic. It does not do an arm start, so I'm going to miss the information that's gathered within product B.

The transaction then flows, as it did before, to product C. Product C does not see the arm instrumentation and therefore it will also issue an arm start and reclassify the transaction. In each case, each EWLM will get the information for its local server and pass that information into the EWLM domain manager.

So if the middle tier is not arm-instrumented, I will still have the end-to-end response time information. However, I will not have the breakdown of each one of those hops because I cannot get the information on product B because it does not do an arm start and arm stop.

Therefore, I can't see how much time is spent within product B.

On the next chart, we talk about EWLM arm support and why Arm 4.0. There's a group standard-based instrumentation. It was adopted by the open group in 2003, and basically the Arm 4.0 open level contains important extended capabilities:

I can have application and middleware instant name and grouping. I can get transaction context capabilities, the transaction instances run-time attributes, such as the user and the URI value. I get transaction delay times, the block time where I'm waiting for the next hop because of something in the next hop I cannot send the transaction too. Perhaps there's not enough threads to go to DB2 so I'm blocked in getting to DB2.

And I have a thread to transaction instance accountability. So Arm 4 allows us to use EWLM to gather the most detailed information on each one of those hops and present it to you through the EWLM control center.

On the next chart, Arm 4.0 instrumented IBM middleware, we show here the products that are Arm-instrumented. In the Web server plug-ins, you see (it's) configured into the IHS 2.0. We have the application server based on WebSphere 5.1.1 and later, and we have the database servers, DB2, UDB version 8.2 as well as DB2 on z/OS.

On the next chart, high level of how the installation process works, basically what you'll do is you'll take the product media and you will install the EWLM domain manager code. There's a

wizard that takes you through the installation, and with that you would define the EWLM domain name and the ports that you want the domain manager to use.

Once the domain manager is installed, you will then copy the information for the managed servers, over to the managed server via the IBM director or simple FTP.

Once those installation images are copied, you'll then install and configure the EWLM managed server. Again, you'll have a wizard there and you'll want to (be able to) point to the EWLM domain manager using the host name and the ports that you defined previously.

At this point in time you'll be able to capture the information on the operating system.

The next step what you'll want to do is to arm-enable your applications. Now, this is not a difficult ~~structure~~ (task) if you're running on one of those arm-enabled instrumented products that I mentioned on a previous slide. It's basically including certain Java classes and turning on arm enablement.

However, the next (spot) is where you'll configure your EWLM domain policies. And this is what you'll want to take a lot of time to specify what your goals are. This is where, if you have service level agreements, you will use those service level agreements to specify the performance goals that you want EWLM to use.

On the next chart, we break down this domain policy for EWLM. We show the components. You'll see the task classification and business objectives that are met to describe through classes, goals, and description rules.

The service classes are an essential concept to EWLM, and it defines a group of work that has similar performance goals, or business requirements. The goals are how fast you want your

workload to run, and based on certain rules, if you're coming in via certain URI, from a certain user, et cetera.

The classes are (being) grouped into applications, platforms, and workload. So, my goals you'll see here are things like my percentile response time, my average response time, velocity and discretionary goals.

You can also specify how important it is to your business that the goal be met. Importance plays a factor only when the service class is not achieving its goal. For instance, as resources may be taken away from a less important service class in order to meet the goal for a more important service class.

On the next chart, you'll see the EWLM monitoring and reporting. When a policy has been activated, you can now monitor it using the EWLM control center. You get information like that shown in the first level reports. These are high-level reports.

You can see things for exceptions if I'm not meeting my goals. I have views that I can look at to see what my goal is versus the actual performance that's being obtained for my service classes, my transaction classes, as well as my process classes.

I could also look at my managed servers. I could see my CPU, (the) ~~my~~ memory of those servers, what operating system it's running on, and what level those operating systems are at. For each one of those operating systems and applications, I have detailed reports.

So I can drill down into each one of them. I have topology reports that show the flow of a transaction as it goes across the various hops. And that's the picture you see on the right-hand side.

I have the Apache server, a WebSphere and a DB2, and I can drill down into each one of those application servers to find out how many transactions are running there, the response time, the success rate, et cetera.

I also have real-time performance monitors, such as the goal achievement. Am I meeting my goal? What is the processor utilization? And what's the transaction count and the rate of the transactions entering into this environment?

On the next chart we show the EWLM and the z/OS workload manager and the integrated reporting facility. EWLM reports delivers performance data in a multi-tiered application, inclusive now of the DB2 database that's running on z/OS.

(If) we have performance problems in a z/OS application tier, ~~and~~ that's still going to be managed by the z/OS workload manager support. There'll be a service class for workload manager that corresponds with service class for EWLM.

So, previously we had an EWLM policy working on an end-to-end environment. You have an EWLM reporting facility exclusive of the z/OS. Now, the z/OS reporting facility is included into the EWLM reports and, for z/OS, I can include the end-to-end performance goals that’s set by EWLM.

So you’ll see more and more of an integration between EWLM and zWLM over time.

On the next chart, enterprise Workload Manager Version 2, what you’ll see here again is a complete eServer platform coverage. We have the LPAR management for Power 5, and that has the ability to make changes in the amount of CPU resources that’s used in a partition.

We have the Enterprise Resource Director, or ERD, and that's designed for un-Armed instrumented applications, such as Oracle. There's velocity-based goals for the entire partitions that can be set, which are really similar to IRD on zSeries.

So, if you're a zSeries person that knows IRD, this function here is very similar to it. And it's primarily aimed at a single application per server or per operating system. There'll be an EWLM TIO integration service offering that I showed before that integrates the EWLM product and the Tivoli Intelligent Orchestrator that automatically will provision new servers based on your service goals being breached.

On the next chart, LPAR management for Power 5, what you see here is a management of CPU resources based on your EWLM domain policy. It's very similar to what's been done for zSeries IRD. We'll take advantage of the virtual processor optimization.

And as you see here, I can have several partitions, regardless of the operating systems, sharing a CPU. And based on how I've defined my domain policy and my goals, I will move CPUs from one of those partitions to the other.

It works on shared processor partitions only. However, I can have multiple EWLM domains per machine.

On the next chart it goes into a little further on the LPAR management and talks about the platform and the operating system support. It is available on Power 5 hardware, AIX (5L) version 5.3 with some maintenance, as well as i5 O/S version 5.3 and additional PTFs, as well as PPC, Linux (SuSE).

On the next chart, we just show an example of this LPAR management. So you see here I have four partitions. Each partition has three virtual processors. On the left-hand side, work is coming in ~~from~~ externally. It's called service class 1 (SV1), and it's going to an Apache server.

The Apache server will route the work over to the WebSphere. WebSphere will then direct the work over to DB2. On the right-hand side, I have lower importance work, ~~and~~ probably some internal work, and we classified this as with service class called (SV2).

So initially all the configurations were equal, all work with the transaction response-time goals were set, and we defined the workload on the left as being more important than the workload on the right.

Now, if you scroll forward once, what you see here is there's a surge in the (SV1) work, and it begins to miss its goals. EWLM will detect this and see that it's a CPU delay, and what it will do is it will take resources away from the WAS partition that's supporting the lower important work, (SV2). EWLM will also determine that (SV1) will do better by taking resources away from the Apache partition.

Perhaps the partition had defined three CPUs but only needed one. So, it will take the CPUs away from the Apache server and give it to the WAS/DB2 partition. So, now the number of virtual processes are adjusted to match the new resource allocation.

As the workload continues to run, EWLM will continue to monitor it, and if one of the other processors, or the partitions, needs a processor back, EWLM will do the same thing, taking a processor away from either the WAS or DB2 partition and moving it back to the Apache or the second WebSphere.

If we scroll forward, management of non-arm instrumented environments. Without arm-instrumented middleware, EWLM doesn't have that detailed end-to-end perspective, so management is based solely on the local performance of the servers.

There's two options for managing these environments where the applications aren't Arm instrumented. Option one is you can classify the process (the operating system process). Now, the processors can be assigned to a service class that can either have a velocity goal – and this is most likely good for, you know, your long-running, never-ending type of job, or it can have a response time goal. And this is something for a very short-running job, such as a short-running batch job.

The LPAR management is based on how well these process service classes meet their goals compared to the goal achievement of other partitions running on that same processor. The approach adds complexity because now you need to understand the role of the individual processes and the velocity goals.

The velocity goals are not as intuitive as response-time goals. This approach is best matched for partitions when I'm running on mixed, non-arm instrumented workloads.

On the next chart we'll see option two, and this is managing at the partition level instead of the process level. Here, the entire partition is given a goal, and the management of the partition is based on the partition's achievement of this goal.

The assignment of the goal and its importance allow EWLM to put the partition performance in perspective of the work that it's managing. We've been calling this ERD or ~~EWLM~~ Enterprise Resource Director.

(The) partition is assigned a velocity goal that measures how well it's being treated by the hypervisor. Now, velocity goal says that the percentage of the time that the workload is ready and able to run, and not delayed for any resources, actually is able to get the processor.

That's why ERD best matches partitions running a single primary workload that can be viewed as having a single business importance. This option is less complex than option one because you don't have to understand the role of the individual processes.

It allows partitions to be viewed as a single entity from a point of view of the performance administrator.

On the next chart you see examples of use of the partition level goals. On the left-hand side, you'll see four partitions, and on the left-hand side you see the two partitions that's managed by the transaction response time goals.

On the right-hand side, you have a database that's not arm-enabled, and you have some other print servers and DNS servers. So, here you'll see that those partitions are managed at the partition level, or managed at the velocity goal.

The partitions with non-arm instrumented work are given partition-level goals, while on the same server, I can have two other partitions managed at the transaction response time goal.

If you scroll forward, again you'll see examples of the use of the partition-level goals. Partition one is running arm-instrumented work as well as non-arm instrumented work. Partition one can be managed both at the transaction response time goal as well as the partition-level goal.

Missing either of those goals can make a partition receiver of the resources, for example, CPU from partition two or three, and we could take our resources from the partitions and make sense for both goals on partition one, partition two or three.

So, here you see four partitions. The first one's managed both by response time goals and partition goals. The middle one is only running WebSphere, so we can run this solely based on response-time goals, and on the right, partition three and four is managed by partition level velocity goals.

Now, on the next chart we show how EWLM is working with the workload balancers. EWLM managed servers are getting the information from its local resources and sending that data up to the EWLM domain manager.

So, the EWLM domain manager has a global view of all of its resources in the management domain. It will then send this information to the load balancer. The load balancer will query saying, "How well is the work running on each one of these three processors."

EWLM will feed the performance information to the load balancer. It will influence the routing decisions made by that load balancer, such as a Cisco router. So, now workload (performance) can go to the load balancers. The load balancer can then take recommendations supplied by the EWLM domain manager and start routing work based on those decisions made by EWLM domain manager.

On the next chart, we see EWLM and z/OS WLM, the goal management. Now, it shows here zSeries keeps its goals. And basically what we want to do is simplify the analysis of z/OS performance through integrated DB2 data and include this in the overall EWLM reports.

The EWLM goal can be assigned to a z/OS WLM goal. The z/OS workload manager will continue to manage the z/OS resources for goal satisfaction, but now it includes the information on an end-to-end basis from EWLM.

The service class correlation - handing over goals. The goal specification for a specific EWLM workload is made known to the z/OS workload manager. EWLM service class is correlated to a z/OS workload manager service class, and this is done by the z/OS administrator.

The z/OS workload manager manages EWLM workloads now towards the goal of correlating this service class. z/OS workload manager reports the resource consumption of their EWLM workloads in the WLM report class.

On the next slide, we show a new EWLM arm serviceability adapter. The purpose of the EWLM arm service adapter is to capture and record arm classification and transaction data. The data will then be used in diagnosing classification problems.

The arm serviceability adapter is available on all platforms supported by EWLM, except Windows, which has its own internal tool, (Win80) that will perform the same data capturing.

On the next chart, we continue the discussion on the EWLM arm serviceability adapter and just give an example of why we'd want to use the serviceability adapter.

Say the customer has an EJB installed on a WebSphere and provides a bank account services for a variety of users. The customer likes to group all transactions with this EJB with a user name beginning with a character A into service class A, and every other transaction into another service class called service class B.

The domain policy and the service policy reflecting these goals are created, deployed and activated. Now from the EWLM control center, the customer is seeing that (no) transactions are being classified into service class A and everything's classified into service class B.

This problem is reported to the IBM service. Where would you start? We'd probably want to look at the arm serviceability adapter. What that will do is show you how your services were classified, and in this example, what we're seeing is that they are basically looking at the last name followed by the first name where they're expecting to be looking at the first name followed by the last name.

So, it's a way of looking at, "Did I classify my workload correctly? Where the workload is coming in, is it what I expected it to look like?" And if I have any conflicts in my transactional definition, this adapter will pick it up.

If we look at the next chart, EWLM competitive landscape, what you'll see on the left-hand side is Aurema, Dell, EMC, HP, IBM, Microsoft, and Sun, and shows you where they position themselves and comparing this against the EWLM product.

Now, our biggest competitor here is the gWLM. The second EWLM release closes many of these functional gaps that HP gWLM has. Now, IBM EWLM has a big advantage over the platform support because it is platform agnostic.

So, you may want to use this chart if your customer has an EMC or potentially a Microsoft, and show you where EWLM will play according to what they have today.

On the next chart we talk about the ISV enablement. IBM is working with all the top ISVs to enable EWLM. Those that run on IBM middleware, such as WebSphere, are in a good position because they can take advantage of the arm enablement within WebSphere.

ISVs that don't run the IBM middleware can adopt the arm instrumentation. They're most likely to instrument the middleware if the customer demands, so we're asking the customers to go to their vendors to say, "I need this function in a certain product."

And IBM is certainly willing to consult with them on EWLM enablement. There are a lot of ISVs now that are either in a proof of concept or they're actually going to market, and we've listed these on the right-hand. So, you'll see Cisco, SAS, Chrodiant, et cetera.

On the next chart we show EWLM strategy and road map. In 2004, we released the first release of EWLM. This release included end-to-end collection and reporting of the performance goals and dynamically adjusting the routing weights for Nortel and Cisco switches.

In release two, we're focusing on the IT optimization. We're using LPAR resource adjustments to meet the business goals. We're using Enterprise Resource Director to manage non-instrumented applications, and we're monitoring and managing from z/OS as well as all Linuxes.

In 2006 and beyond, we're extending EWLM's scope to include integration with WebSphere XD, manage the local CPU to prioritize workload within a given operating system, and to integrate with a virtual machine manager, such as VMWare, Xen, zVM, and Microsoft virtual server.

We continue now with the agenda, looking at the sales and marketing. We're now on the slide market, client-suitable for implementation of EWLM. So those who are suitable: customers who need to manage a multi-vendor; multi-tier distributed application environment; those that have concerns about meeting business-based performance goals, customers going to an on demand environment; those requiring reporting service in terms of a service-level agreement, your business groups, business policies; customers who value the nimble reacting to change in demands; and those who favor open technologies in managing their server (farms).

We have listed here on the next slide a customer use scenario. ~~And~~ these are business models that are moving from a vertical silo to a horizontal process. In this example, there's a customer relationship management, the CRM. The silos are being broken down in customer environments that are managed globally across into enterprise.

So, here you see on the left-hand side, developments, sales, finance, et cetera. Each one has its own dedicated server, database, and storage, and moving more to a virtualized approach where these entities are shared across all the business components.

On scenario two, the customer and use scenario B, we have customers who have three or more operating systems. They have 50 or more servers, and can move into an infrastructure with a BladeCenter environment running AIX, Linux, Windows, et cetera, virtual storage.

Again, EWLM will sit on the left-hand side, but you'll have more benefits in this virtualized infrastructure on the right-hand side.

We can take advantage of the SAN volume controllers, as well as some of the virtualization capability in the BladeCenters.

Customer and use scenario C, a function in EWLM version two is to manage the partitions on a partition level with velocity goals. So, here you see four different partitions on the left-hand side potentially being managed by four different administrators.

To the right-hand side where I have transaction response time and partition velocity goals, I can manage these environments through EWLM, and I can move the CPUs where my more important work is running automatically with EWLM.

On the next slide you see customer and use scenario D. Here is a complexity management, a customer expands a large pSeries servers installation and requires tools to monitor the resources, applications, and database transactions.

And here we're providing a holistic approach on the right-hand side where, again, I'm moving all of these tools and functions and management capabilities into the right-hand side. I look at the pSeries, clustering it with a virtualization technology.

I now take this information and use the VE console to give me a single console approach to managing my servers.

Customer and use scenario E, here again is a BladeCenter, hot migration and provisioning scenario. The customer would like to move from existing x86 servers to a BladeCenter without any down time. The BladeCenter will then be used more efficiently by sharing these resources between test development and production.

Again, what we're trying to do here is take advantage of the virtualization technologies within the servers and then use the virtualization engine management products to manage that. And on top of that, you have that virtual view using ~~the virtual,~~ the VE console, as well as the IBM TotalStorage productivity center.

On scenario F, what you see is a similar thing where I'm going to tie in the IBM director and EWLM to manage these Power 5 boxes. I'm going to take the information that IBM director and EWLM gets and provide that information up to the VE console so that I'll have a single view of my consoles of all the LPARs running on my Power 5 box.

On customer and use scenario G, again, what we're doing is we're taking advantage of the EWLM function that's allowed on the IBM Power 5 servers to automatically change the LPAR configurations, depending on the business goals defined in EWLM.

So on the left-hand side, I'm going to take a Power 5 box running with five LPARs, potentially multiple tools, multiple operating systems, and I'm going to move it into an enterprise-level workload management function.

I'm going to monitor these LPARs using EWLM. I'm going to use my business-base goals to see how my performance is working, and I'm going to use the EWLM interaction on the LPAR to move the CPUs from one LPAR to the next.

On customer and use scenario H, here shows the interaction between EWLM and z/OS workload manager as far as the goal management goes. On the left-hand side, you had a system-level workload manager, and that was your z/OS workload manager.

You had EWLM working on the left-hand side with your HP servers and your Web servers running on Power 5. If there's a performance degradation, perhaps it was in this back-end z/OS DB2 environment.

Therefore, you'd have an end user operator to go into the z/OS to manage more CPU for that DB2 environment. Now, with the integration of DB2 z/OS into the EWLM environment, I can now use EWLM's end-to-end technology to look at the performance from my Intel server to my Power 5 running WebSphere to my back-end z/OS DB2, looking at the end-to-end performance and to actually use service class information, the correlation with the z/OS workload manager to give more priority to the DB2 back-end so that I'm meeting my end-to-end business goals.

On the next slide, virtualization engine packaging and fulfillment, we want to make sure that people understand that there is no VE suite in version two. Each product is ordered as a separate entity. There'll be no trial package available for EWLM as it was on release ONE.

The offerings are comprised of the virtualization engine common components, which consist of the VE installer, the VE console, and various common run-time components, like WebSphere, DB2, ITDS and ISC and the following specific functions like the IBM director, the IBM VE EWLM, and the IBM VE RDS (or resource dependency server).

The products will be fulfilled and distributed via standard IBM series fulfillment channels, and the media will be sent either in CD or DVD format.

On the next chart we just showed you here how the labels will be looking, and you can find these labels on the various HTTP Web site on w3.rchland.ibm.com.

And what we've included here is the EWLM packaging information. So you'll see here for the EWLM managed server on an AIX CD, what the order numbers will be.

So, we can scroll down here. This is for your use, so if you have a specific customer that's interested in any of these environments, you know which CD that you would need to order, either on an AIX, Linux and Power i5 O/S, Linux on xSeries or a Windows.

So we'll scroll to the chart in the agenda that talks about EWLM values and conclusions. Now the EWLM values for your IT starts by monitoring the performance of your business goals. With EWLM, you will get end-to-end performance data in a centralized view.

You will get verification of your business goals defined and how well your business goals are being met. Basically you can call this performance data in a box. You will progress towards

business level network balancing by having EWLM influence your network routers as well as operational management.

You can balance your requests coming into your IT environment with a network balancer and manage the CPU resources to achieve your business-level goals. Now, the values for the business is you're going to move more towards an IT resource optimization.

Optimization and support, business processes and changes, tools to help optimize and enable an on demand operating environment, evolve from a system level automation to a business level orchestration.

EWLM will provide you infrastructure simplification as well as optimization.

On the next chart again we list the values of EWLM is really in the IT productivity. EWLM is an automated workload manager for distributed heterogeneous infrastructure. We'll manage business process service level and improve the utilization of IT resources.

We will rapidly understand the quality of service delivery, we'll identify the bottlenecks within a transaction and link those bottlenecks to a given server resource, and the performance expectations are really looked at from a CIO level because we're basing the service goals at a business level, not on a certain IT or a certain workload.

So, we're going to be able to evolve from system level automation to business level orchestration. And this will close the communication gap between your business as well as your IT.

Now, last on the agenda: we're including informational resources. These are HTTPs, URL sites, as well as information centers. We have identified some Redbooks and Red pieces that will help you understand the EWLM as well as the other virtualization engine components.

I thank you for the time, and this is the end of the EWLM presentation.

Female: OK. We'll stop right there. You were right at 70 minutes, but that's raw. You know, that's – once we get it edited it will probably 65 or so. So that was great. Do you feel OK?

Lynn Winkelbauer: Yes, I'm OK. My throat's sore so I'm glad I made it through it.

Female: Oh, I'm sure. Well, let's go ahead and open up everyone's lines. And don't worry about those stumbles. We'll take care of those for you. But I think you did a great job, so it went really well.

Lynn Winkelbauer: Now, is there a way to redo a chart? Because I noticed we have ERD just stuffed, explained two different ways.

Female: What chart number…

Lynn Winkelbauer: Oh, I just…

Female: …do we want to – because if we kind of now, you know, where we want to redo something.

Lynn Winkelbauer: On page 25…

Female: OK.

Lynn Winkelbauer: …we just say ERD.

Female: OK. How would you like to redo?

Lynn Winkelbauer: I'm not sure if Holger's on the phone.

Female: Do we have everyone's line open, Vickie?

Operator: Yes, we do.

Female: OK.

Holger Wittman: Yes, Holger's on the phone.

Lynn Winkelbauer: OK. Page 45, we have ERD but it's specified a different way, enterprise resource director.

Holger Wittman: Forty-five.

Lynn Winkelbauer: And there's another chart that had EWLM resource director.

Holger Wittman: Yes.

Lynn Winkelbauer: On 51 it says EWLM resource director. So before these charts get published, we need to make sure we have the correct definition.

Holger Wittman: Yes.

Lynn Winkelbauer: I'm not sure what it is. I thought it was enterprise but.

Holger Wittman: Yes. But anyway, if we have this abbreviation, Lynn, do you think that on page 51 you only used ERD as an abbreviation or do you really mention that we have been calling this ERDD EWLM resource director? Do you remember?

Lynn Winkelbauer: On page 51, I do not believe I said EWLM resource director.

Holger Wittman: OK.

Lynn Winkelbauer: But on the first, I forgot which slide I said, the first time it was introduced I couldn't remember the name of it and I wrote it in my click-to-add-notes.

Holger Wittman: Yes.

Lynn Winkelbauer: And I know I said it EWLM at that time. On page 25…

Holger Wittman: Yes.

Lynn Winkelbauer: …I wrote, EWLM resource director, and I did say EWLM resource director.

Holger Wittman: Yes.

Female: What we can do, Lynn, is we're going to have the audio file and the transcript sent over. I mean, once you get the audio file, I'm not sure, you know, if you'll be able to take a listen to it and know then if there needs to be an adjustment.

Lynn Winkelbauer: Yes.

Female: But that's, you know, that's always an option as well.

Lynn Winkelbauer: Yes, I think as long as the chart is correct and the speaker notes, most people won't listen to my voice anyway. You know, they're using the presentation, and when they're reviewing it, if the speaker notes are correct with the correct words, I'm not so concerned that, if I said it incorrectly one time and correctly another time.

Female: OK.

Holger Wittman: I would also agree because the presentation itself will stay longer than the audio. And therefore if the (yes) because if notes are correct, I think it's not a big problem if there may be some errors in using this abbreviation incorrectly.

Donna Fein: OK. So, if you need to – if you change the presentation then you have to get me an updated copy because I have sent this presentation on to Veronica who's doing the WOS production. So she will have to replace the current one if you need to.

Lynn Winkelbauer: OK. I think it would just be those words once we find the correct definition. And if it's spelled out incorrectly on a chart to change that.

Donna Fein: OK. There's no problem with that, but just get the correct presentation and then I will tell Veronica to replace it.

Lynn Winkelbauer: Not a problem. And this is who? Donna?

Donna Fein: This is Donna.

Lynn Winkelbauer: OK.

Donna Fein: But it has to be fairly soon because, you know, like I won't – I'll get the audios tomorrow so I'll need the presentation before, you know, by tomorrow so that I can – so that they can sync the, you know, sync the audio to the correct presentation.

Lynn Winkelbauer: OK. Not a problem.

Female: And, Donna, I need to run. I've got another record session right after this one.

Donna Fein: OK.

Female: So I'm going to get these notes into the editors, have them get working on it...

Donna Fein: OK.

Female: ...and get them to you as soon as possible.

Donna Fein: OK. Thanks, everybody.

Female: All right. Thank you. And, Vickie, you can stop the tapes and let (somebody) go.

Operator: OK. Thanks.

Female: Thanks. Bye, everyone.

Female: Bye, everybody.

Holger Wittman: Yes. Bye.

END