Application Performance Monitoring Demonstration Launchpad

Transcript

Application Performance Monitoring Demonstration Launchpad

Monitoring Launchpad Intro

Today's dynamic IT environments present a vast collection of components that require constant monitoring and management—hardware, middleware, applications, databases, transactions and operating systems. In the typical IT organization, operators and administrators can spend the day moving from one console to another, logging onto one agent after another, trying to track down the information they need to ensure high performance and availability.

Managing today's composite applications — the complex applications on which a business depends — requires a complete application management solution. IT organizations should look for an integrated solution that helps them track transactions from end to end, across multiple system types, then drill down into problem areas to rapidly diagnose and resolve problems. IBM Tivoli provides a comprehensive application management solution — including tight integration between application management and software development tools. This solution helps organizations proactively identify slowdowns, rapidly perform accurate problem diagnosis and help the appropriate staff correct the problem.

From the most granular detail to KPIs tied to Business Services, IBM Tivoli provides a comprehensive monitoring suite to efficiently monitor and manage your entire IT enterprise

Some of the typical challenges that customers face today include: the complexity of integration between point products, lack of a consolidated view of the IT infrastructure, and how to train staff on all of the point products. Customers also find that there is more manual effort required for maintenance and have difficulty accurately pinpointing performance issues in a composite transaction, which may be due to reduced availability or struggles in meeting stringent SLAs. There is limited functionality for proactive warnings and alerts, and it's difficult to obtain the hardware required to install each point product. Also, a lack of flexibility in licensing means that customers cannot switch licenses between different products to adapt to changes.

With monitoring solutions from IBM Tivoli you can get your payback in less than 12 months

We offer:

- A single solution with a common infrastructure shared by all the agents.
- A consolidated view of the IT Infrastructure.
- A reduction in training and maintenance needs through standardization on a common infrastructure.
- The ability to trace the entire transaction path and accurately pinpoint performance issues in a composite transaction.
- Deeper dive diagnostics where needed.
- Increased availability to meet stringent SLAs with preventive warnings and predictive analytics.
- And licensing flexilbility, meaning that a customer can switch licenses between different agents to adapt to changes and fully utilize the investment in IBM solutions.

IBM Tivoli's Application Performance Management and Monitoring solutions combine deep IT management capabilities with comprehensive monitoring functionalities,, delivering a truly integrated solution for monitoring and management across the entire enterprise. ITCAM solves the integrated management problem by offering a single set of tools that can help your organization optimize performance and availability at every level of the IT infrastructure.

With these solutions,, integration begins at the data layer, where a common data model enables a consistent view of information across all components and agents. This information is then consolidated in one easy-to-use role-based interface, the IBM Tivoli Enterprise Portal. This portal gives you single sign-on to all of the tools you need for server, application, and transaction management, application as well as providing advanced management capabilities.

IBM Tivoli's Application Performance Management and Monitoring solutions provide end-to-end server management across the enterprise by helping you manage service-oriented architecture (SOA) platforms, enterprise applications, Web services, databases, Microsoft applications, and operating systems.

They provide comprehensive transaction monitoring and management, from auto-discovery of transactions at the services and application layer to transaction tracking and analysis for faster problem resolution. These capabilities extend beyond your infrastructure to support your external partners, suppliers and customers, providing the detailed performance and availability information you need to deliver the service levels your customers require.

Tivoli has added a unique transaction tracking capability to allow better visibility into our end user experience and drill down to transaction level detail Tivoli has introduced a unique, single solution for managing the end user experience and enabled transactions for ease of implementation and acquisition for the customer. For greater application visibility Tivoli has 100 percent mapping capability to view the relationships between the infrastructure and business applications We can easily see IT assets and applications and their dependencies for a more rapid isolation of problems.
The advanced management capabilities allow your organization to build on your core competencies, expanding your management toolset to include strategic capacity monitoring, predictive analysis, and service level evaluation. These solutions also supports Tivoli Common Reporting, a shared infrastructure for creating, viewing and managing Tivoli reports, to help make communication more productive.

Reduce costs and increase your ROI
Easy to deploy and easy to use, IBM Tivoli's Application Performance Management and Monitoring solutions can give you quick return on investment right out of the box. It also offers you the flexibility to configure the solution to meet the unique needs of your organization. By consolidating your IT management functions in one solution, IBM Tivoli can help reduce complexity across your infrastructure, increasing the efficiency of your staff and helping you reduce IT costs.

With these solutions you can reduce the amount of labor needed to manage IT by up to10% and free up labor for new products, as well as reduce hardware costs for operations by up to 10%. You can also reduce incidents by up to 12%, increase availability by up to 7%, and reduce network downtime by up to 98%. Our single price point can reduce your cost of licensing by as much as 65% in an average Microsoft environment.

Tivoli invested significant development dollars in industry unique integration across the portfolio, in order to reduce the total cost of ownership of application management tools so customers can shift people and budget to other projects such as new application development.

Today, Tivoli's customers are able view and resolve problems and transactions in a virtual environment, whether using VMWare, System p aix, hyper v-, citrix, xen app, or system z. As customers implement new initiatives such as Virtualization, Energy Management, Cloud computing etc to create a dynamic and fleixible infrastructure, Tivoli's Application Performance Management and Monitoring solutions can support their current and future needs.

<transcripts of Expert Videos are viewable within the video and hence are not included here>

Demonstrations:

1 - ITM: Model Situation

This video presents how ITM622 helps us determine a situation threshold based on historical data or statistical functions.

In this scenario our company provides a mission-critical application server that must provide rapid and reliable service to our customers. We are required to monitor the response time of the application server on multiple systems. We would like to create a situation whose threshold will only fire when the value is outside the range for that managed system. We would also like to visualize the current and historical values for the system, perform statistical analysis on the date, and adjust the values before generating a Situation definition.

In this example there are a few things we'll have to consider before creating a situation.

The first is that, when using the model situation function, the administrator will need to determine the managed system to use to create an initial Situation. In this example we'll use the server IBM-E63D99ECF56, which is a typical, bare-bones application server for our environment.

Next, the admin will need to determine which metrics to use in the Situation. In our example, we've determined the response time is related to the application's Disk Writes/Sec – when Disk Writes goes up response time goes down, and vice versa.

Based on this information, the admin has created an Application Performance workspace with two charts that display the historical and current values of the Disk Writes/sec metric for an application server (of course, this example already has history configured for the table, and the TDW already has the recommended 7 – 30 days of detailed historical data – not summarized, since model sit doesn't operate on summarized data).

Optionally, the admin can switch to a product provided workspace displaying the metric we're interested in modeling.

To model a Situation using the metric, we right click on its bar in the chart (or point in a plot chart, row in a table, etc..) and select "Model Situation…". This will launch a dialog that allows us to choose the metrics used in our baseline exercise. Disk Writes/Sec will be selected by default, so we can just click OK. If we wanted to, we could change the metrics here.

After clicking ok, the Model Situation panel appears.

There's a lot to this panel; let's take a look at it… the top half is a chart showing the last 24 hours of data for the selected metrics. It defaults to a line chart, since they work best showing trends over time, but if we wanted to, we could change it to a bar or area chart by selecting one of the buttons in the toolbar (bar charts are good at showing holes in the data – any periods of time when no data was collected).

Underneath the chart there's a table with the metrics we selected, panels to change the time span and perform statistical calculations, and a button to create a situation.

The Condition column in the table has a default threshold for the each metric; the value comes from the object that was right clicked on to launch the model sit dialog. Notice when we select a metric in the table its threshold is drawn in the chart and the statistical table becomes enabled. The threshold is drawn as a line or shaded area in the chart and helps us visualize the condition - in this case it's disk writes/sec > 16… so the area above 16 is shaded; if we were to change it to < 16 the area below 16 is shaded.

The threshold value can be adjusted by using the value modifier drop downs, typing directly in the text field, or clicking in the chart.

The Time Span panel lets us change the period of time under consideration, both in the chart and by the statistical functions.

In this example the admin would like to base the threshold on the last weeks' worth of data so let's set the time span to last week. And in the function table, let's select PERCENTILE and AVG. Since poor performance in this case is associated w/ high values of Disk Writes/Sec let's set the PERCENTILE to 97 (this draws a line on the chart below which 97% of the data falls). And since the data values seem pretty spread out let's do AVG+3 std dev.

Click recalc. The statistical functions being used here are the same set available in the CLI commands tacmd suggestbaseline/acceptbaseline… in 622 the bit of code that does the statistical functions was refactored to live in the TEP Server, so now the CLI and the TEP client can share it. The TEP actually uses it in 2 places, in the model sit dialog we're looking at now and in a workspace view using 'add statistical baseline'.

When the chart refreshes we can see the data from the past week and the results of the statistical functions. The results are both drawn in the chart and in the table. The admin decides the result of the percent function looks good, and clicks the 'copy to' button to copy the 122 result to the Disk Writes threshold. This creates a condition that triggers on the top 3% of samples, based on the data from last week.

The admin clicks create sit.

Name: Application High Disk Writes
Description: the application is being delayed by a high disk write condition

The description is limited to 64 characters.

Click ok, sit editor appears,

The formula has been generated by the model sit dialog, with the defaults we see here – a 15 min interval, set to run at startup, and a critical state. By default the situation is only distributed to the managed system used to model it.

Click Distribution tab…

in our case we'd like to change the distribution to include the list of application servers in our env, so we assign the NT_Application_Servers MSL.

Click formula tab…

The situation has not actually been created yet, so if for any reason we change our minds we can click the cancel button. In this case let's change the interval to 30 seconds and then click ok, which will create the situation and start it on the set of application servers.

That's it - from here, the administrator might consider modeling the situation on different servers, or during different time periods. Unfortunately, in 622, the model sit dialog can't be used to create situation overrides for conditions like this, but the override value could be calculated using the model sit dialog and then set using the CLI or TEP override editor.

That's the end of the example… I hope we've learned how to create a situation from a workspace, and determine a threshold using historical data and statistical functions.
Thank you for watching.

2 – ITM: Add Historical Baseline

This video presents how historical baselines can be used in TEP's line charts for either trend or predictive analysis.

Our scenario starts with a report that the uploading to one of our application servers has performance problems around 7:30 – 8:30 am – the time when all of the stores of our retail company are uploading receipts from the previous day's business.

The administrator opens the Application Performance workspace and sees the spike in activity around that time. We'd like to determine if this is abnormal activity or business as usual, so the admin adds historical baselines and attempts to spot patterns or identify trends.

Next, the admin switches to the Realtime Application Performance workspace to check the current state of the app, and adds historical baselines to spot any abnormal activity or forecast future trends.

demo starts here…
Administrator opens up the Application Performance workspace, which uses a couple of plot charts to display the Disk Writes/sec metric for the problem server. The chart on the bottom displays historical data from this morning, and the chart on top tracks the current value in real time. In our example history has been configured for the table and the TDW already has the recommended 7 – 30 days of historical data (this needs to be detailed historical data, not summarized, since historical baselines do not operate on summarized data).

The admin see the delay reported in the problem report and would like to add historical baselines to perform trend analysis and determine if this is abnormal activity or represents business as usual… Notice the new set of tools on the toolbar… when 'add historical baseline' is pressed a dialog appears with a simple set of checkboxes that overlay the chart with data from a day in the past, so today's values can be compared with yesterday's, or the day before, or the past 5 days…

The historical data is overlaid on the chart by adjusting the timestamps to use the current day – for example, if it's July 31, 12PM and 'Show Yesterday' is selected, then data from July 30, 12PM is displayed at July 31, 12PM so today's results can be compared with yesterday's.
Let's select the past few days and click ok….

Data from the selected days are displayed on the chart and the admin attempts to spot trends or identify patterns. Since these baselines contain at most one day of data they work best spotting daily trends – when the time span of the originating plot chart contains less than 24 hours.

In this example the application appears to spike daily around 7:30am to 8:30am, so this problem appears typical for the time of day. The administrator might choose to add another application server to handle the morning traffic or write a Situation Override for this period.

Next, the admin applies the same set of historical baselines to the realtime chart on the top so we can visually compare the current value with the historical range for the system.

After the baselines have been drawn we can tell if the current values appear normal as they do now,

Or we can tell when the metric is behaving erratically (do magic here to spike the metric) and something might be wrong,

And we can also use the historical baselines to predict future trends. For example, we can forecast what will happen today at 5pm by changing the chart duration to include that period of time….

Here we can predict the metric will start trending upwards as 5pm approaches and expect it to stay pretty high until 6 or so, based on the behavior from the past few days.

This ends our example – hopefully we've learned how historical baselines can help us identify and predict daily trends.   Thank you for watching ;-)

3 – ITM: Add Statistical Baseline

This video presents how ITM622 can help us determine when a metric is outside its normal range for a system.

Our scenario starts with a report that the uploading to one of our file servers has performance problems during the hours of 9 - 10 am – the time when all of the stores of our retail company are uploading receipts from the previous day's business.
our administrator will switch to Historical Application Performance workspace to determine if the metric is outside its normal range for this server (IBM-E63D99ECF56) by using 'Add Statistical Baseline'

So let's look at this example in more detail…

Live demo starts…

Administrator opens up Historical Application Performance workspace for the problem server with a plot chart displaying historic results for Disk Writes/sec metric. Please note that historic data was configured for this attribute group and TDW was populated with a month worth of data. Administrator notices the spike in activity around 9am. He can adjust the historical time span to include a different time period.

Administrator presses "Add Statistical Baseline" tool that opens up a dialog with a list of statistical function to select from.

Several statistical functions are available:

- Range between minimum and maximum
- Average
- Minimum
- Maximum
- Percentile
- Mode (most frequently observed value)

This is the same set available in the CLI commands tacmd suggestbaseline/acceptbaseline… in 622 the bit of code that does the statistical functions was refactored to live in the TEP Server, so now the CLI and the TEP client can share it. The TEP actually uses it in 2 places, in this baseline and in the model sit dialog.

In our example the administrator selects Functions AVG and MAX by selecting checkboxes. Selecting a checkbox will enable a function and clicking Apply will both populate the Result column and draw the function in the chart. Administrator also adjusts time Span to Last 1 Week.
When administrator clicks OK the AVG and MAX values for the last week are drawn on the chart. The ITM administrator visually compares the data from this morning to the MAX and AVG values over the past week.   In this example the values from this morning are way above average, almost reaching the maximum value from last week.

Please note that Statistical Baselines work with detailed (not summarized) historical data from the data warehouse.
'Add Statistical Baseline' can be used in area or plot chart views. For plot chart views the 'Select Plot Points' property must be 'Attribute(s) in first row'

Thank you for watching.

4 – ITM: Add Monitored Baseline

This video presents how ITM622 can be used to draw Situation thresholds in a chart. A user can see in real time how a metric compares to its threshold and anticipate when events will be raised.

Our scenario starts with a report that the uploading to one of our file servers has performance problems around 9 - 10 am – the time when all of the stores of our retail company are uploading receipts from the previous day's business.

First, the administrator switches to the Real Time Application workspace to check the current state of the system, and adds a monitored baseline to track how the system is doing compared to its threshold.

Next, the admin adds these baselines to a historical chart to gauge how effective the Situation has been over time, identifying when the threshold is set too low and the situation is constantly open, or set too high and the situation never triggers at all.

Let's look at this example in more detail…

Administrator opens up the Real Time Application Performance workspace, which has an event console and a line chart displaying current results for the application. The administrator wants to compare the current values to the 'Application High Disk Writes' Situation that was created in our previous example.

Notice the new set of tools on the chart's toolbar – one of these is 'Add Monitored Baseline'. When the button is pressed a dialog opens with the list of Situations that have been distributed to the system and use an Attributes from the chart. You can see the situation name, its status, whether or not overrides exist, and the Situation reference formula. You can edit a Situation or its Overrides by selecting a Situation and using the buttons on the toolbar, or by right clicking on a Situation in the table.

It's important to notice if an Override Exists for this Managed System because in 622 Monitored Baselines do not support displaying a Situation Override, only it's reference formula. If Overrides do exist their values will not be drawn in the chart.

This limitation is being addressed in the 622 follow on release.

In our example the Administrator selects Application High Disk Writes and the threshold value is drawn as a line on the value (Y) axis at 178. The situation predicate is Disk Writes / Sec *GT 178, so the Situation can fire if the chart line goes above the red marker (assuming the Situation is evaluated at this time). Of course if the predicate were *LT 178 the Situation could fire if the line dropped below the red marker.

We can simulate the condition now to observe this in real time…

The situation should trigger if it's evaluated while the chart line is over the red marker at 178 (the threshold in this case). Our situation runs on a 30 second interval so we shouldn't have to wait long…

When the situation fires we can see the red light in the tree and the event in the console.

And when the condition clears the situation will close and the light turns off in the navigator tree.

Now, let's look at how MB can be used on historical charts to analyze how effective a Situation is over time. One of our systems is state of the art with faster storage and lots of memory. The administrator opens the Historical workspace for this system, adds the same situation as a baseline, and notices it appears high since the values on this system are much faster than others. After changing the time span from a few hours to a few weeks we can see this situation will likely never fire on this system.

The administrator might consider creating an override for the faster systems using the Override editor or CLI. This would allow the threshold to be dynamic, and depend on the system it's running on. This is better than creating new situations for a few reasons – it takes half as much processing power to evaluate 1 situation instead of 2, and it simplifies maintenance by reducing the # situations in the environment.

Thank you for watching.

ITCAM for Applications 6.2.2 PeopleSoft Agents Demonstration

Introduction

Welcome to the IBM Tivoli Composite Application Manager (or “ITCAM”) for Applications 6.2.2 PeopleSoft Agents demonstration. Today we'll be providing a brief overview of the PeopleSoft agents followed by a demonstration highlighting key agent capabilities.

In this video, we are going to show you how these new agents dramatically improve an organization's ability to monitor key PeopleSoft systems. The agents provide deep monitoring capabilities of PeopleSoft components by leveraging ITM-based expert advice, “take actions”, situations, historical data collection, and integration with the Tivoli Enterprise Portal.

The PeopleSoft agents help the data administrator to respond to requests by providing information about the status of the PeopleSoft system. The PeopleSoft agents include over 70 out-of-the-box solutions that help administrators reduce alerts, raise confidence and avoid incidents.

“Best practices”-based expert advice provides the knowledge that administrators need to quickly resolve critical issues in their PeopleSoft environments without having to rely on subject matter experts.

A variety of PeopleSoft-specific out-of-the-box “take actions” extend ITM by providing administrators with the ability to control key PeopleSoft services. The wealth of PeopleSoft specific "take action" policies and metrics can be combined together into powerful ITM-based policies that can be invoked automatically or manually depending on the administrator's preference.

The agents support the monitoring of PeopleSoft running against PeopleTools 8.48 and 8.49 in combination with DB2, Oracle DB and Microsoft SQL databases. The agents are available for AIX, Solaris, HP-UX Itanium, Linux Intel (32-bit), and Windows (32-bit).

Next, we are going to transition into a series of video demos. First, we'll walk you through a basic overview of each agent. Following that, we'll walk you through two specific PeopleSoft scenarios.

Demonstration

Let's take a tour of the PeopleSoft domain and Process Scheduler Monitoring Agents. On the top level node, you will find basic information about CPU and memory utilization along with the status of critical application domain processes.

The first workspace contains information about analytic servers. The availability workspace contains application domain process details such as CPU, memory consumption, and availability.

Agent messages contain diagnostics about the monitoring agent as well as a view of the agent's own log file. The client node contains information about the number of connected PeopleSoft clients. The configuration node has information about the application domain’s configuration values including supported languages, enabled services and many other details.

The PeopleSoft Application Domain Monitoring Agent also monitors the logs of the PeopleSoft application domain and Tuxedo middleware. The scheduler's workspace shows a dashboard view of all process schedulers, to which the configured application domain has access.

Alternate workspaces on the scheduler node provide domain wide information about run counts and distribution status. These workspaces also have the ability to link to individual process schedulers for a more granular view of the data. And the final node contains specific details about the Tuxedo services running under the application domain.

Let's move on to the next monitoring agent. The primary scheduler monitoring agent node contains a dashboard view of the PeopleSoft process scheduler's CPU and memory consumption. An availability node contains per process memory and CPU utilization details. The batch process node contains multiple layers of information about the specific batch process being monitored.
There are three alternate workspaces containing process load information for different batch processes deployed by the scheduler. The run status counts workspace shows the exit status of previous jobs. The distribution status counts workspace displays the distribution status of executed batch processes.

Finally, this workspace provides a view of any jobs that have exceptional status. It shows the name of the process and other high-level information. Each view has a link to other views that contain more specific details about those processes. A configuration node provides information about the process scheduler's configuration.

The logs node provides information about scheduler logs; the master scheduler log and the scheduler log are shown here. The Tuxedo workspace indicates the status of Tuxedo keys.

Let's see how the monitoring agents are useful for administrating in a PeopleSoft environment. This is the enterprise view provided by Tivoli. Note that there are two situations that have fired.

We'll investigate by finding the situation on the top level node and begin to drill down. As we follow the situation down through the nodes, we will eventually reach the availability node. A glance at availability reveals that there are no running PeopleSoft processes.

So now we navigate to the situation by clicking on the fly-over link. Now, we can see more detail about our situation along with some expert advice providing a suggested solution to the situation.

The first bit of advice informs the user that he may use a take action to restart the domain. Click on the drop down box and select the appropriate situation. Make sure to click on the correct monitoring agent. Hit run. After a few short seconds, a dialogue box will inform you of the user's take action success. A minute or so later, the PeopleSoft domain will finish booting and the situations will clear up on their own.

The Scheduler Monitoring Agent is useful for examining your environment's batch processes. Let's look at a scenario where a monitored batch process scheduler has a job with an exception. First we'll navigate to the Scheduler Monitoring Agent's batch process node.

The views reveal several successfully run batch jobs and one error. To investigate the error, we will click the batch process node and select the batch process exceptions alternate workspace.

We have navigated to a workspace that gives us some high-level information about the failed job. Following the link attached to the record, we will navigate to one more workspace. This one contains all of the detail an administrator would want for a failed batch process.