APM Description / Dictionary

Thursday, January 31, 2013

10:02 AM

CA Application Performance Management which includes CA Introscope and CA Customer Experience Manager.

APM (Application Performance Management) is a tool that monitors processes that occur during transaction processing of applications.  It provides real time monitoring and alerts of stalled processes as well as gathering data that can be used to by development to improve the flow and speed of transactions.

Introscope, CEM, and ADA make up the APM tool suite.

VAST ID 23753 Acronym VAP - EWS

Verizon Application Performance - Early Warning System for our APM tracking.

APM Suite of Tools:

Introscope – (formerly known as Wily):         Is the base Agent that is deployed to an application server and is embedded within the actual running JVM.

ADA - Application Delivery Analysis :             Operates at the network (TCP/IP) traffic level to provide capturing of traffic packets

CEM - Customer Experience Manager:         Allows viewing / recording of a transaction real time.  It also serves as the component that merges the Introscope/ADA components together.

Acronyms

VAP-EWS         Verizon Application Performance – Early Warning System                       Internal tracking application in ITAM/VAST consisting of a Portal (internally developed web tool) and a vendor

                                                                                                                                    suite (APM) along with the hardware/software

MOM:             Manager of Managers, Load balances across EMs, front end interface for users

EM:                  Enterprise Manager, Primary interface for the application metrics traffic (Introscope Agent)

TCS:                 TIM (Transaction Inspection Monitor) Collection Services, The process that receives the TCP/IP traffic from the MTP

EEM:                Enterprise Entitlements Manager, Provide logins at user level via integration with LDAP

CDV:                Consolidated Data Viewer, Enables visibility across MOM/EM clusters

Portal:             VAP-EWS allows you to see enterprise-wide dashboards

MTP:                Multi-Port Collector, Appliance (server) used to capture TCP/IP packets which are sent to it from the network switch

EHNET:            Enterprise Host Network                                                                    Original VZ Telecomm Network

IDN:                 Internal Data Network                                                                        Original VZ Business Network

Java processes handle APM functionality

APM uses Isengard communications protocol

HW is a mix of LINUX and Windows

EM Serves as engine, co-ordinating hub

To process the data received from the agent and send it to the workstation.

Saves data to SmartStor, which is an external data store for data analysis.

Generates alerts and actions, which are notifications based on the specified threshold values.

IntroscopeEnterpriseManager.properties – This file configures EM, SmartStor, PostgreSQL databases.

MOM – EM Manager Of Managers collects information from all collectors in the cluster. Workstations log in to MOM to view all metrics from every collector.

SmartStor - EM stores metrics in SmartStor. SmartStor is an embedded database that runs inside the EM. It is tuned for storing Introscope metric data for later historical analysis. SmartStor stores data in two primary formats, one optimizes for fast writing and another for fast reading.

Introscope agents - collect performance data from application in the JVM and .NET Common Language Runtime(CLR).Agents send data to the CA Introscope EM. You can use the Introscope workstation to view the data.

The Java agent enables Introscope to collect minute details about how your applications are performing. Data is collected from your applications by the Java agent. The type of data collected depends on which PBD files you select to implement. Several standard PBDs are included whe u install the Java agent and specific PBDs for your application server.

IntroscopeAgent.profile – this file configures the the agent.

JVM AutoProbe – inserts probes into bytecode during class loading as a method of invoking Probebuilder.

Uses Probebuilder directive in .pbd files to define what to monitor.

Monitors J2EE environments out of the box using the default .pbd directive files.

CEM – View of the app

Allows u to see where the latency is. Knows end customer desktop, can see the timings on all packets going back and forth. Deep inspection tool, code level, java, html based, weblogic, writ

Sometimes a component may perform poorly because a component ir calls perform poorly. When you know how to use the blame technology, you can trace a performance problem to its root cause.

Traditional blame (3-20) : In this transaction a servlet is called which calls a session EJB. The session EJB calls an entity EJB, which picks up a JDBC resource and interacts with the database. The results from the database are then process by the entity EJB and sent back to the stack and parsed out as HTML.

Workstation Investigator – present a simplified blame stack – only the front-end component and a backend system are represented as blamed components in the Workstation Investigator tree. This indicates whether a response time problem is internal to the application server or external. This feature is referred to as boundary blame and is enabled by defaut.

Frontends Metric Category – Frontend metrics are reported in multiple levels of granularity:

For each application

For each URL in an application

For each backend system requested by a URL

Backends Metric Category – Metrics from each backend include:

Average Response Time

Concurrent Invocations

Queries per interval

Stall Count

Error Per Interval

Q&A –

Q. What kind of protocol is used between agent and MOM

A. The protocol is called "ISENGARD". communicates over port 5001

Introscope provides alerts to keep you informed when things go wrong. It also helps with root cause diagnosis and triage. CEM provides more information to help you prioritize your work.

CEM provides information on –

Actual affected end-user populations

Comparisons to stated SLAs

The performance of transactions

The quality of transactions

The cost of missed transactions

Case study

You just released a new product and want to ensure that the web transactions are completed within an acceptable time.

Some transactions are completed within an acceptable time and some are not, but you don’t know why. The response time for certain customers is too slow, which is causing big issues.

In this situation we can use Introscope metrics

Indicate slow response times per interval

Can point to a back-end problem, if there is one

CEM metrics

Can indicate user populations that are experiencing slow transaction times

Indicate the performance and quality of transactions

Can quantify the cost of lost transactions.

APM as a whole correlates the slow transaction time information to help find the root cause diagnosis triage.

I have looked at the historical data from the agent on vpbfrdpa14, and can see what the problem was.  The heap size was approaching exhaustion, and each time the garbage collector ran more and more often until it couldn’t keep up.  It happened even after someone boosted the size of the heap.

There are 2 components to this product – Vaidehi did the Introscope part of it and the other part is CEM which I am going to talk about. CEM is Customer Experience Manager. User experience wherever they are using it, using ur UI, in that process they are conducting an action, those are actions that are captured within CEM. They do not require an agent, they do not require anything on User end. This is all thru package sniffing. We are doing an analysis of the package, apply rules to that pkts based on SAN ports that are setup and we are capturing that information. Introscope captures what is going on inside the application. Introscope and CEM are tied together by a dashboard. So, we start with the dashboard.

VEC is running on a series of production servers, there is JVMs, you would use Introscope to correlate

CEM – View of the app

Allows u to see where is the latency. Knows end customer desktop, can see the timings on all packets going back and forth. Deep inspection tool, code level, java, html based, weblogic, writ

CEM reads http from your switch and monitor what customer is seeing. Map that with app service team and then see the complete transation from User to BE. Identify the transactions uniquely.

TIM & CEM data Collection

There are multiple locations within your network where it is possible to locate CA CEM and alternative methods for connecting the system to the network. This chapter describes the options.

TIMs and the Enterprise Manager reside on separate hardware. The primary consideration is connectivity to the TIMs that report to the Enterprise Manager (Collectors and Manager of Managers (MOM)).

There are multiple options for where you place a TIM; they are described in [About CA CEM insertion locations](file:///C:\Introscope\Bookshelf\APM-Bookshelf-9.1.2\Bookshelf_Files\HTML\APM_Config_Admin_EN\1473595.html). The location of a TIM determines what information you can collect.

The preferred insertion point for TIMs is logically between the network load balancer and the web server aggregation switch, or between the load balancer and the web servers if no aggregation switch is used. For more information, see [Between the aggregation switch and the network load balancer](file:///C:\Introscope\Bookshelf\APM-Bookshelf-9.1.2\Bookshelf_Files\HTML\APM_Config_Admin_EN\1473595.html#o1473598).

When you install a TIM in the location of your choice, you must also select a method of connecting it to your network.

Memory Leak – Memory Leak is a particular type of unintentional memory consumption by the Java virtual machine (JVM) where the program fails to release memory when no longer needed.

Failure to release memory when no longer needed leads to memory exhaustion.

Orphaned objects can also exhaust JVM memory space.

Available memory steadily decreases until no memory is left in the JVM environment.

Response time becomes progressively slower.

The out-of-memory errors begin to be reported from JVM.

A system lockup is foreseeable.

Without resolving memory leaks, your application and the system in which it runs can freeze or hang.

Memory leaks in the Java application can be located using CA Introscope Leak Hunter.

As the response times become slower and out-of-memory errors appear, restarting the system can resolve the problem-but only temporarily. The program components or orphaned objects that are causing memory leaks will continue to use exhaust memory, resulting in an increase in bytes in use in the GS heap. Ultimately, if unresolved, memory leaks will result in your application and the entire system freezing or hanging.

Throughput – The percentage of total time not spent in GC, considered over long perioeds Garbage collection overhead. The inverse of throughput, that is, the percentage of total time spent in GC.

Pause time: The length of time during which application execution is stopped while garbage collection is occurring.

Memory is allocated in two regions:

Stack:Memory is allocated for local variables when a method is invoked and released when the method is exited.

Heap:Memory is allocated for all objects created with the New Keyword.

If the JVM cannot obtain heap memory to allocate more objects, they display an out-of-memory error.

GC is the process of reclaiming unused memory in the heap so it can be used by the JVM to allocate more objects.

STACK is a location in computer memory where all the variables that are declared and initialized before run time are stored.

HEAP is the location in computer memory where all the variables created or initialized at run time are stored.

Introscope displays the following metrics for each operation in WebSphere Portal:

* **Average Response Time (ms):** average time taken for executing or invoking a specific operation or action.
* **Concurrent Invocations:** how many requests are going through at a given time (throughput).
* **Errors Per Interval:** how many errors are recorded by the Introscope Agent by catching a Java or exception in a 15-second time slice.
* **Responses Per Interval:** the number of invocations that completed during a 15-second time slice.
* **Stall Count:** how many operations, invocations, or calls are taking longer than 30 seconds in a 15-second time slice.

In the Investigator tree, under the WebSphere Portal > Portal node, you can see Pages and portlets that belong to the Portal. Under the *WebSphere Portal > Virtual Portal > <VirtualPortal\_name>* you can see pages and portlets that belong to the virtual portals. Under the *WebSphere Portal > Portlets* node you can see all the portlets for the WebSphere Portal.