**DYNATRACE**

* once the test is completed derived statistics are not meeting expected statistics

we have to find the root cause …

* for application side metrics monitoring we have used JVM, JMC, JCONSOL but using this profiling tools we can’t identify
* which transaction is executing which method?
* What is the response time of particular method?
* for suppose 100 methods are executing in this we don’t know which method executed by which transaction? Weather it is launch or login.
* We can’t map between **method** and **transaction.**
* To overcome this issue, we can use APM tools like Dynatrace, App D.
* Among both tools Dynatrace is good.
* Current Dynatrace version is 7.0.
* Previous Dynatrace versions are 4.5,6.5,7.0.

**APM:**

Application performance management is the art of managing the performance, availability, and user experience of software applications. APM monitors the speed at which transactions are performed both by end-users and by the systems and network infrastructure that support a software application, providing an end-to-end overview of potential bottlenecks and service interruptions.

**Features:**

**End User Experience Monitoring:**

The first dimension of APM is End User Experience Monitoring (EUEM). It’s a term used to describe the multiple approaches to monitor what an end user might be experiencing.

The company divides its EUEM capabilities into 3 products: “User Experience Management”, “Synthetic Monitoring” and “Data Center RUM (Real User Monitoring)”. Don’t be scared of the names, they are meant to sound explosive and expansive while they simply monitor the user experience from different perspectives.

* The first tool, “User Experience Management” offers a browser-based JavaScript injection and SDK approach for mobile apps that captures metrics related to user requests and sessions, across different browsers and devices. For example, you can search for specific user actions in order to get a history of their performance / how long did they take to execute.
* The “Synthetic Monitoring” service displays stats from different locations around the world. It’s also helpful for simulating test executions, comparing performance across problematic behavior patterns, monitoring host availability and third party services or even simulating high volume traffic.
* “Data Center RUM” is the last solution which offers an agentless approach to end user monitoring by sniffing incoming HTTP requests (and other protocols) so it can piece together the latency of end user requests without needing to explicitly instrument web pages or native mobile apps.

**What is Dynatrace**

* **DynaTrace** Software is an **Application Performance Monitoring  tool (APM)** ,Which is widely used nowadays .It comes with advanced features for monitoring Java. Through which we can easily identify the performance of our application.
* We are using it to discover the existence of abnormalities in CPU Performance, response time, transaction rate, throughput and system usage.It has helped to diagnose and fix many performance issues at an early stage and make our application more valuable.

**Dynatrace Working:**

* Dynatrace server should be installed on your system and it consists of host and agent . It has a dashboard through which we can diagnose our system on one place.
* Our server can easily interact with other agent if we want , for that we have to run some command  through terminal  and then we can easily watch the user interaction and also we can customize the time.

**Dynatrace Architecture:**

****

* First, we have to install Dynatrace collector in server next Dynatrace server in a server and you can add multiple DT servers and you have taken one more mission so I would like to store old data Ill make it as Session i.e. install D.T session storage
* Dynatrace is agent Based tool.
* For application information we have Dynatrace client(**Client** is a visually rich user interface that provides intuitive access to the many facts of your application's performance and system health data captured by dynatrace)
* Being an end user ill access Dynatrace client to fetch the data from Dynatrace server which collected by Dynatrace client which is informed by Dynatrace agent.
* dyna agent (it’s a onetime activity)
* Suppose let’s take PMS Application for Monitoring Purpose.
* First, we have to tell dyna admin to on board my application
* Dynatrace admin will ask our application App server, web server, D.B server, I.P address and Username and Password.
* Dynatrace admin will install the 3 Dyna agents in web, app D.B servers.
* We can add multiple applications to Dynatrace.
* Dynatrace collector will collect information from every agent.
* It will send to the Dynatrace server. (in real time we have so many dyna servers).
* Then I would like to store some offline data the old data I will make it as dyna session storage
* He will create profile in the Dynatrace, so this Dynatrace has capability to collect the information from PMS application web server, App server, D.B. servers.
* We have to convert L.r scripts into Dynatrace scripts.

**How to convert Lr script to dynatrace script**

**Before lr12.55:**

Dynatrace 🡪 Goto tools🡪load runner script converter🡪browse the file and choose the required script🡪select the all the request and path them.

**After**

Goto replay vugen🡪runtime settings🡪preference🡪advanced🡪enable dyntrace monitoring

* Add Dynatrace headers on top of the requests this header has (transaction name and method name),
* Dynatrace header parameters

SN🡪script name.

VU🡪 virtual users.

AN🡪URL address

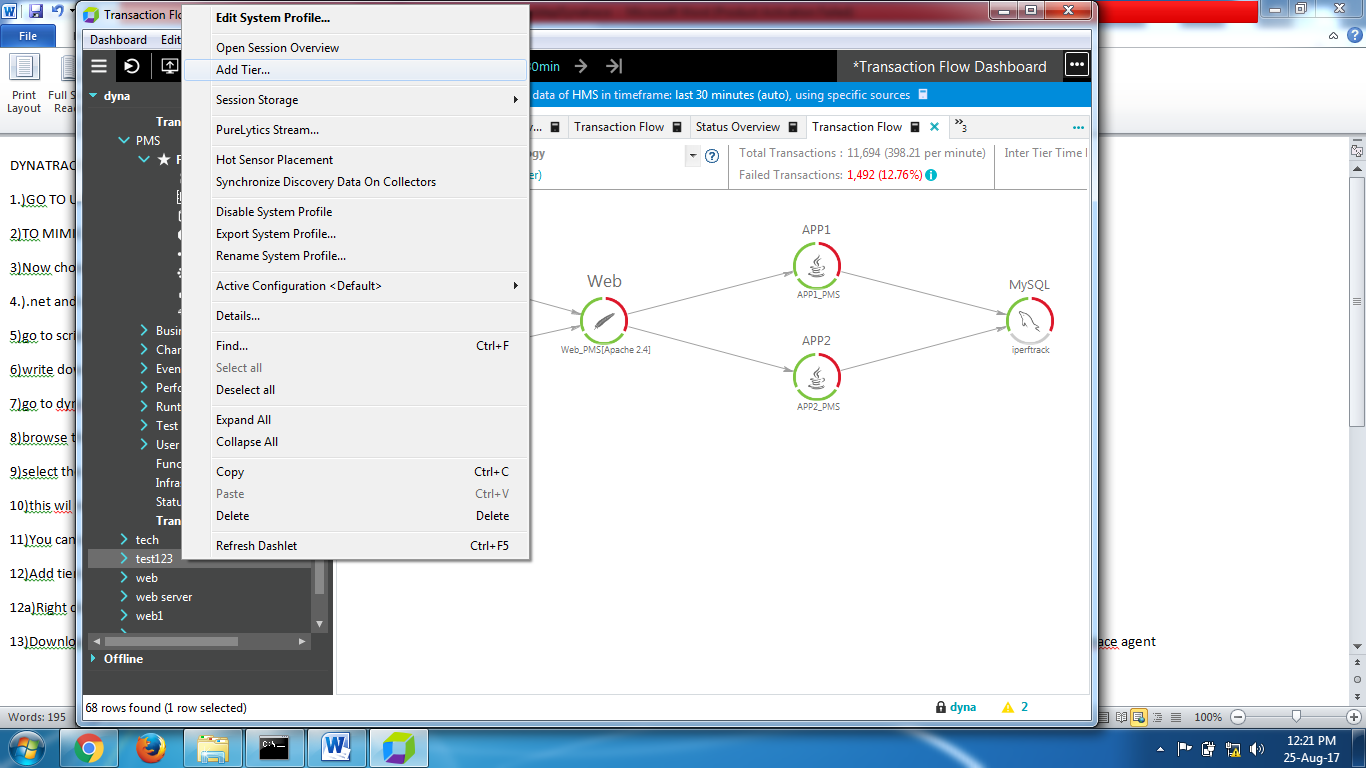
PC🡪URL.

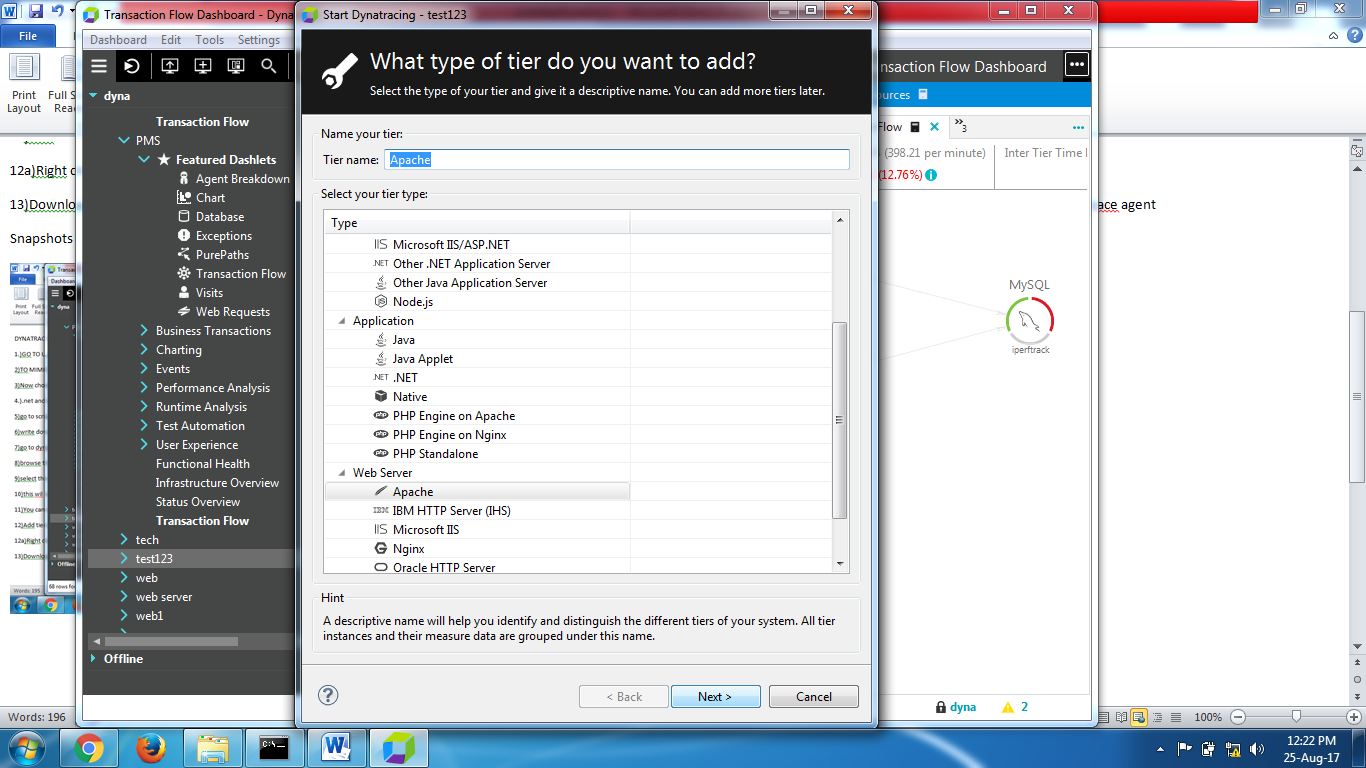
ID🡪Time in mili sec.

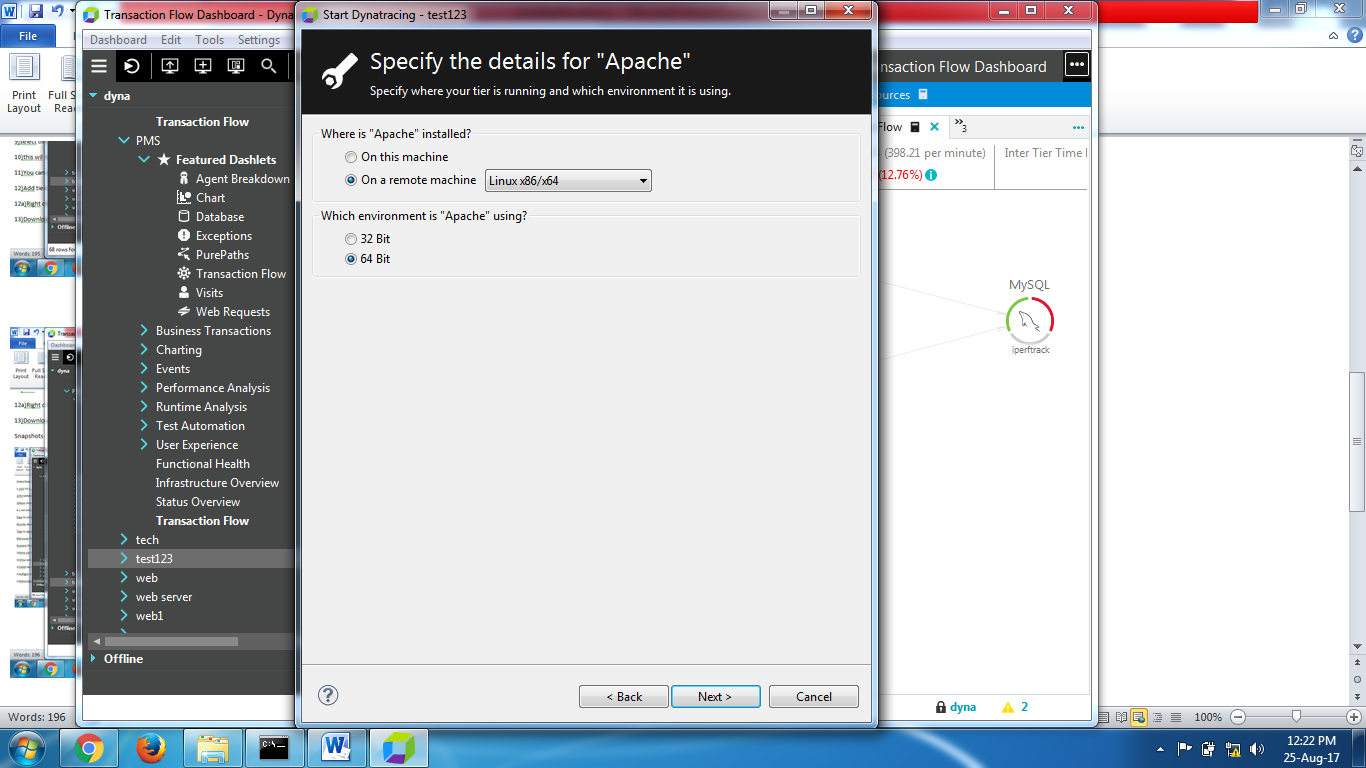
NA🡪Transaction name.

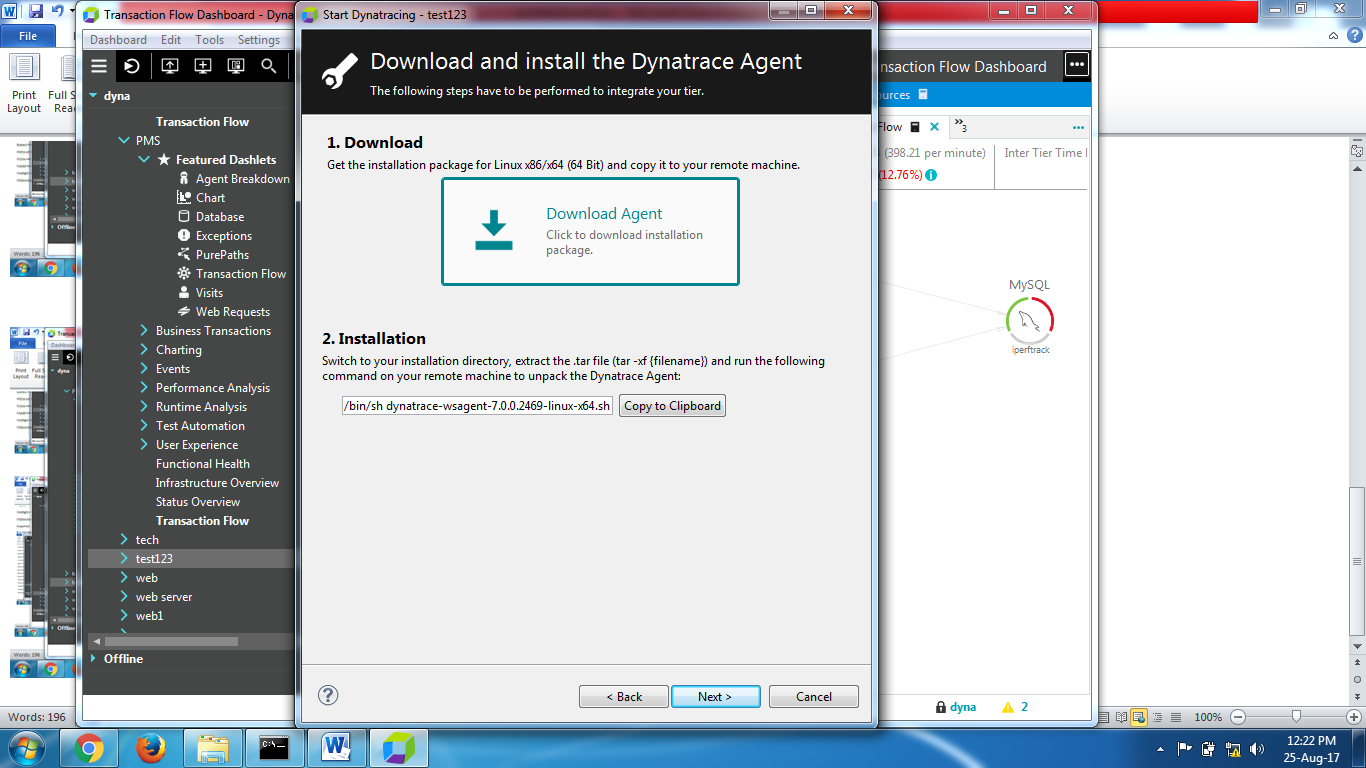
* Using Dynatrace we can monitor web server,C.P.U, memory, Disk, thread dump, memory dump, connections
* So, we can monitor 3 ways
* Infrastructure level =CPU, memory, Disk
* App level=Code level
* Db. level= query level
* Push the converted script into controller

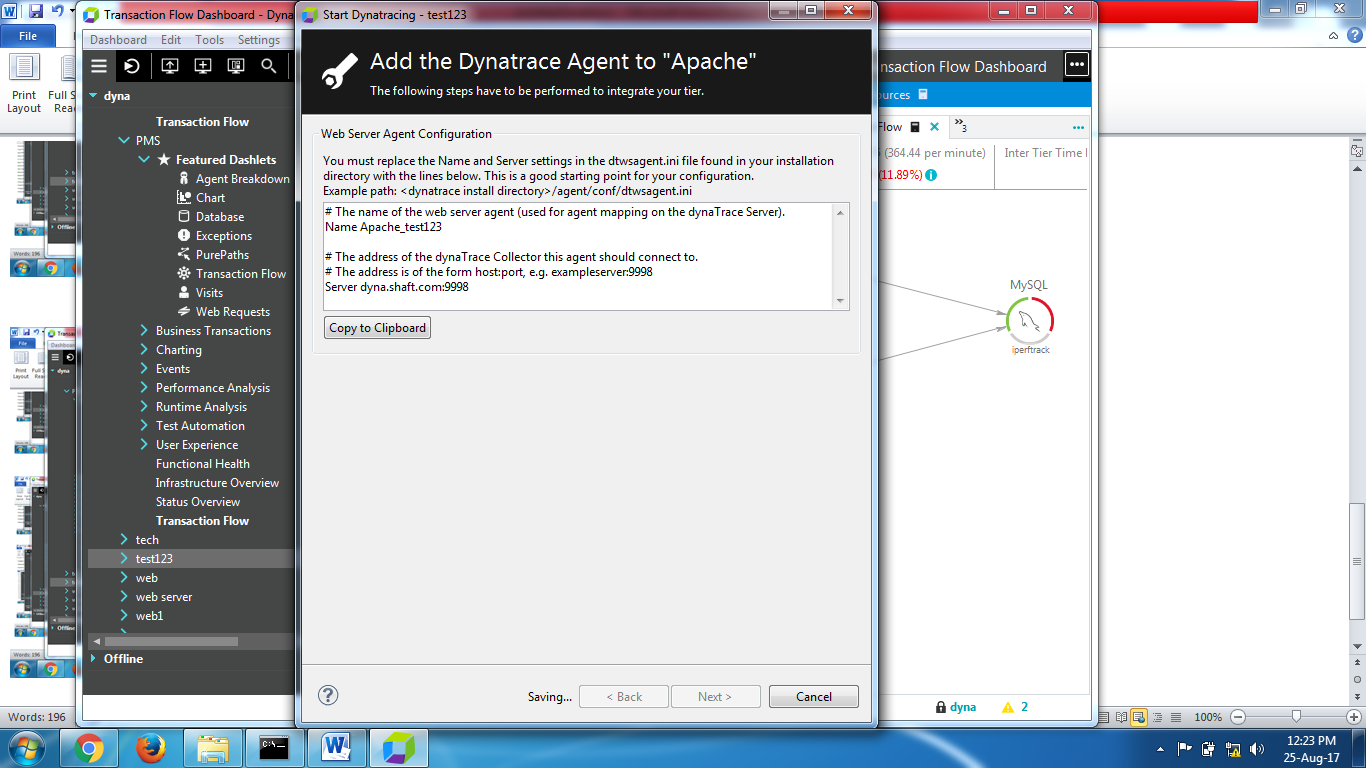
**Configuration of all servers in Dynatrace:** Create system profile🡺 we need all servers i.p address, usernames and passwords, hostname those are deployed in Linux o.s machines.

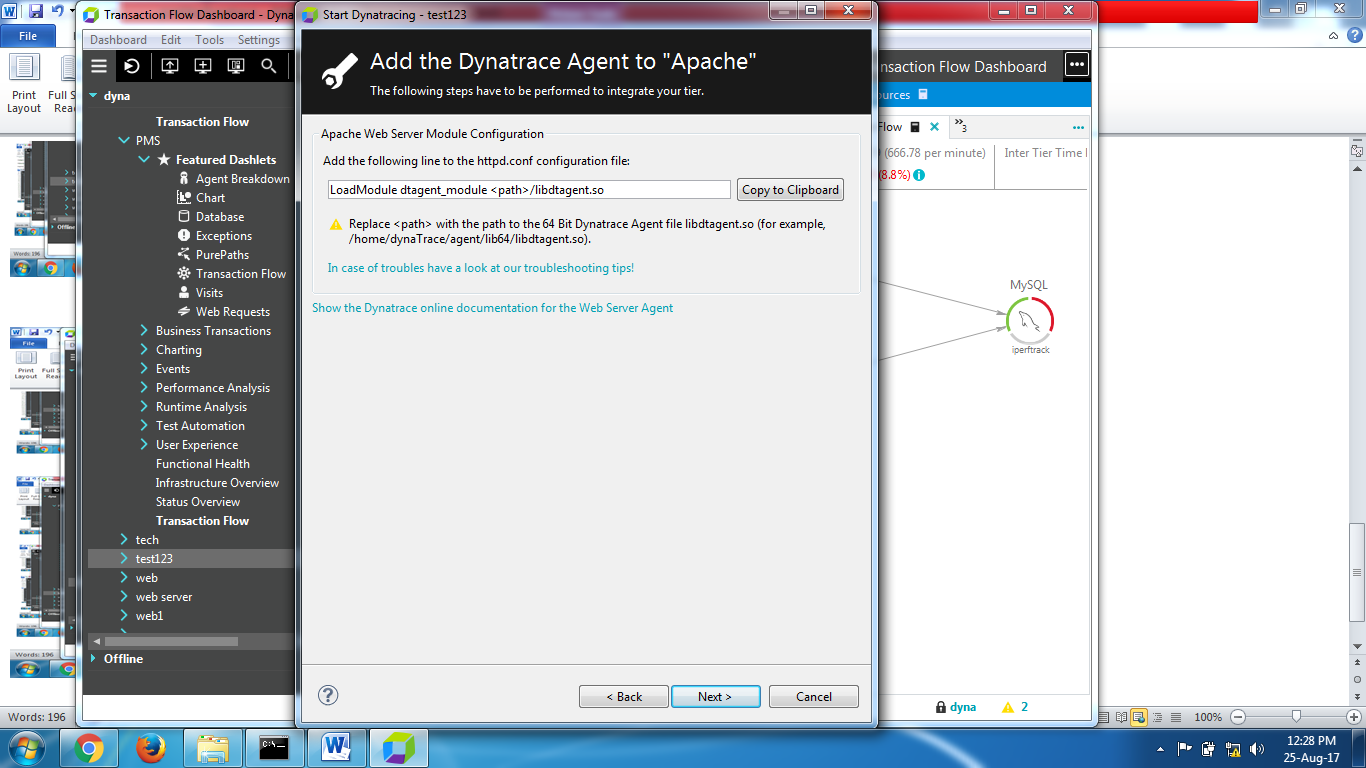
* Click on system profile 🡺 right click and create profile
* Name = Facebook
* Add tier
* Tier name = web server
* Apache server
* 64Bit
* It will down load the TAR file
* Copy this TAR file to shared drive to web server
* Open web server and connect Putty
* You can mount this TAR file to here
* Change access level CHMON, -EZ
* You can extract and execute it in web server
* If you execute this TAR file in web server it will configured I the Dynatrace.

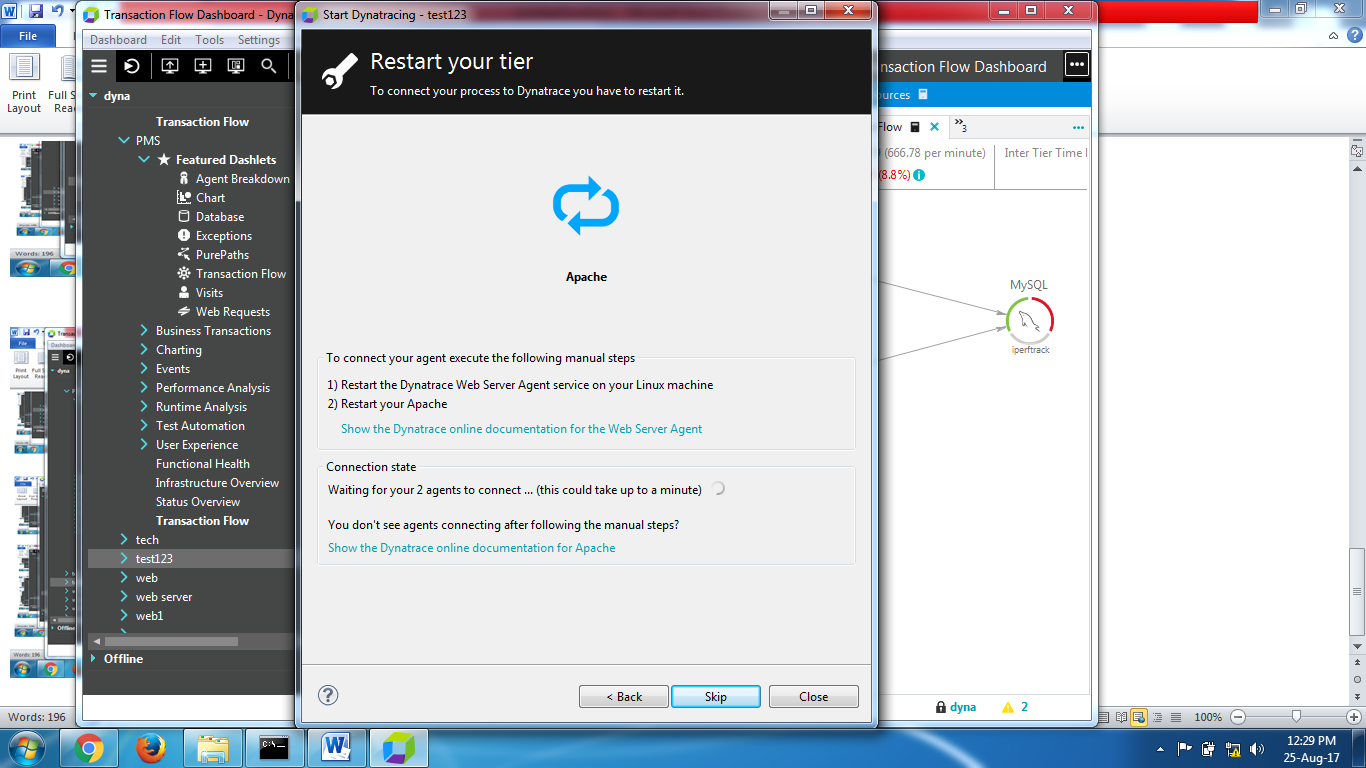


****

****

****

****

****

Next remaining server’s also same process for configuration

* Click on system profile 🡺 right click and create profile
* Name = Facebook
* Add tier2
* Tier name = App server
* Tom cat Apache server (remote machine) means it deployed another machine in real time
* O.S Linux and java version of the application is 5.0 above
* 64Bit
* It will down load the JAR file
* Copy this jar file to shared drive to App server
* Open web server and connect Putty
* You can mount this JAR file to here
* Change access level CHMON, -EZ
* You can extract and execute it in App server
* If you execute this JAR file in App server it will configured In the Dynatrace. As the same way D.B configuration also.

**PURE PATH:** PurePath technology uniquely captures and analyzes every single transaction end to end across every tier of your application technology stack—beginning at the browser and extending through all services, all the way down to the code and database level.

first of all you have to differentiate the breakdown of PurePaths and the breakdown of methods:

* PurePath breakdown: this breakdown relates to the sum of all methods/subpaths, NOT the response time and NOT the duration
* method breakdown: this breakdown relates to the total time of the method (including called methods)

what we measure exactly (meaning: by instrumentation and callbacks) is:

* total exec time of methods (= 100% of the method breakdown)
* total CPU time of methods
* suspension time inside methods (more exactly: over all currently executed methods)

by using the information from the called methods, we get for each method in the PurePath:

* exec time (only this method)
* CPU time (only this method)
* total suspension time (this method and all called methods)

additionally, by using auto sensors and thread states we also get (additional to timings for non-instrumented methods)

* wait time of methods
* sync time of methods

IO and socketread() : This simply means that your PurePath is reading data from the network - and therefore it is considered IO. If socketread is a call you see "underneath" your JDBC calls you know it is related to retrieving data from the database server. If you see lots of socketread you may want to check whether the application is requesting a lot of data that takes a long time to transfer. Or - maybe your network bandwidth is limited and thats why socketread takes that long. It could however also be that your database server is overloaded and therefore socketread takes that long.

**Monitoring:**

**🡪** Once you converted your script you can kick off the test and which transactions is taking longer time which is reported by analyser copy that transaction

For example LRP Application 🡺 session overview🡺 you will see architecture.

in test execution we should monitor 1st hardware resource utilization of web, app, D.B

Then we have to monitor App level and Db. level

Using Dynatrace we can monitor hardware level and application level and db level

1. **Hard ware level monitoring**

Once we selected application 🡺 session🡺process level🡺thread dump, CPU sampling we can take, memory sampling, transaction flows, health of each &every process this for one app server.

1. **Application level:**

🡪goto dynatrace client and verify process level and find if any infrastructure level issues if there is any no infrastructure level issues

🡪then business flow dash let and where you can check the same transaction.

🡪Then Right click on the transaction flow diagram where you can check which layer is taking longer time🡪R.C on that🡪drilldown🡪purepath.

🡪keep the transaction in ascending or descending order you can take the highest response times.

🡪R.C on that and go to pure path methods.

🡪where you can check the which method are executing and which is taking longer time and which EJB are executing and which is taking longer time and which servlets are executing and which is taking longer time and which packages are executing and all the internal information will be available along with which operations is taking longer time.

🡪R.C and copy the details and thread details and paste it to developer.

For suppose Launch transaction taking time now we have to find out the issue.

* Go to
* Dash board 🡺 add dash let🡺business transaction (here it will show all your transactions)
* From this we can identify which transaction is taking longer time

Launch\_001 🡺 right click 🡺 go to transaction flow diagram 🡺drill drown transaction

* Now we can see 3 layers

Web app D.B

0.2 1.2 0.3

* For this we can identify in which layer this transaction taking time
* We found an issue in app layer
* From this app layer we have find out which code is causing high response time

Applier🡺right click🡺drill drown🡺 purepath

\*\*\* pure path means round trip of the request.

* + - Which instance is taking longer we don’t know just click on the response time.
    - Right click 🡺 drill down 🡺 pure path methods, methods
    - We can see how many methods are executing
    - Which api is it executed
    - Average response times
    - Copy the details of that method
    - Post to the developer
    - So, we can identify method level api level, EJV level, I/O level, servlet level.

🡪such a way that we will fix the issues ,so this is the process we are following usually.

🡪if we want we can go for particular method hotspots and DB hotspots by right click on that particular transaction.

**Data base level:**

🡺 right click🡺 drill down🡺 database exception, or data base hotspots

Which query is executed

**For knowing errors:**

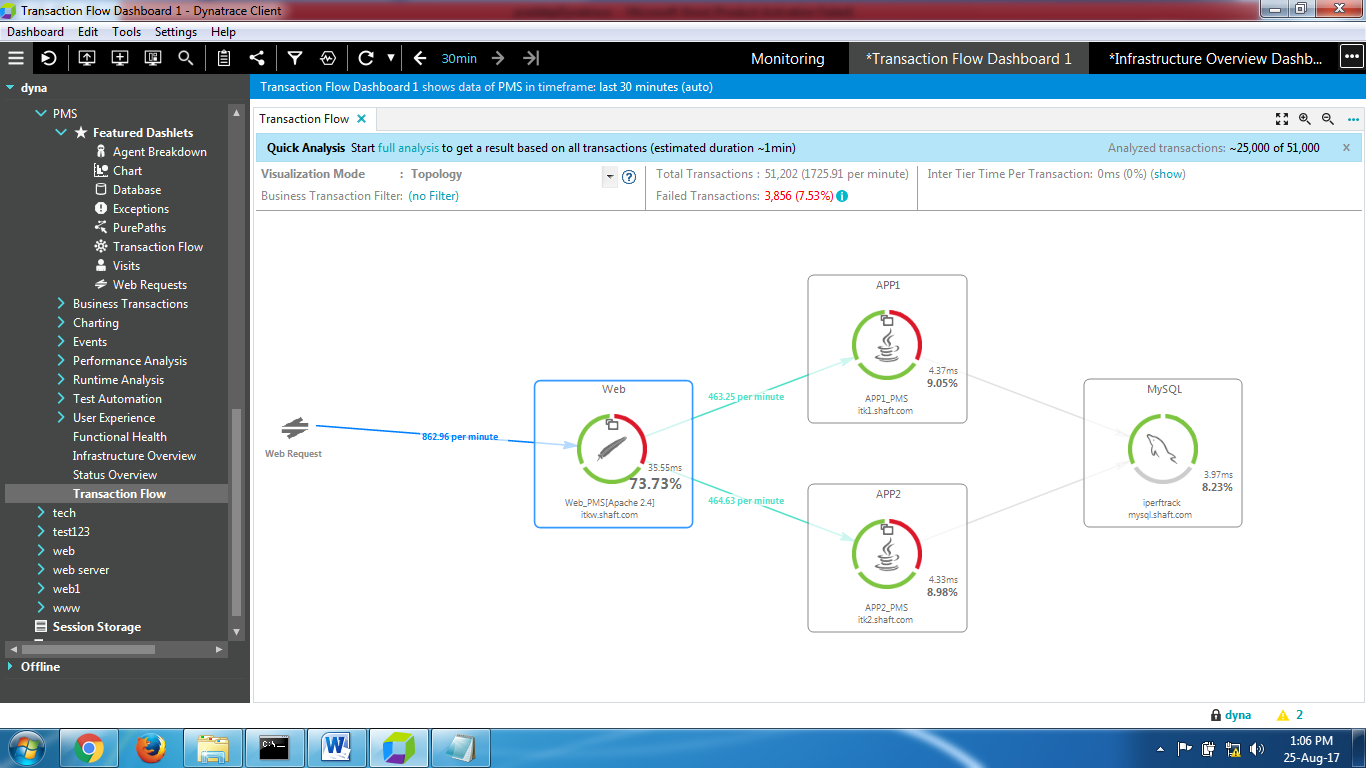
Add dash let🡺error & exception through adding this we can check any errors

Errors🡺right click 🡺 drill down🡺it will show transaction method errors.

🡪To take the thread dump in dynatrace go to session overview🡪process bar🡪process dialog.

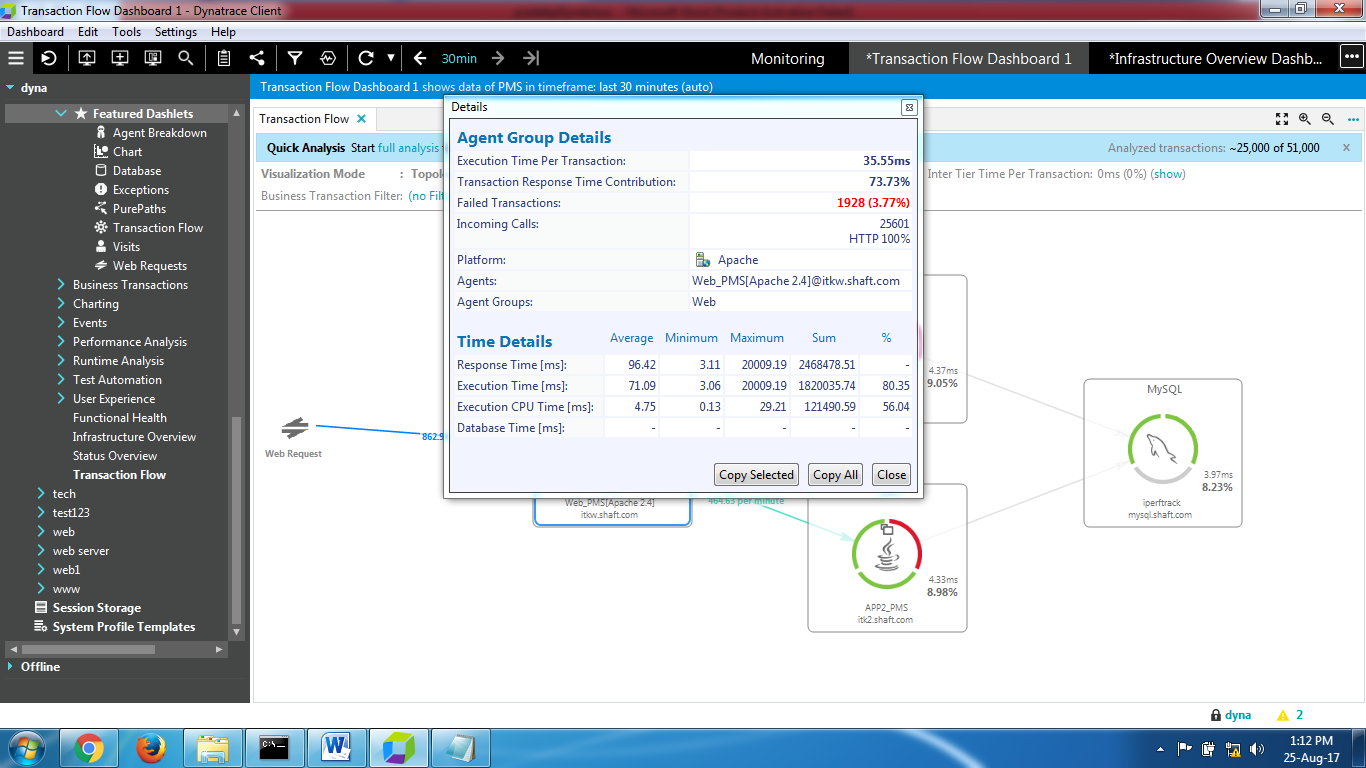
**Some screenshots guide you:**

1)Go to transaction flow .double click and view the transaction response time contribution:%

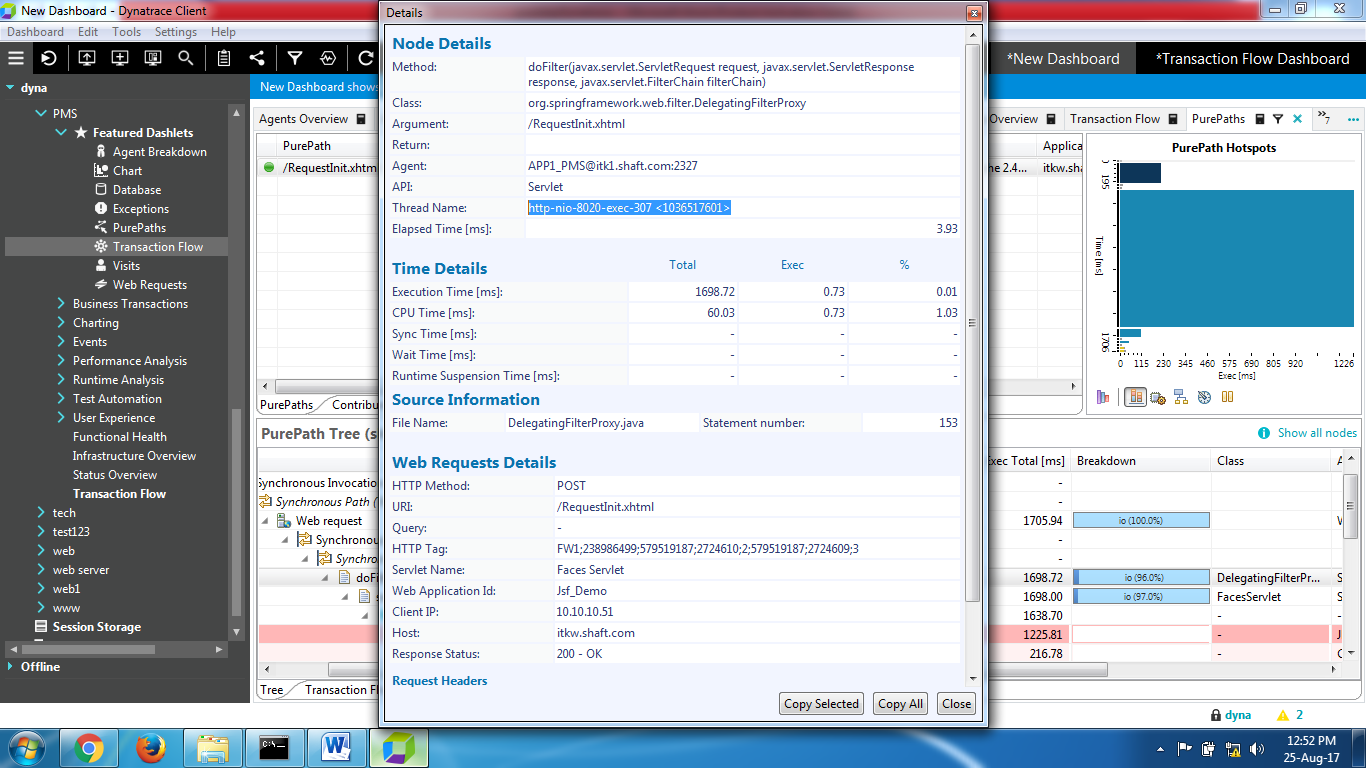
****

2)Whichever node has the highest response time. Right click on that and drill down and chose pure path

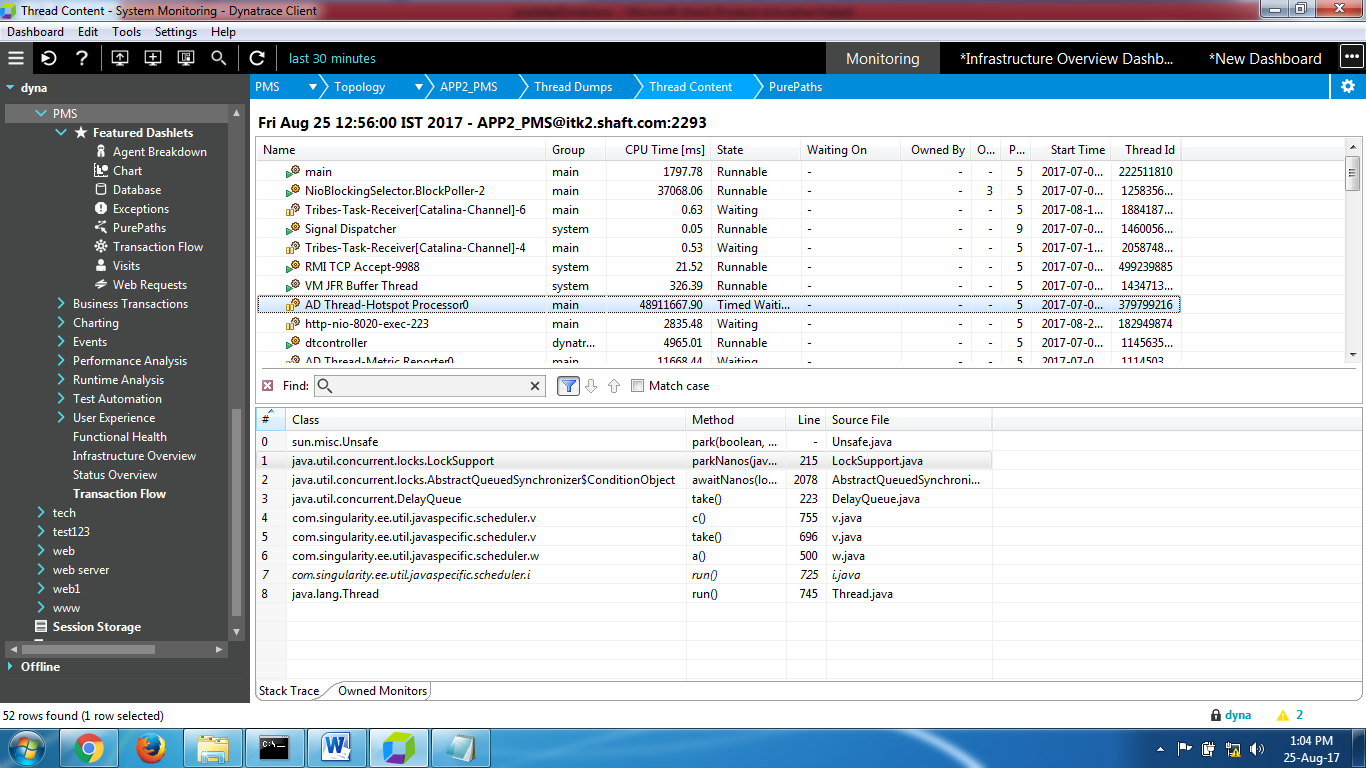
Go to transaction flow double click to see the following details like exec time,response time etc for each node



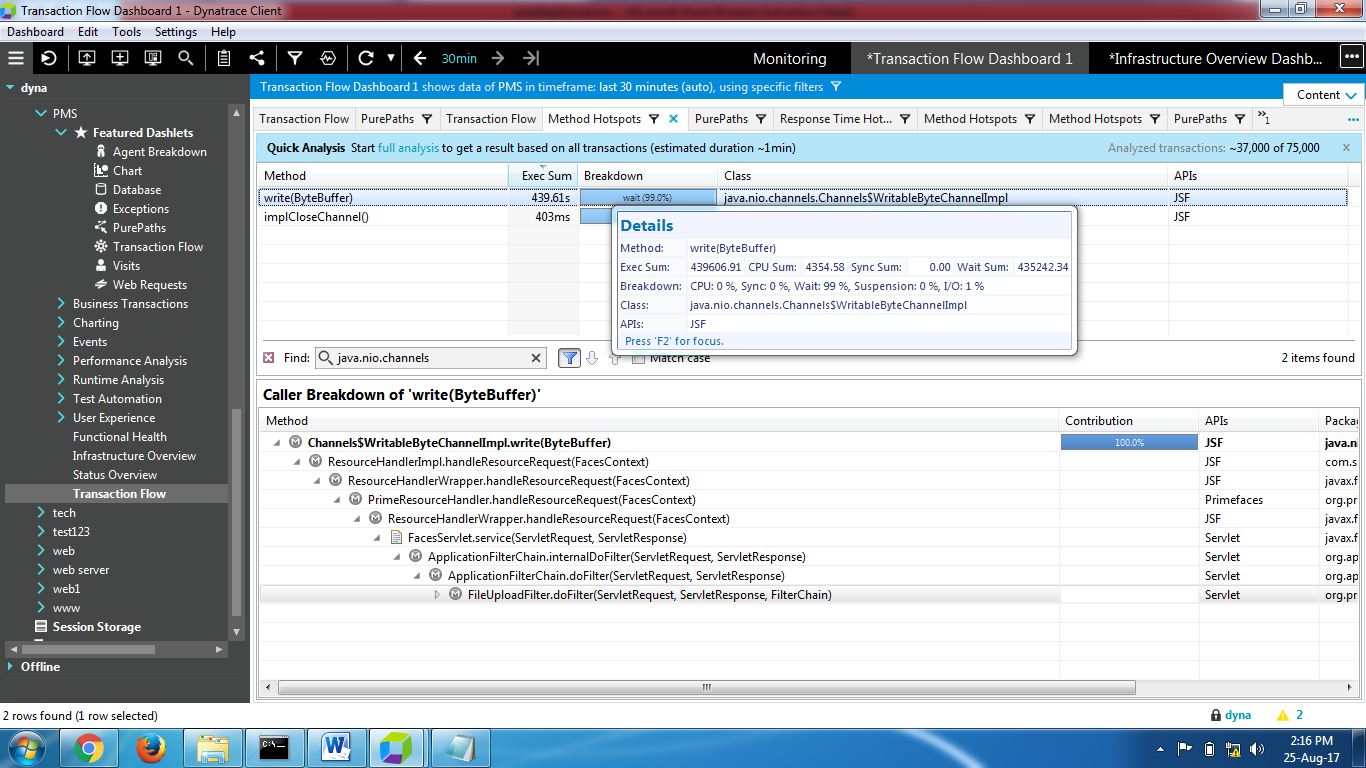
3) Elapsed time for a method,total time and execution time.



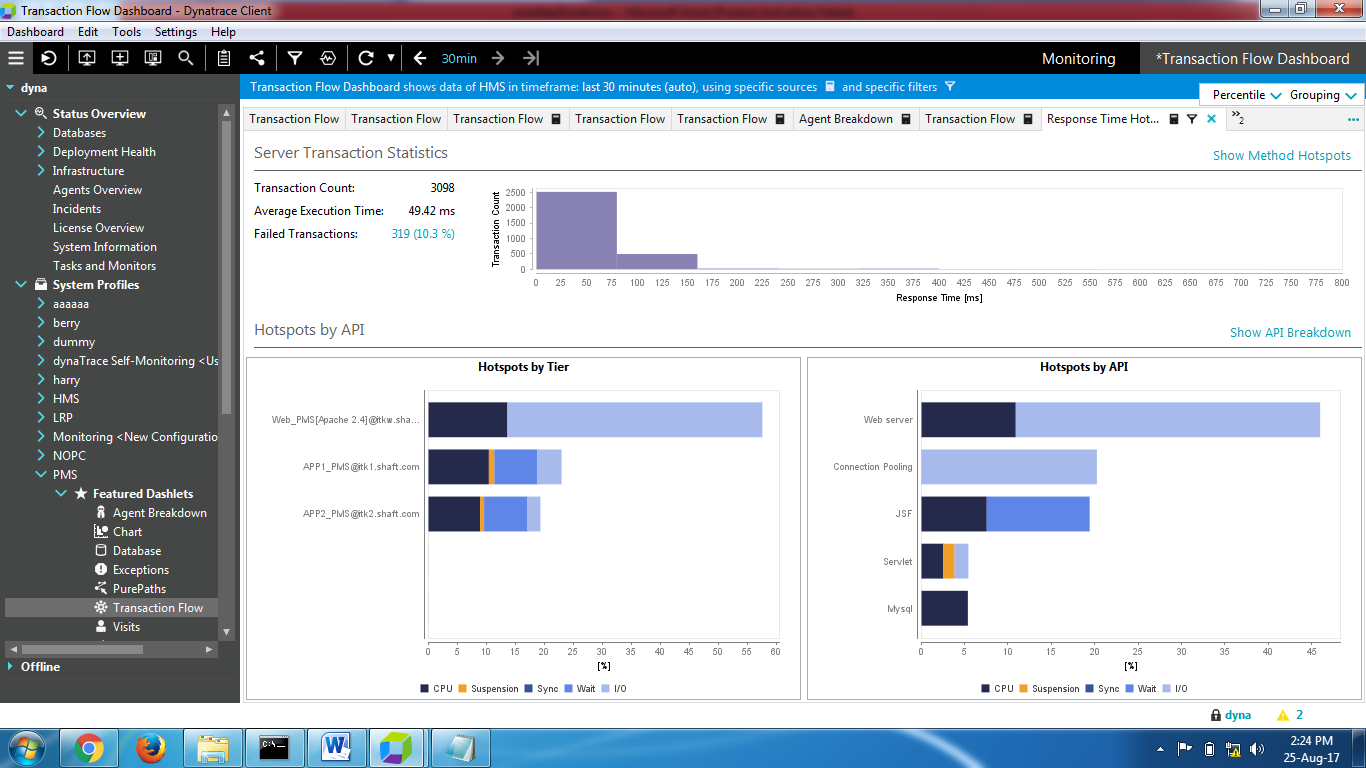
4) To find thread details.Right click on the profile and chose open session overview.Right click on the server name(ex:app\_1\_pms) and chose open process.Select thread dump (bottom right hand corner)and create a thread dump.



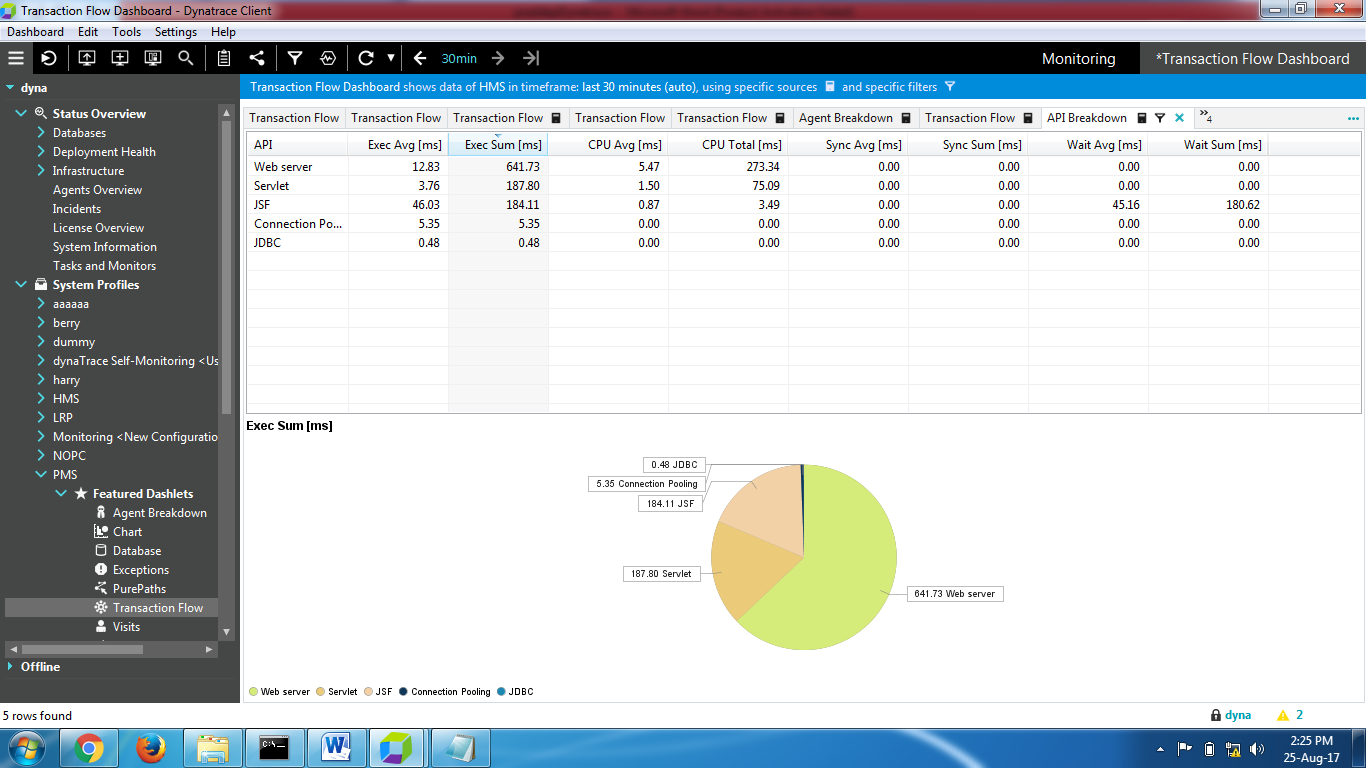
5) To know hotspots at a node level under transaction flow.Right click on the node->drill down.For example chose method hotspots



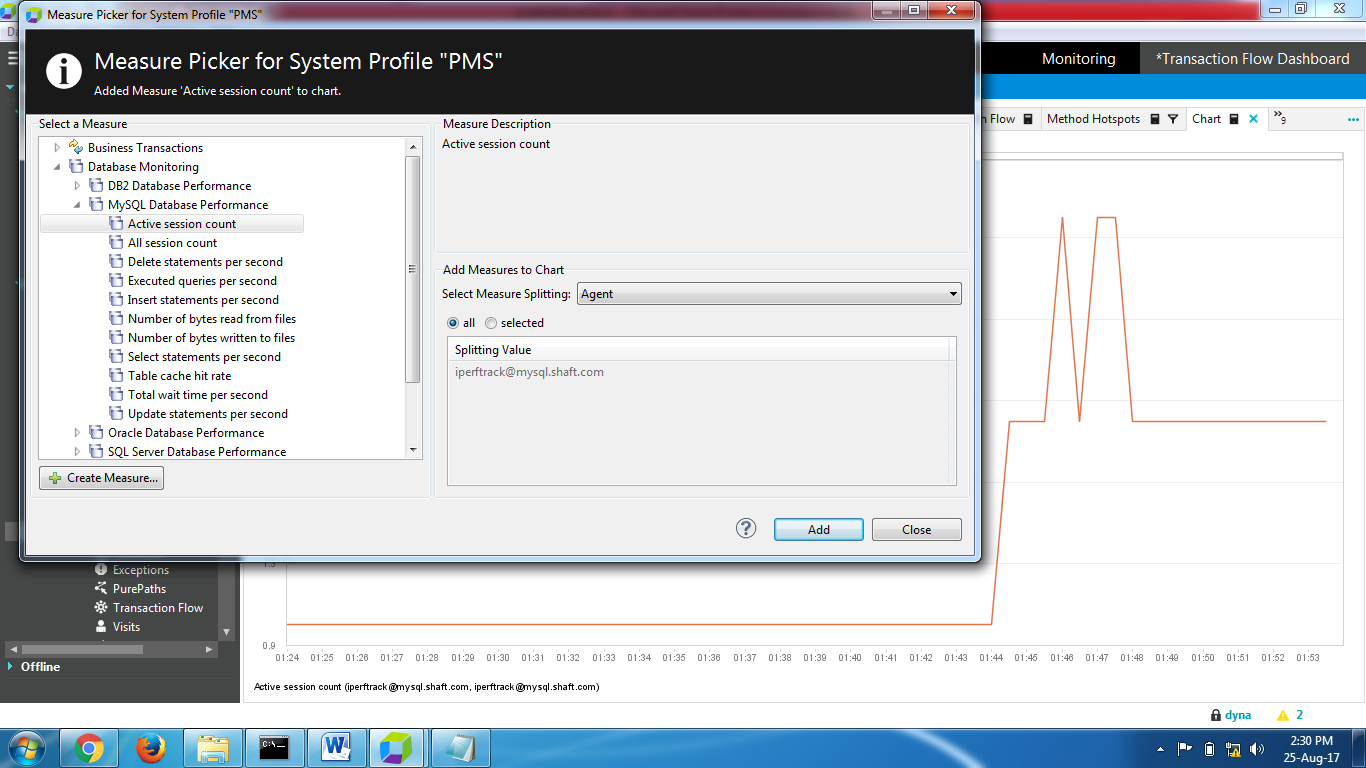
6) Response time hotspots we get after right click and drill down on the node.



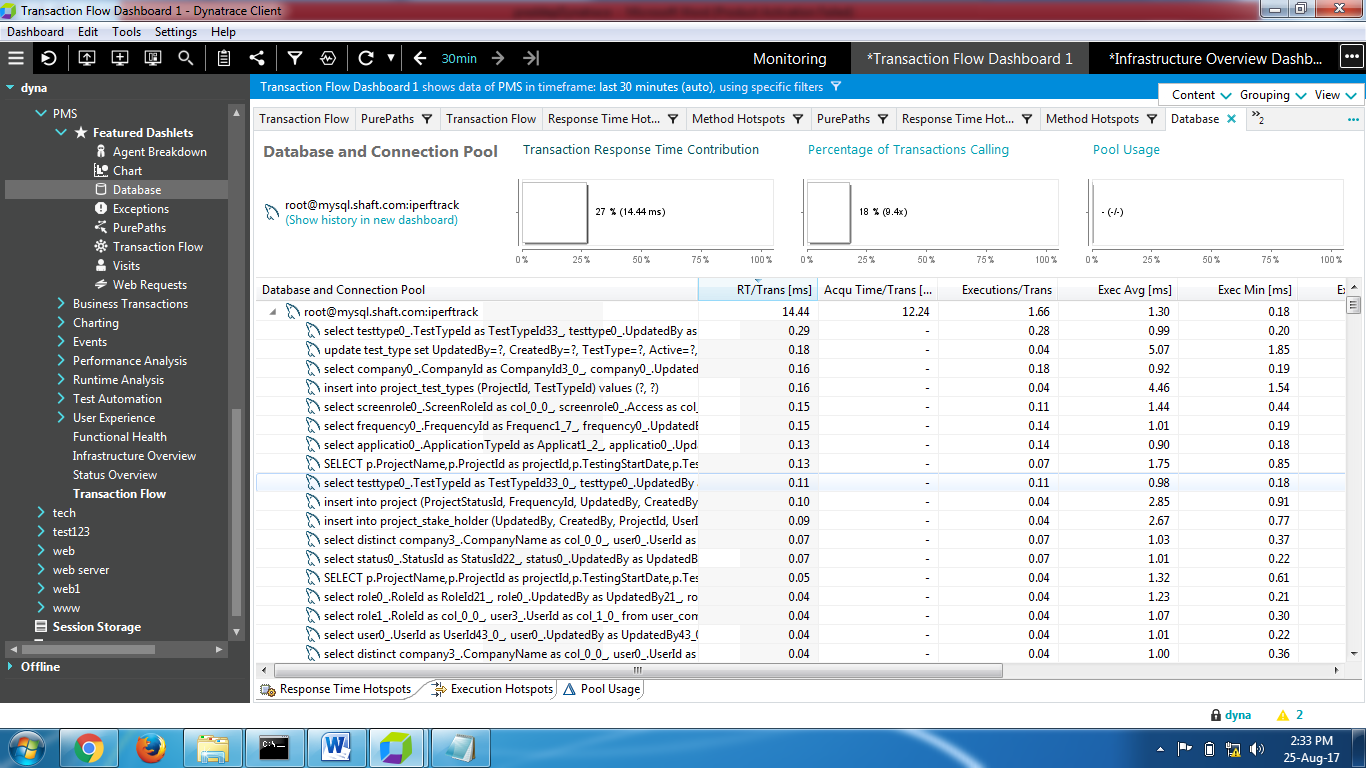
7)Right click on node and chose agent breakdown



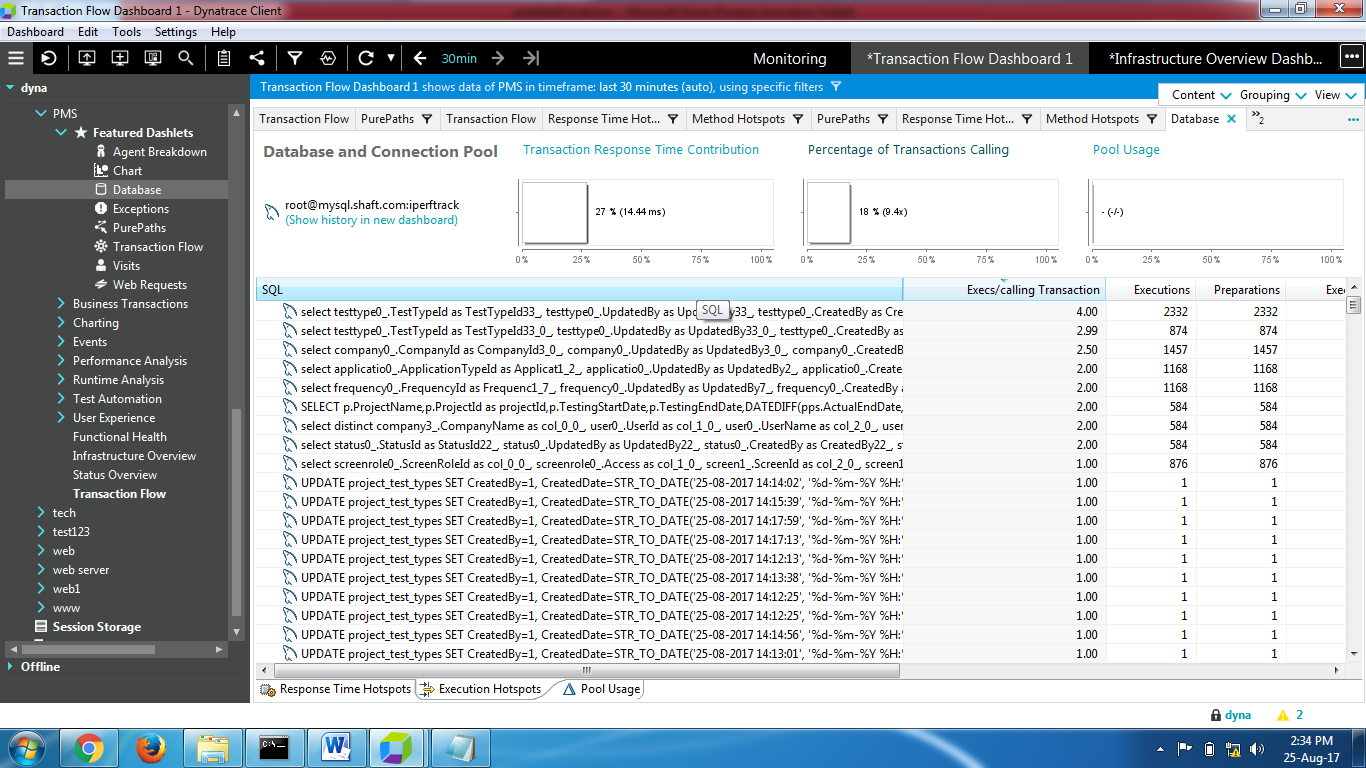
8)Click on charts under featured dashlets and add add series then add active session count and view the same



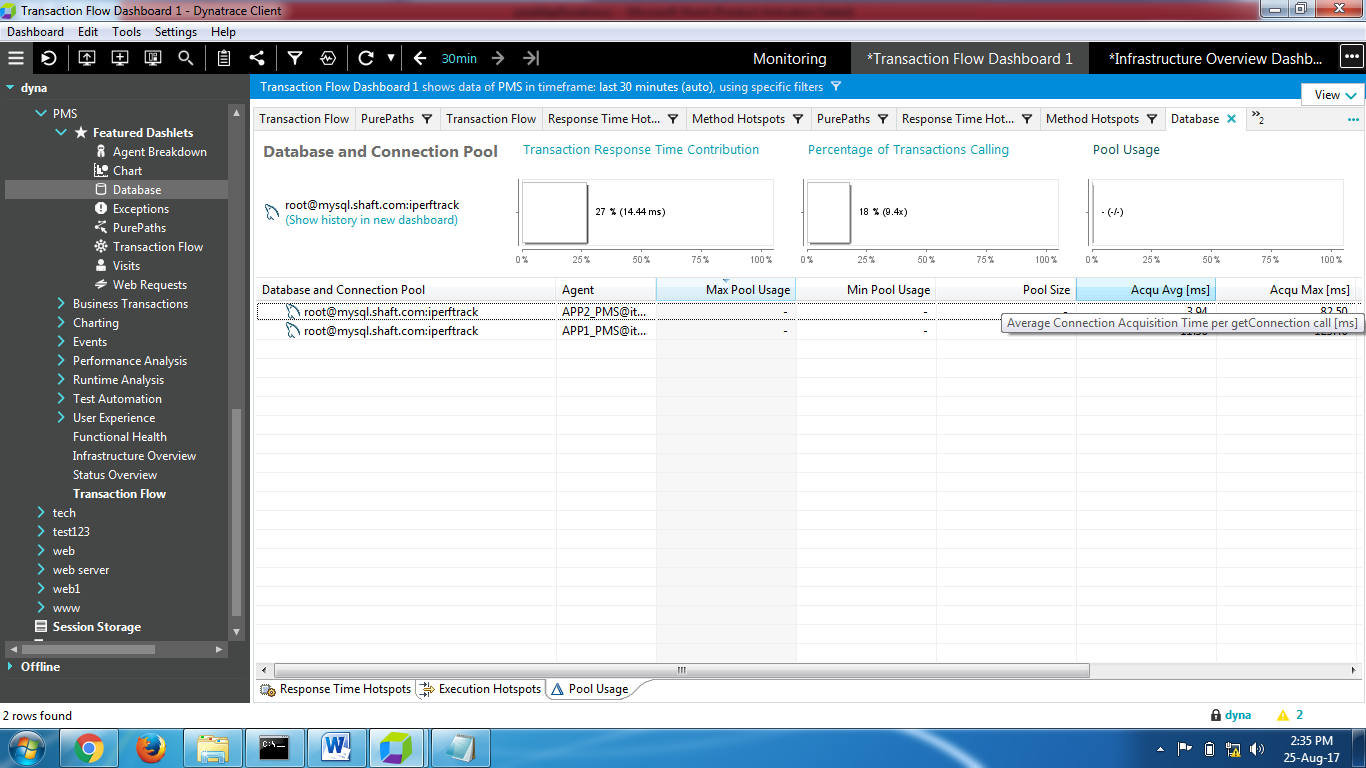
9).click on database



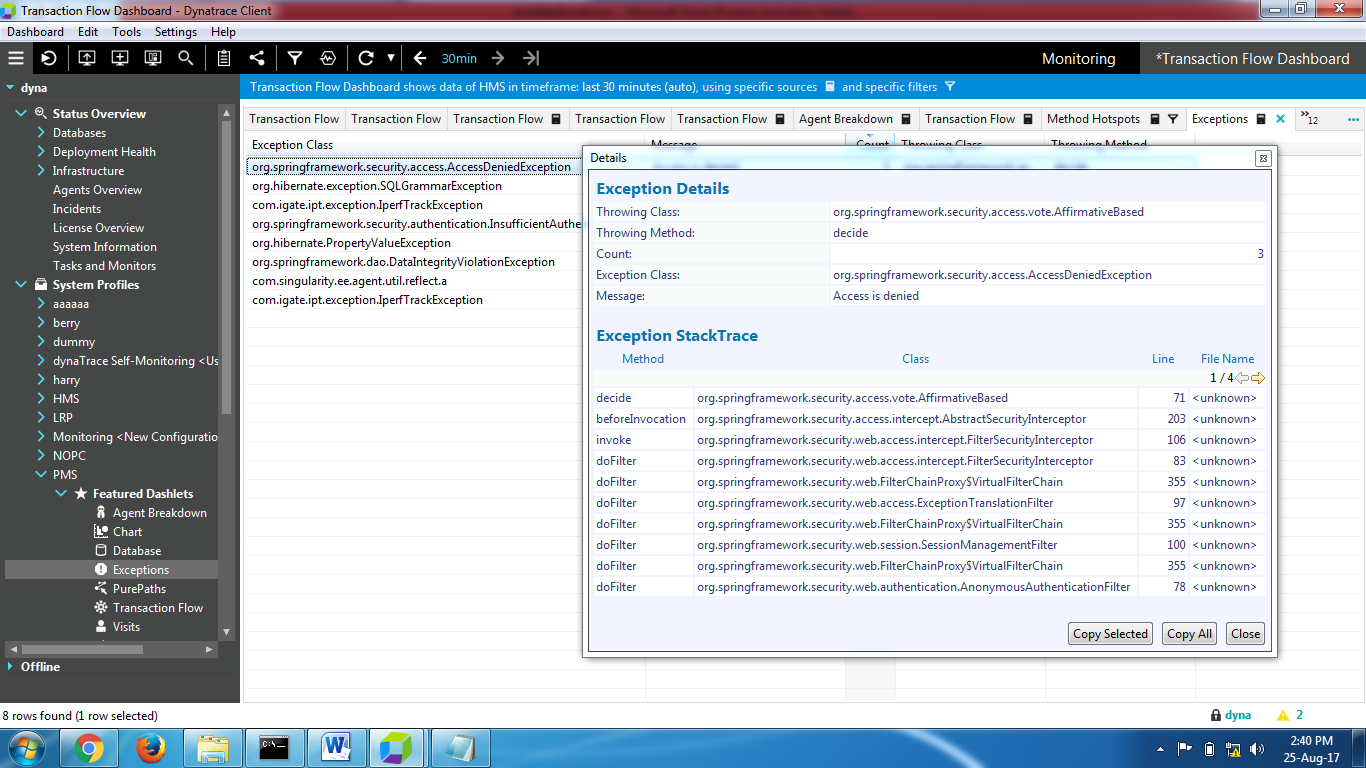
10)Percentage of transaction calling



11) Pool usage:



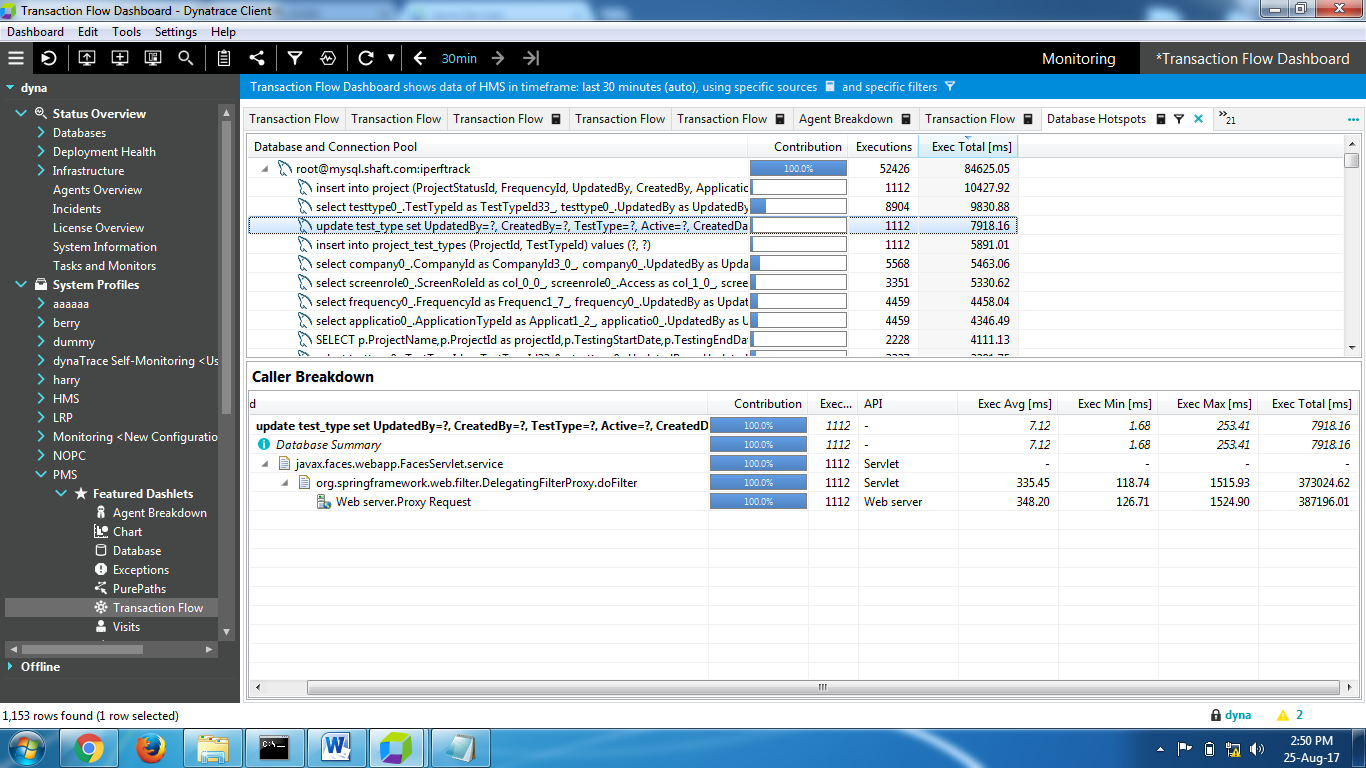
11)Exception classes under dashlet properties



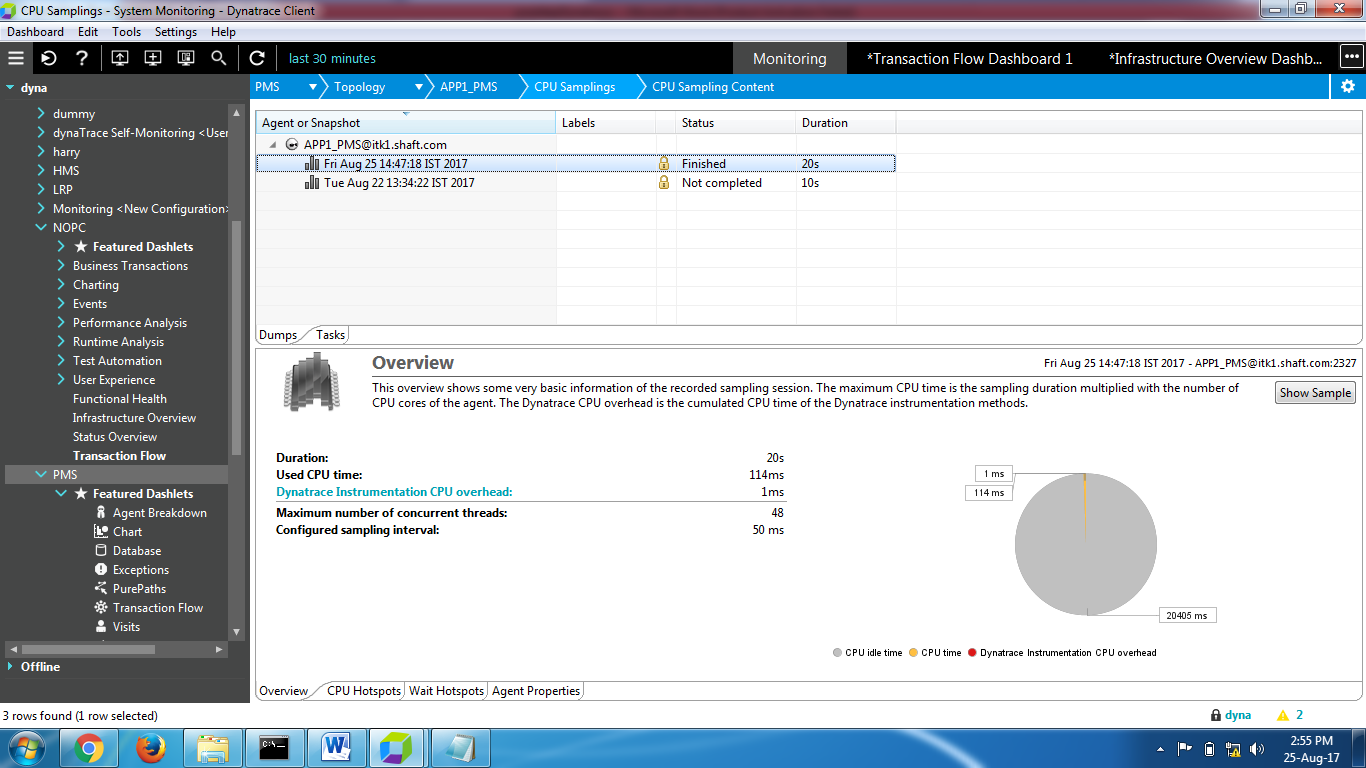
12.Web requests



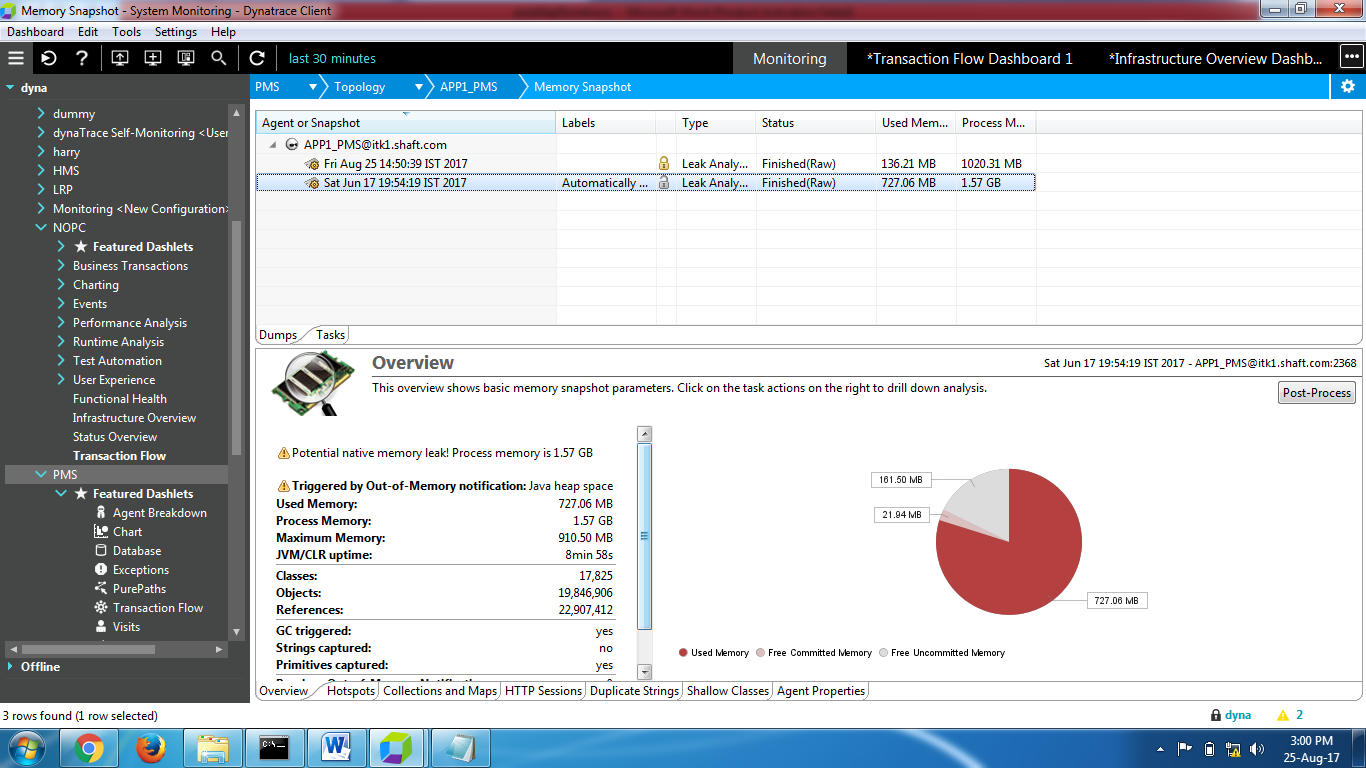
13)Right click on data node and chose database hotspots



14)CPU samplings



15)Memory snapshot leaked



16)Post processed:This can be done for all the tabs as mentioned below: Create snapshot ->Memory consumption trending

