MAGNASISTEMAS

WorkShop

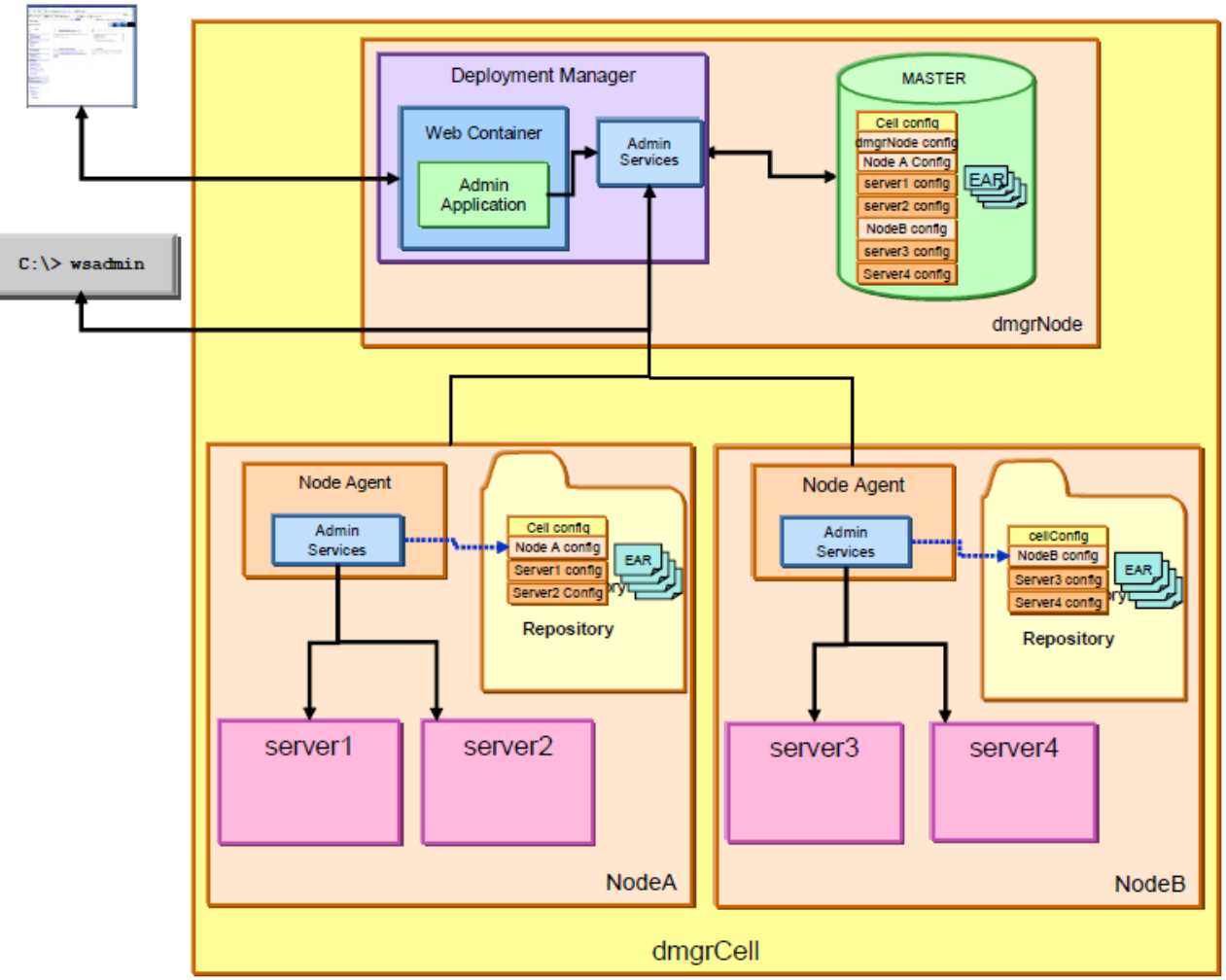
Diagnóstico de Problemas WebSphere Application Server

Tópicos do workshop

MAGNASISTEMAS

- Estrutura básica WebSphere
- Principais comandos administrativos
- Logs(Sysout, Syserr, native_stderr, native_stdout, ffdc,traces)
- Configurando traces
- PMI(Performance Monitoring Infrastructure)
- Apresentação de ferramentas IBM
- Gerando Heapdump e javacore
- Analise de JVM
- Análise de Garbage Collection
- Análise de Thread Dump
- Análise Runtime - IBM WebSphere Performance Tuning Toolkit
- Melhores práticas

Estrutura básica WebSphere



PRINCIPAIS COMANDOS

Inicialização:
startManager.sh/bat
startNode.sh/bat
startServer.sh/bat

Parada:
stopManager.sh/bat
stopNode.sh/bat
stopServer.sh/bat

Outros:
syncNode.sh/bat
serverstatus.sh/bat

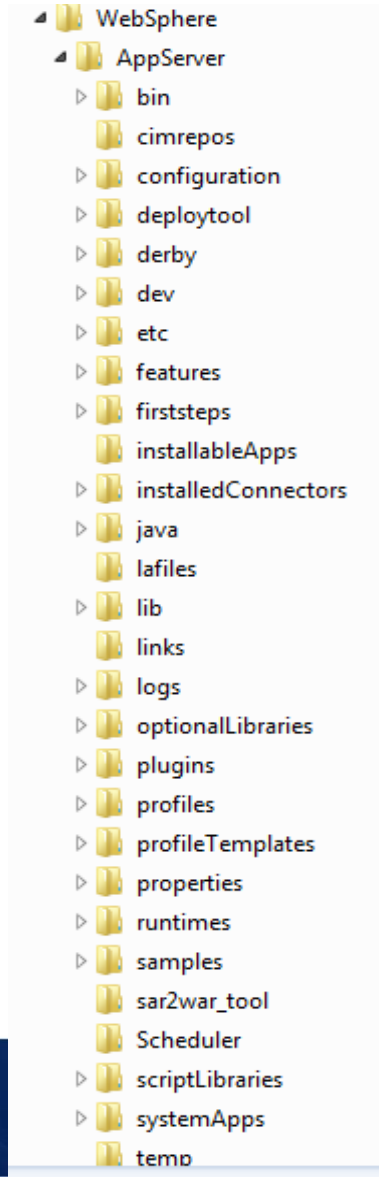
Estrutura básica WebSphere

Diretório de Instalação do Produto

ex: /opt/IBM/WebSphere/AppServer

Neste diretório temos:

- Binarios de administração
- Aplicativos do próprio sistema
- Libs do produto
- Desinstalador do produto
- Etc



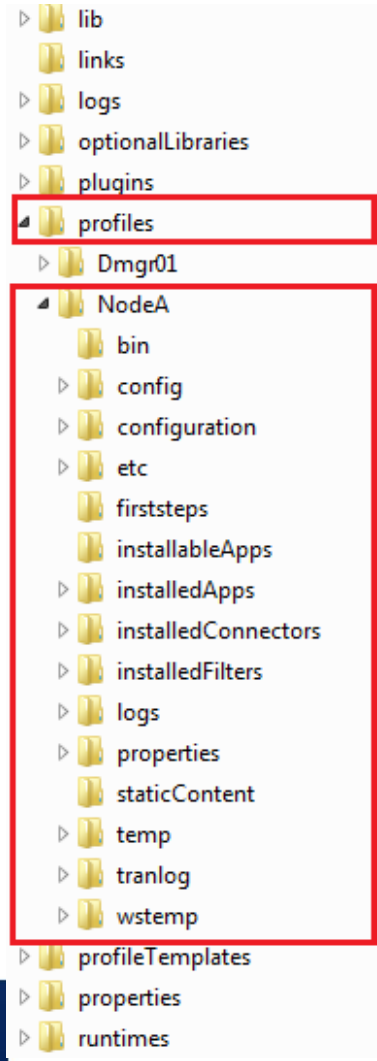
Estrutura básica WebSphere

Diretório de Profiles

ex: /opt/IBM/WebSphere/AppServer/profiles/NodeA

Neste diretório temos:

- Binarios de administração
- Arquivos de configuração do Node
- Caminho Default das aplicações instaladas
- Logs
- Temporários
- Etc



Principais comandos administrativos

Parando um ambiente WAS

Passos para parada:

1 – stopServer.sh server1 – Parando o Application Server “server1”

2 – stopNode.sh/bat – Gerenciador do Nó(Máquina)

3 – stopManager.sh/bat – Parando o console Administrativo

OBSERVAÇÃO:
Caso a segurança do WebSphere esteja habilitada, na parada dos serviços será necessário passar os seguintes parâmetros adicionais:

Ex: stopServer.sh server1 –username wasadmin –password senhadowasadmin

Principais comandos administrativos

Inicializando um ambiente WAS

MAGNASISTEMAS

Passos para inicialização:

1 – startManager.sh/bat – Console Administrativo(http://dmgrhost:9060/admin)

2 – startNode.sh/bat – Gerenciador do Nó(Máquina)

3 – startServer.sh server1 – Iniciando o Application Server “server1”

Principais comandos administrativos

Sincronizando o Node com o Dmgr

-Alguma configuração que fizemos via console administrativa, não funcionar ou não refletir conforme planejado.

- O Node perder o sincronismo integral com o Dmgr 

Comando: syncNode.sh DmgrHost PORTASOAP

EX: Acessar o diretório dos binários do profile:
/opt/IBM/WebSphere/AppServer/profiles/NodeA/bin

Rodar o seguinte comando(linux)
./syncNode.sh ServidordoDMGR 8879

Principais comandos administrativos

ServerStatus - Status dos servidores WebSphere

Checar o status dos application servers sem precisar logar no console adminsitrativo:

1- Acessar o diretorio de binários do profile:
/opt/IBM/WebSphere/AppServer/profiles/NodeA/bin

2- Executar o seguinte comando(linux):

./serverstatus.sh -all

```
D:\IBMND\WebSphere\AppServer\profiles\AppSrv01\bin>serverstatus -all
ADMU0116I: As informações da ferramenta estão sendo registradas no arquivo
           D:\IBMND\WebSphere\AppServer\profiles\AppSrv01\logs\serverStatus.log
ADMU0128I: Iniciando ferramenta com o perfil AppSrv01
ADMU0503I: Recuperando status do servidor para todos os servidores
ADMU0505I: Servidores localizados na configuração:
ADMU0506I: Nome do servidor: Member1
ADMU0506I: Nome do servidor: Member2
ADMU0506I: Nome do servidor: nodeagent
ADMU0506I: Nome do servidor: Server1
ADMU0506I: Nome do servidor: Serverws01
ADMU0506I: Nome do servidor: Serverws02
ADMU0506I: Nome do servidor: webserver1
ADMU0509I: O Application Server "Member1" não pode ser alcançado. Ele parece
           estar parado.
ADMU0509I: O Application Server "Member2" não pode ser alcançado. Ele parece
           estar parado.
ADMU0508I: O Node Agent "nodeagent" está STARTED
ADMU0509I: O Application Server "Server1" não pode ser alcançado. Ele parece
           estar parado.
ADMU0509I: O Application Server "Serverws01" não pode ser alcançado. Ele parece
           estar parado.
ADMU0509I: O Application Server "Serverws02" não pode ser alcançado. Ele parece
           estar parado.
ADMU0508I: O Web server "webserver1" está RUNNING
```

Diagnóstico de Problemas

Logs do WAS – parte 1

Principais Logs do WebSphere

SystemOut.log e SystemErr.log – Padrão de saída da JVM e de erros

Caminho: /opt/IBM/WebSphere/AppServer/profiles/NodeA/logs/ServerA/

startServer.log e stopServer.log – Logs de start e stop dos Application Servers

Caminho: /opt/IBM/WebSphere/AppServer/profiles/NodeA/logs/ServerA/

activity.log – Eventos que monstram o histórico de atividades

Caminho: /opt/IBM/WebSphere/AppServer/profiles/NodeA/logs/

Observação: Parar conseguir ler de forma fácil rodar com o comando ./showlog $Path/activity.log

Diagnóstico de Problemas

Logs do WAS – parte 2

MAGNASISTEMAS

Principais Logs do WebSphere

trace.log – Saída dos logs de trace(habilitado no Dmgr)

Caminho: /opt/IBM/WebSphere/AppServer/profiles/NodeA/logs/ServerA/

FFDC logs – First Failure Data Capture – Preserva informações geradas devido a um processo de falha

Caminho: /opt/IBM/WebSphere/AppServer/profiles/NodeA/logs/ffdc

nativestderr.log e nativestdout.log – Logs de processos. Com esses logs podemos analisar o comportamento do Garbage Collection, seja abrindo diretamente o arquivo ou analisando através de ferramentas IBM. **Habilitar o Verbose GC.**

Caminho: /opt/IBM/WebSphere/AppServer/profiles/NodeA/logs/ServerA/

Diagnóstico de Problemas

Logs do WAS – SysOut e SysErr

Informações importantes

- No início do troubleshooting são uns dos primeiros arquivos a serem analisados

- Quando identificado um Stack Trace, a forma correta para sua leitura e identificação do problema é análisa-lo de baixo para cima

- Problemas que visualizamos neste arquivo: comunicação com banco de dados, serviços externos, threads hung, out of memory, erros de aplicação(qdo o sysout esta como saida padrão da app) entre outros

Diagnóstico de Problemas

Logs do WAS – Stack Trace

MAGNASISTEMAS

```
CNTR0020E: EJB threw an unexpected (non-declared) exception during invocation of method "gerenciarArquivoDePagamento" on bean "BeanId(Vre-Homologa-App#PEQEJB.jar#GerenciaContratoPEQService, null)". Exception data: java.lang.NumberFormatException: For input string: "    "
    at java.lang.Throwable.<init>(Throwable.java:67)
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:60)
    at java.lang.Integer.parseInt(Integer.java:461)
    at java.lang.Integer.parseInt(Integer.java:511)
    at br.gov.sp.prodesp.peq.contrato.business.ContratoPEQBE.gerenciarArquivoDePagamento(ContratoPEQBE.java:4926)
    at br.gov.sp.prodesp.peq.contrato.business.service.GerenciaContratoPEQServiceImpl.gerenciarArquivoDePagamento(GerenciaContratoPEQServiceImpl.java:3835)
    at br.gov.sp.prodesp.peq.contrato.business.service.EJSRemoteStatelessGerenciaContratoPEQService_8eac717a.gerenciarArquivoDePagamento(EJSRemoteStatelessGerenciaContratoPEQService_8eac717a.java:298)
    at br.gov.sp.prodesp.peq.contrato.business.service._GerenciaContratoPEQService_Stub.gerenciarArquivoDePagamento(_GerenciaContratoPEQService_Stub.java:1576)
    at br.gov.sp.prodesp.peq.ztask.TasksDelegate.gerenciarArquivoDePagamento(TasksDelegate.java:427)
    at br.gov.sp.prodesp.peq.ztask.ImportarArquivoDePagamento.getTratarArquivo(ImportarArquivoDePagamento.java:263)
    at br.gov.sp.prodesp.peq.ztask.ImportarArquivoDePagamento.importarArquivo(ImportarArquivoDePagamento.java:102)
    at br.gov.sp.prodesp.peq.ztask.ImportarArquivoDePagamento.execute(ImportarArquivoDePagamento.java:56)
    at br.gov.sp.prodesp.peq.servlet.scheduler.SchedulerExecutor.service(SchedulerExecutor.java:41)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:831)
    at com.ibm.ws.webcontainer.servlet.ServletWrapper.service(ServletWrapper.java:1655)
    at com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrapper.java:937)
    at com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrapper.java:500)
```

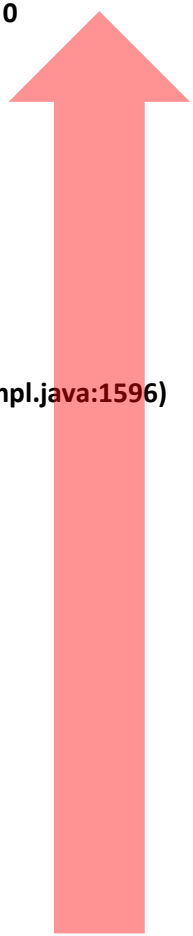
SystemOut.log

Diagnóstico de Problemas

Logs do WAS – Stack Trace

Caused by: com.ibm.websphere.ce.cm.StaleConnectionException: Falha na conexão TCP/IP com o host 10.200.45.242, porta 1433. Erro: "connect timed out. Verifique as propriedades da conexão, se uma instância do SQL Server está sendo executada no host e se está aceitando conexões TCP/IP na porta, e verifique se nenhum firewall está bloqueando as conexões TCP na porta.".DSRA0010E: SQL State = 08S01, Error Code = 0

at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:44)
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:39)
at java.lang.reflect.Constructor.newInstance(Constructor.java:504)
at com.ibm.websphere.rsadapter.GenericDataStoreHelper.mapExceptionHelper(GenericDataStoreHelper.java:605)
at com.ibm.websphere.rsadapter.GenericDataStoreHelper.mapException(GenericDataStoreHelper.java:667)
at com.ibm.ws.rsadapter.AdapterUtil.mapException(AdapterUtil.java:2111)
at com.ibm.ws.rsadapter.spi.WSRdbDataSource.getPooledConnection(WSRdbDataSource.java:2289)
at com.ibm.ws.rsadapter.spi.WSManagedConnectionFactoryImpl.createManagedConnection(WSManagedConnectionFactoryImpl.java:1596)
at com.ibm.ejs.j2c.FreePool.createManagedConnectionWithMCWrapper(FreePool.java:2032)
at com.ibm.ejs.j2c.FreePool.createOrWaitForConnection(FreePool.java:1709)
at com.ibm.ejs.j2c.PoolManager.reserve(PoolManager.java:2470)
at com.ibm.ejs.j2c.ConnectionManager.allocateMCWrapper(ConnectionManager.java:1059)
at com.ibm.ejs.j2c.ConnectionManager.allocateConnection(ConnectionManager.java:696)
at com.ibm.ws.rsadapter.jdbc.WSJdbcDataSource.getConnection(WSJdbcDataSource.java:668)
at com.ibm.ws.rsadapter.jdbc.WSJdbcDataSource.getConnection(WSJdbcDataSource.java:793)
at net.sf.hibernate.websphere.WSDataSourceConnectionProvider.getConnection(WSDataSourceConnectionProvider.java:69)
at org.hibernate.jdbc.ConnectionManager.openConnection(ConnectionManager.java:446)
... 44 more



SystemErr.log

Diagnóstico de Problemas

Logs do WAS – startServer.log/stopServer.log

Se no momento de iniciar o WebSphere, ocorrer algum problema e o serviço não “subir”.

Analisar primeiramente estes logs para identificar os problemas de start e stop.

WSVR0009E: Error occured during startup. com.ibm.ws.exception.RuntimeError: Java.lang.NoClassDefFoundError: com/ibm/ws/process/Win32ProcessGlue

Diagnóstico de Problemas

Logs do WAS – activity.log/ffdc.log

Outros logs que nos ajudam a realizar um troubleshooting inicial são o ffdc.log e activity.log.

Existe uma particularidade na hora de visualizar o activity.log que é a ferramenta showlog.sh/bat

Diretório do binário:/opt/IBM/WebSphere/AppServer/bin/

./showlog.sh "caminho completo do activity.log"

./showlog.sh /opt/IBM/WebSphere/AppServer/profiles/NodeA/logs/activity.log

Diagnóstico de Problemas

Logs do WAS – nativestderr.log

Com este log podemos analisar o comportamento da JVM no que diz respeito a Garbage Collection.

Podemos visualizar abrindo diretamente o log, ou através da Ferramenta: IBM Garbage Collection and Memory Visualizer

Diagnóstico de Problemas

Configurando o TRACE

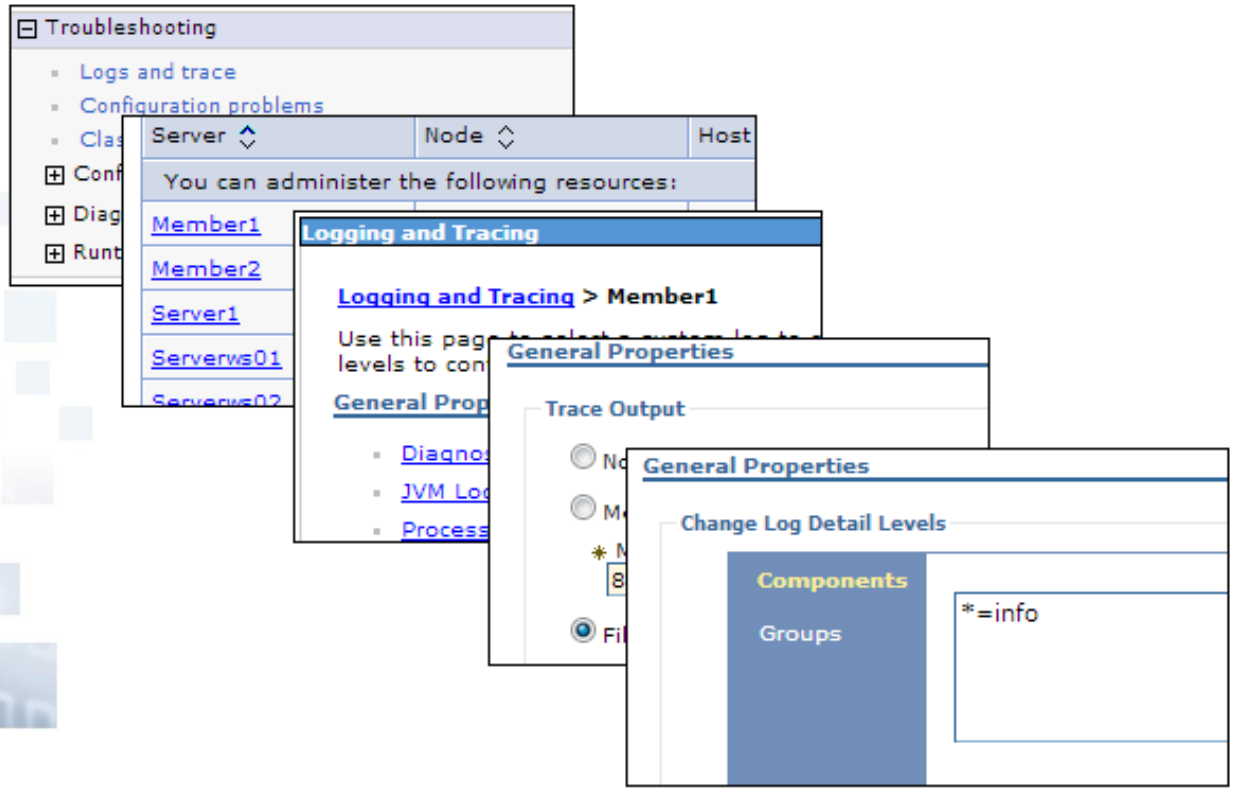
- **Os traces ajudam a diagnosticar problemas ou avaliar o desempenho do ambiente.**
- **Traces podem ser habilitados em tempo de execução ou permanentemente**
- **Componentes específicos podem ser monitorados como por exemplo:**

com.ibm.ws.oracle.logwriter - Trace de strings para banco Oracle
com.ibm.ws.sqlserver.logwriter Trace de strings para banco Microsoft SQL Server
com.ibm.ws.webcontainer – Trace detalhado do WebContainer

Diagnóstico de Problemas

Configurando o TRACE

- **Passos para configurar traces no websphere:**



Diagnóstico de Problemas

Configurando o TRACE

PERMANENTE (Necessário Restart do Application Server)

1- Selecione Troubleshooting-> Logs and Trace via console adminsitrativo(DMGR)
2- No painel Logging and Tracing , clique no application server de interesse. Um novo painel se abrirá
3- Clique em Diagnostic Trace
4- No painel Trace Output alterar o valor de none para file
5- Clicar em Apply
5- Vá no painel Additional Properties e clique em Change Log Detail Levels
6- No painel Change Log Detail Levels, alterar o valor default *=info, para o(s) componente(s) que deseja analisar

TEMPO DE EXECUÇÃO (Não existe a necessidade de restart)

1- Selecione Troubleshooting-> Logs and Trace via console adminsitrativo(DMGR)
2- No painel Logging and Tracing , clique no application server de interesse. Um novo painel se abrirá
3- Clique na aba Runtime
4- No painel Trace Output alterar o valor de none para file
5- Clicar em Apply
6- Vá no painel Additional Properties e clique em Change Log Detail Levels
7- No painel Change Log Detail Levels, alterar o valor default *=info, para o(s) componente(s) que deseja analisar

Diagnóstico de Problemas

Configurando o TRACE

DESABILITANDO TRACE

1- Selecione Troubleshooting-> Logs and Trace via console adminsitrativo(DMGR)

2- No painel Logging and Tracing , clique no application server de interesse. Um novo painel se abrirá

3- Clique em Diagnostic Trace

4- No painel Trace Output alterar o valor de file para None

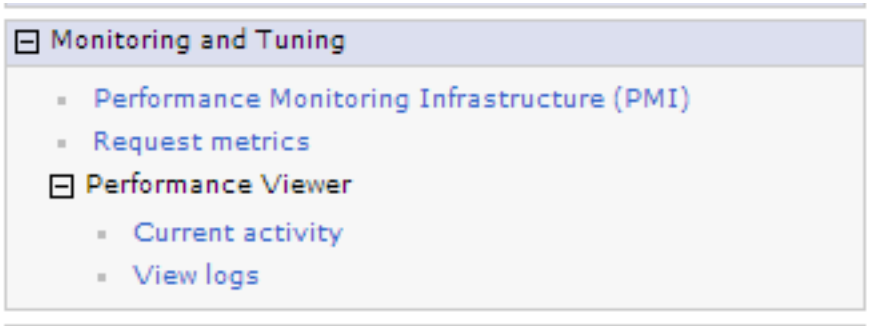
Performance Monitoring Infrastructure

O que é ?

- **O PMI(Performance Monitoring Infrastructure) tem o objetivo de coletar em tempo real estatísticas da aplicação e mostrar informações de desempenho dos Application Servers, monitorando JVM,JDBC, Servlets, Thread Pools , Advisors e etc).**

- **Integrado com o Console Administrativo(DMGR)**

- **Possui performance advisors –Recomendações automáticas , analizando os dados coletados com o PMI.**



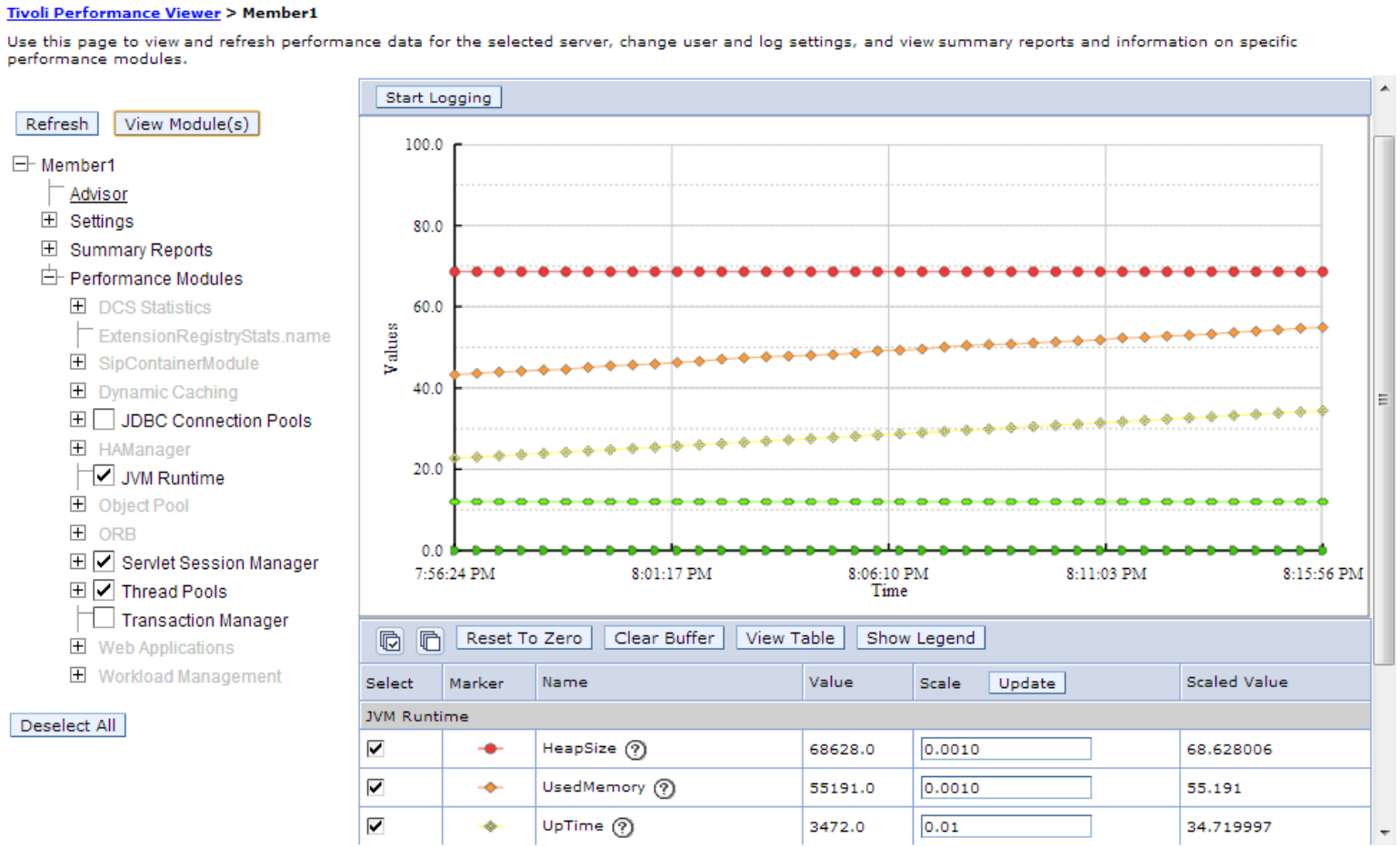
Performance Monitoring Infrastructure

Habilitando a coleta

1 - Expanda a aba Monitoring and Tuning
2 - Clique em Performance Monitoring Infrastructure(PMI)
3 - Clique no application server que deseja monitorar
4 - Selecione a caixa Enable Performance Monitoring Infrastructure (PMI)
5 - No painel Currently monitored statistic set, selecione o nível de coleta
6 - Clique em OK.
7 - Expanda novamente a aba Monitoring and Tuning
8 - Clique em Current activity
9 - Selecione o application server que deseja monitorar e clique em Start Monitoring
10 - Na mesma tela clique no application server que deseja monitorar
11 - Expanda o item do menu Performance Modules
12 - Selecione o(s) módulo(s) que deseja monitorar e clique em View Module(s)

Performance Monitoring Infrastructure

Exemplo

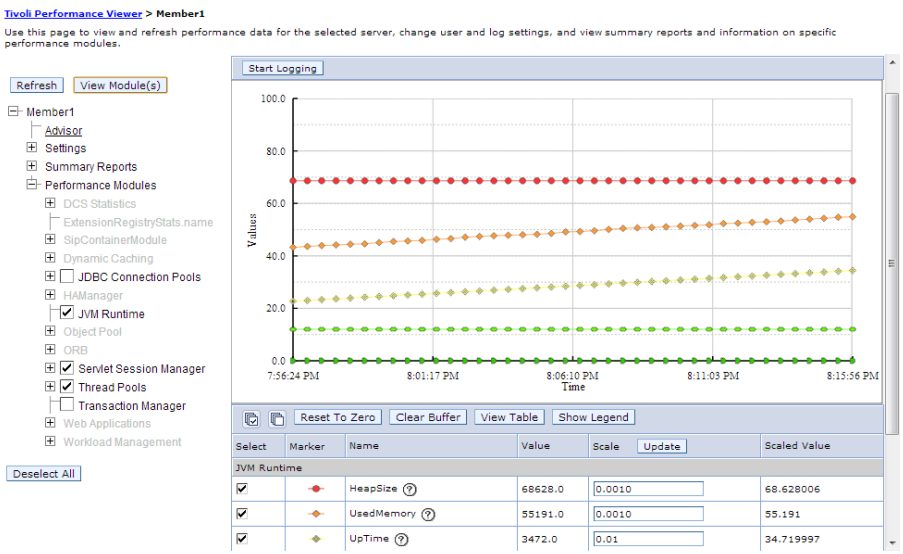


Performance Monitoring Infrastructure

Principais itens

JDBC

WaitingThreadCount - Valor deve ser igual a zero, caso seja diferente aumentar o tamanho do Pool, ou averiguar como esta a conectividade com o banco



JVM

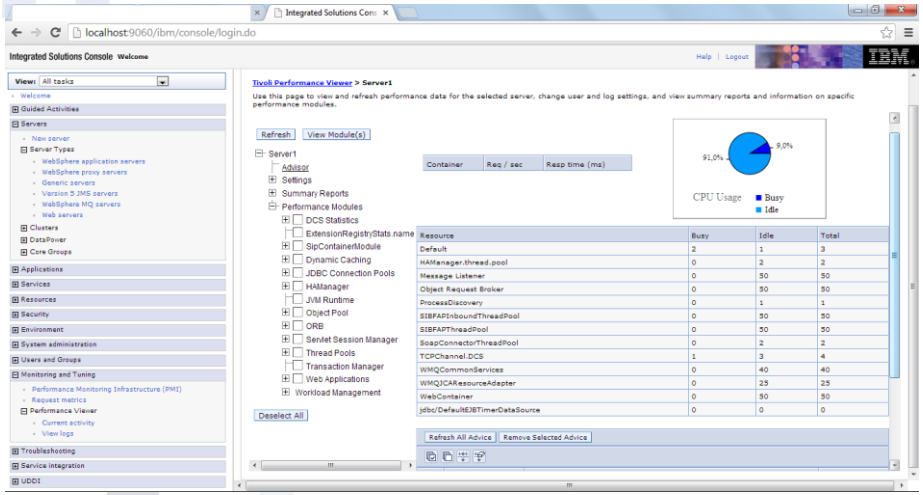
UsedMemory - Deve estar no máximo entre 40% à 70% de uso do tamanho da heap, se estiver com mais, deve ser feito um troubleshooting mais efetivo

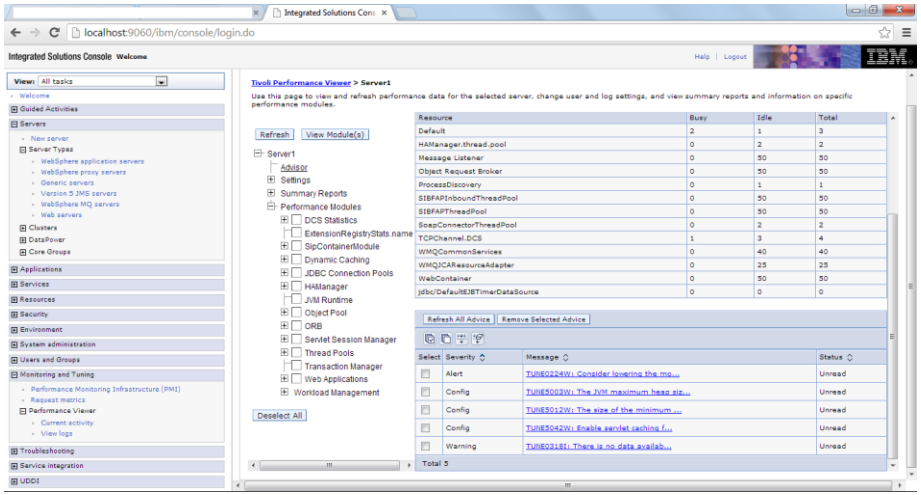
THREADS

WebContainer: Verificar se o PoolSize atinge a configuração de threads configuradas. Caso atinja, paliativamente aumente este valor Maximum Size

Performance Monitoring Infrastructure

Advisors





Visão Global

Clicando em ADVISOR, é apresentado uma tabela com os principais valores configurados , o quanto esta sendo usado, e o quando esta livre.

Advices

Abaixo seguem alguns advices que "podem" ser considerados para realização de ajustes.

Apresentação das ferramentas IBM

IBM Support Assistant(ISA)
Ferramenta responsável por coletar, analisar e enviar os logs para a IBM.

IBM Garbage Collection and Memory Visualizer
Ferramenta que consome os logs do Garbage Collection, visualiza os resultados, e fornece dados analíticos . Também gera relatórios que mostram o que deu certo,o que deu errado, o que precisa de atenção e fornece recomendações para ajudar a melhorar o desempenho

IBM Heap Dump Analyzer
Ferramenta gráfica que identifica possíveis Java heap leaks(Leak Suspect)

IBM Thread and Monitor Dump Anlyzer
A ferramenta ajuda a identificar “hangs”, deadlocks, contenção de recursos e gargalos nas threads.

Gerando Heapdump e threaddump

Método manual (via script)

Arquivos necessários para utilizar com as ferramentas IBM Heap Dump Analyzer e IBM Thread and Monitor Dump Analyzer

Para gerar manualmente estes arquivos necessitamos utilizar a ferramenta **wsadmin** para rodar scripts(linguagem jacl/jython).

Caminho desta ferramenta(linux):
/opt/IBM/WebSphere/AppServer/bin/wsadmin.sh

Gerando Heapdump e threaddump

Método manual (via script)

Gerando Heap Dump (utilizando linguagem jacl)

1- Abrir a ferramenta wsadmin com o seguinte comando:

/opt/ibm/WebSphere/AppServer/bin/wsadmin.sh

2- Executar os seguintes comandos:

<wsadmin> set jvm [$AdminControl completeObjectName type=JVM,process=NomeDoServer,*]
<wsadmin> $AdminControl invoke $jvm generateHeapDump

Gerando Heapdump e threaddump

Método manual (via script)

Gerando Thread Dump (utilizando linguagem jacl)

1- Abrir a ferramenta wsadmin com o seguinte comando:

/opt/ibm/WebSphere/AppServer/bin/wsadmin.sh

2- Executar os seguintes comandos:

<wsadmin> set jvm [$AdminControl completeObjectName type=JVM,process=NomeDoServer,*]
<wsadmin> $AdminControl invoke $jvm dumpThreads

Análise de JVM

IBM Heap Dump Analyzer

Podemos executar esta ferramenta através do IBM ISA ou diretamente executando um jar. – ha450.jar

Segue abaixo como abrir a aplicação:

1- Abrir o command navegar até o diretório acima

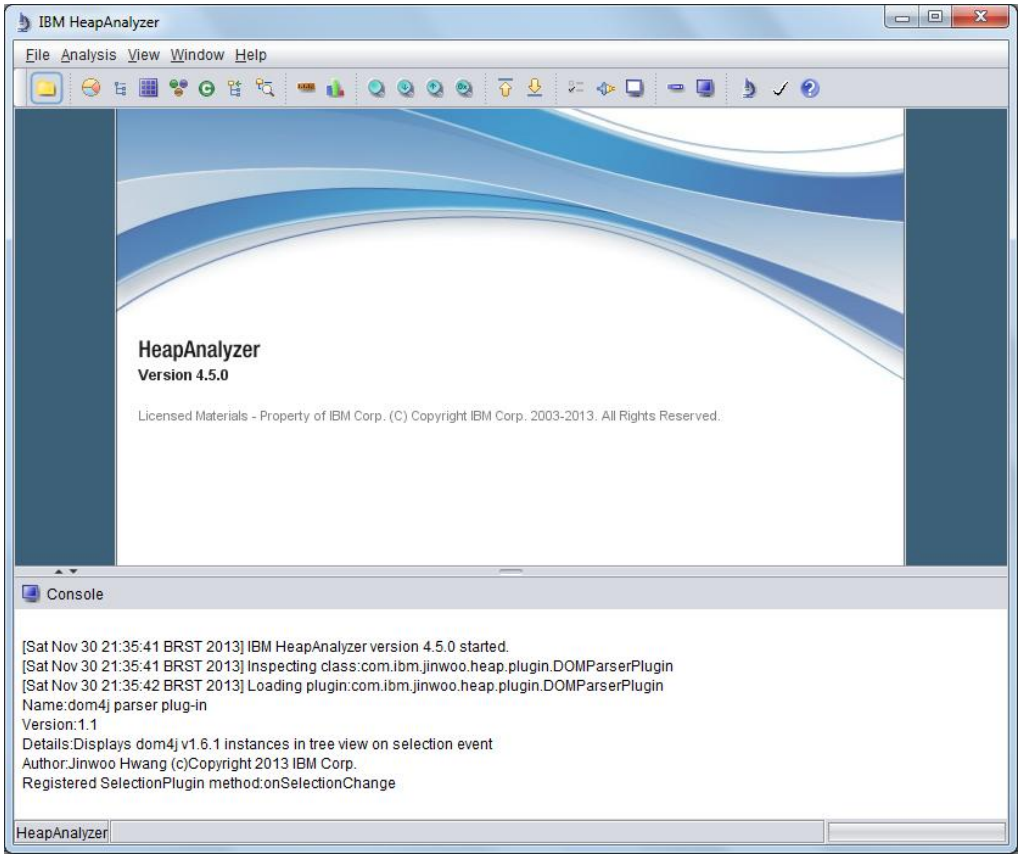
2- Executar o comando: java -Xmx1200m -jar <path to ha jar>/ha450.jar

OBSERVAÇÃO:
O Java precisa estar instalado na máquina onde a ferramenta será executada

Análise de JVM

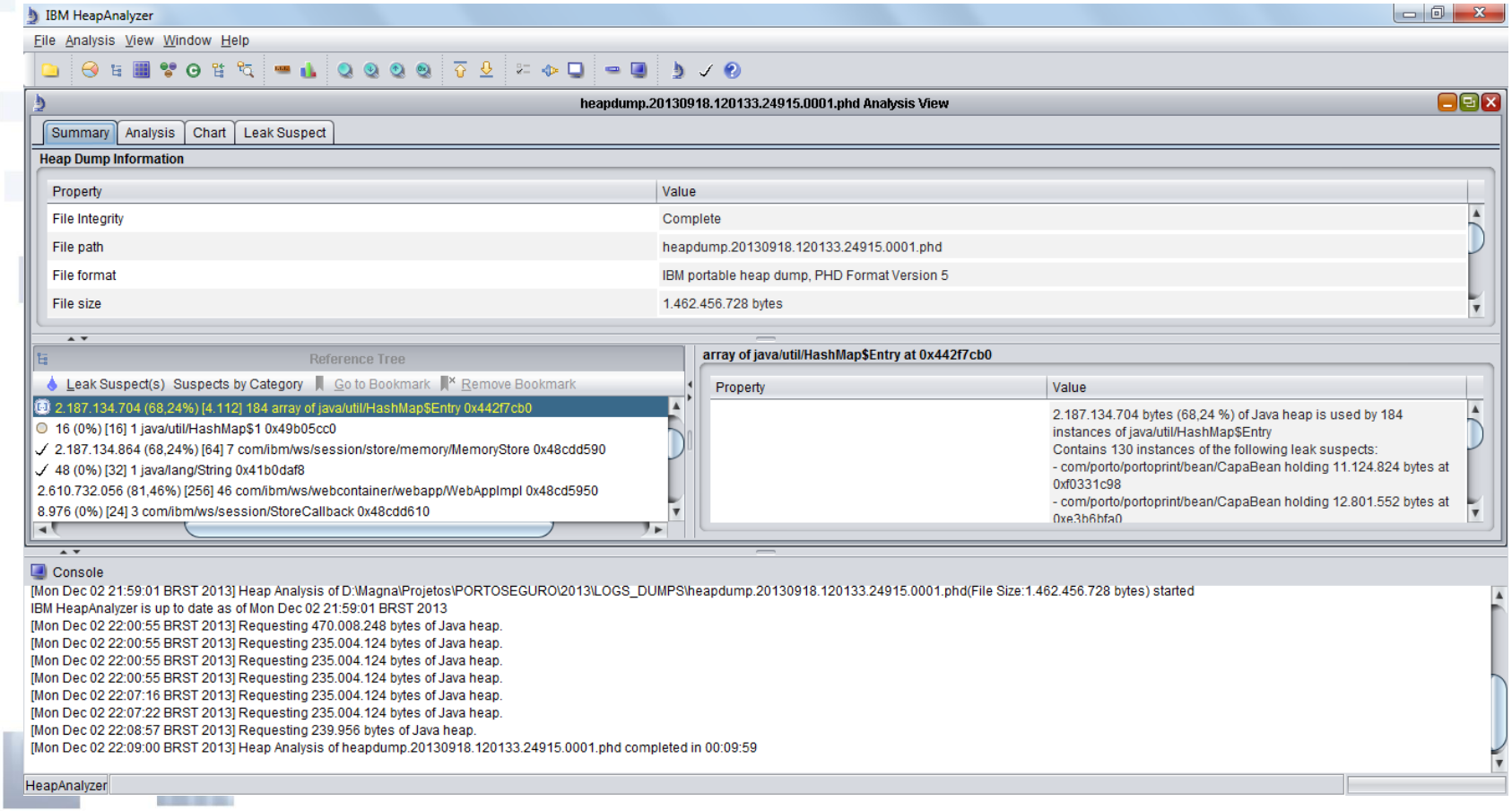
IBM Heap Dump Analyzer – Abrindo a ferramenta

```
C:\Program Files (x86)\Java\jre7\bin>java.exe -Xmx1200m -jar d:\softwares\ha450.
jar
```



Análise de JVM

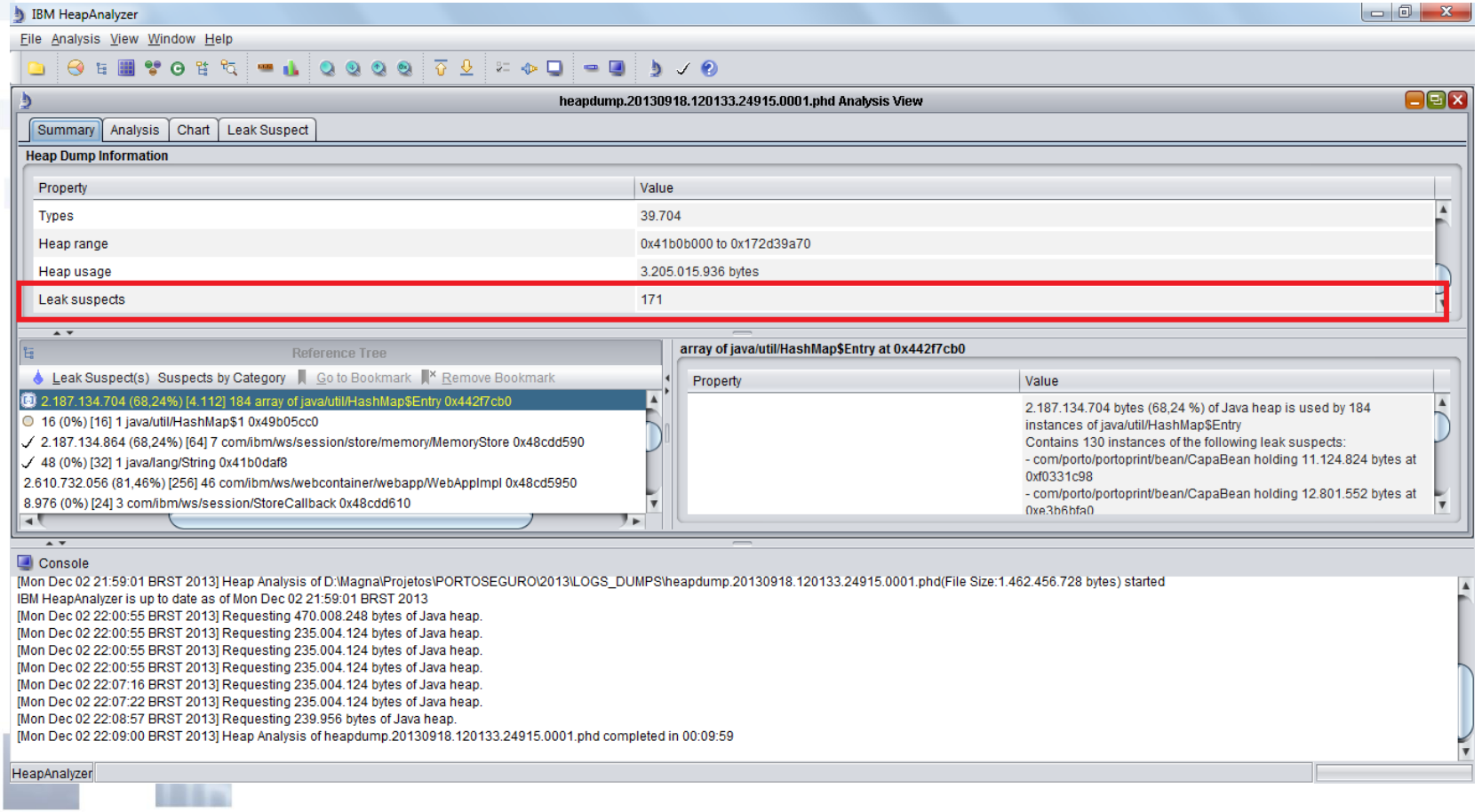
Resultado do Heapdump - Tela Inicial



Paineis de análise

Análise de JVM

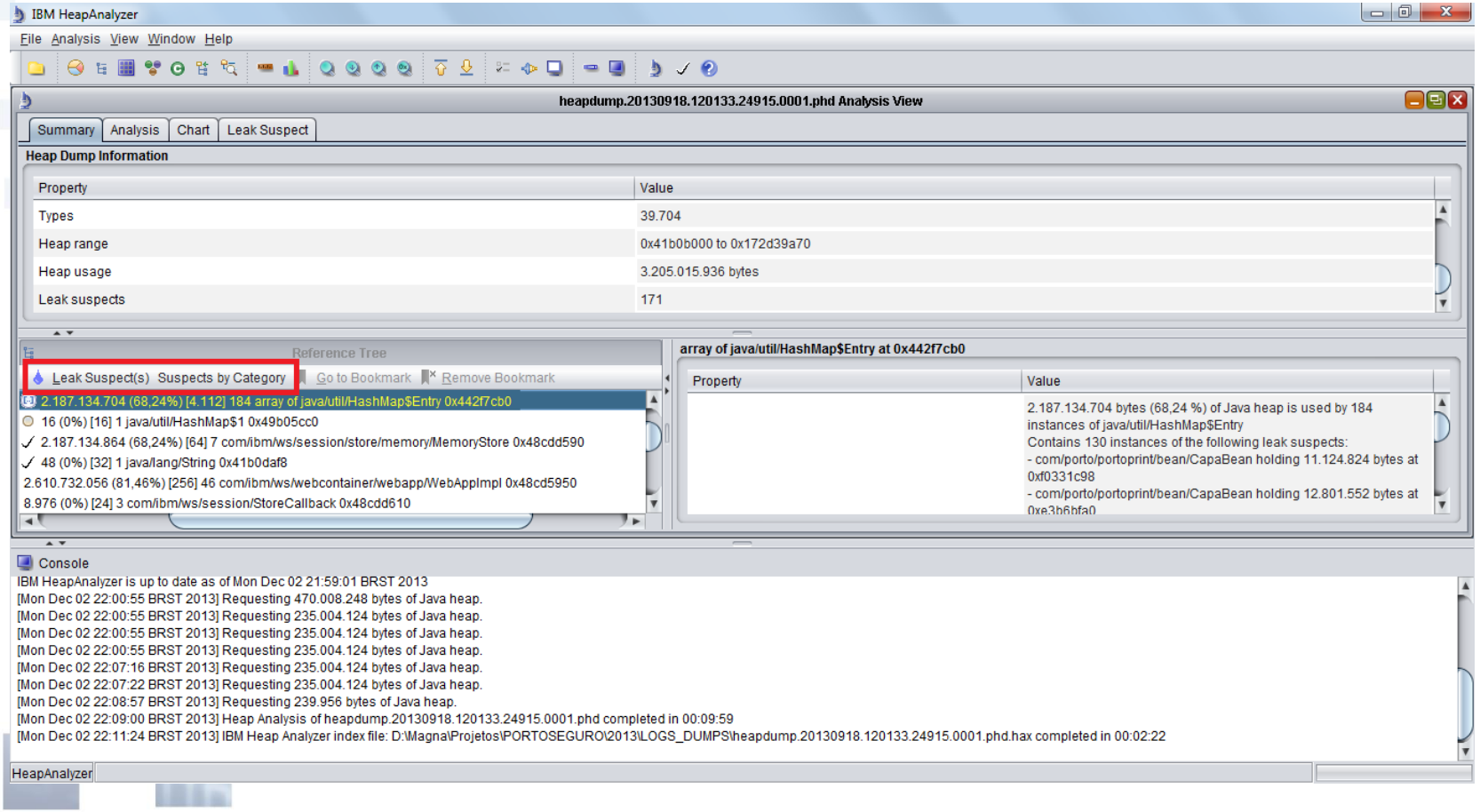
Resultado do Heapdump – Identificando Leak Suspects 1



No painel do topo descer o scroll e verificar se existem ML.

Análise de JVM

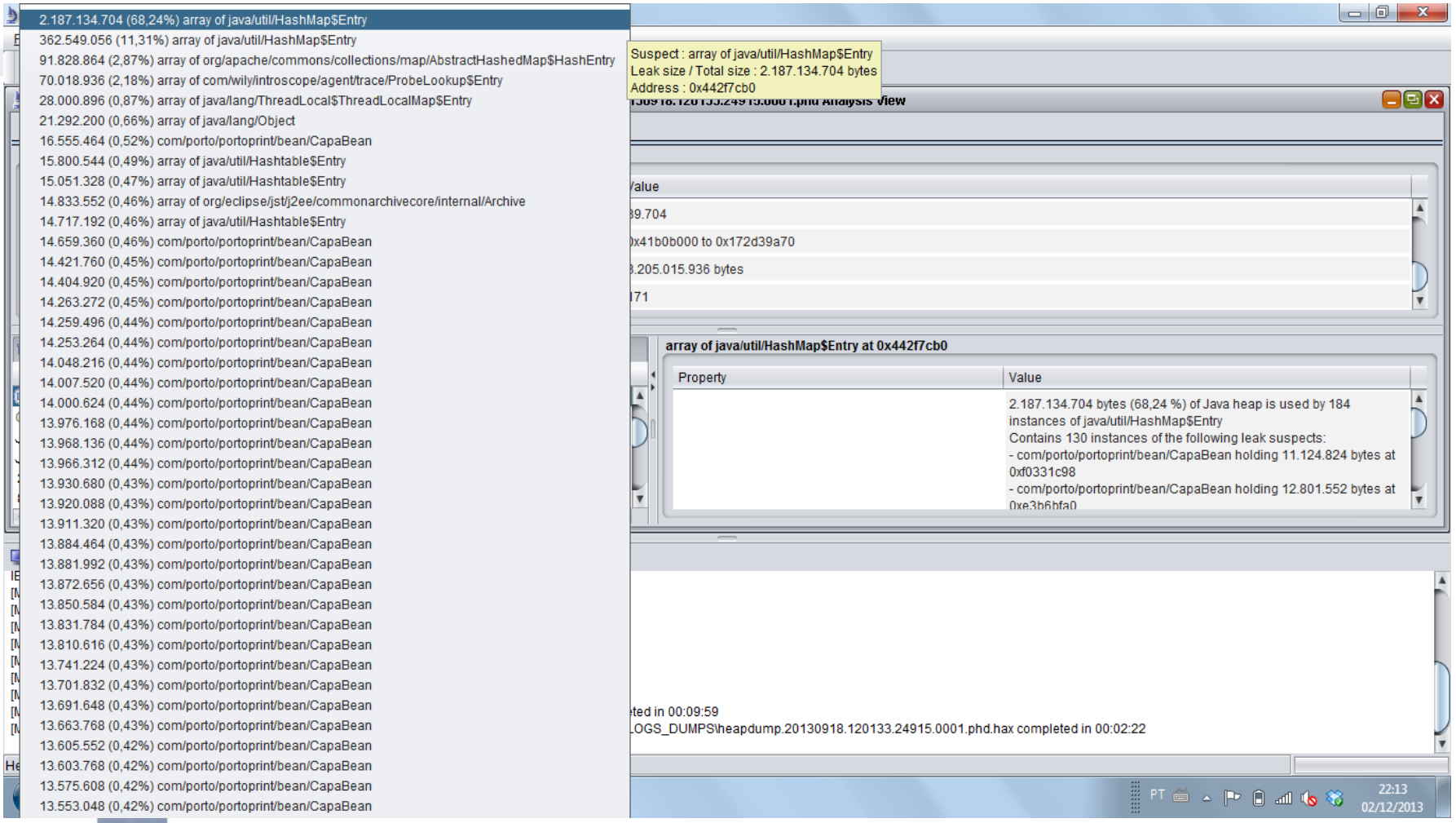
Resultado do Heapdump – Identificando Leak Suspects 2



Acessando a lista dos Leak Suspects.

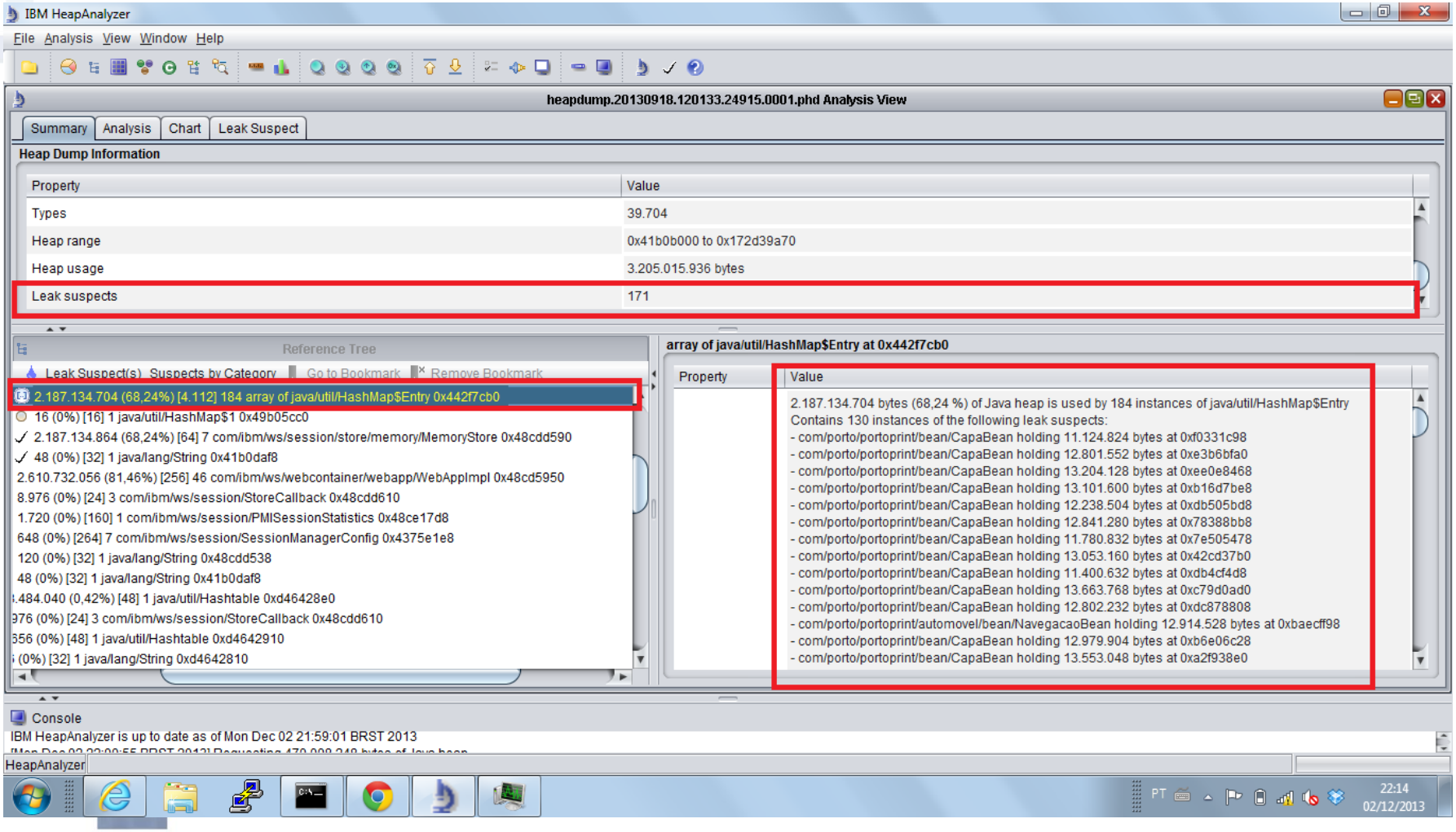
Análise de JVM

Resultado do Heapdump – Identificando Leak Suspects 2.1



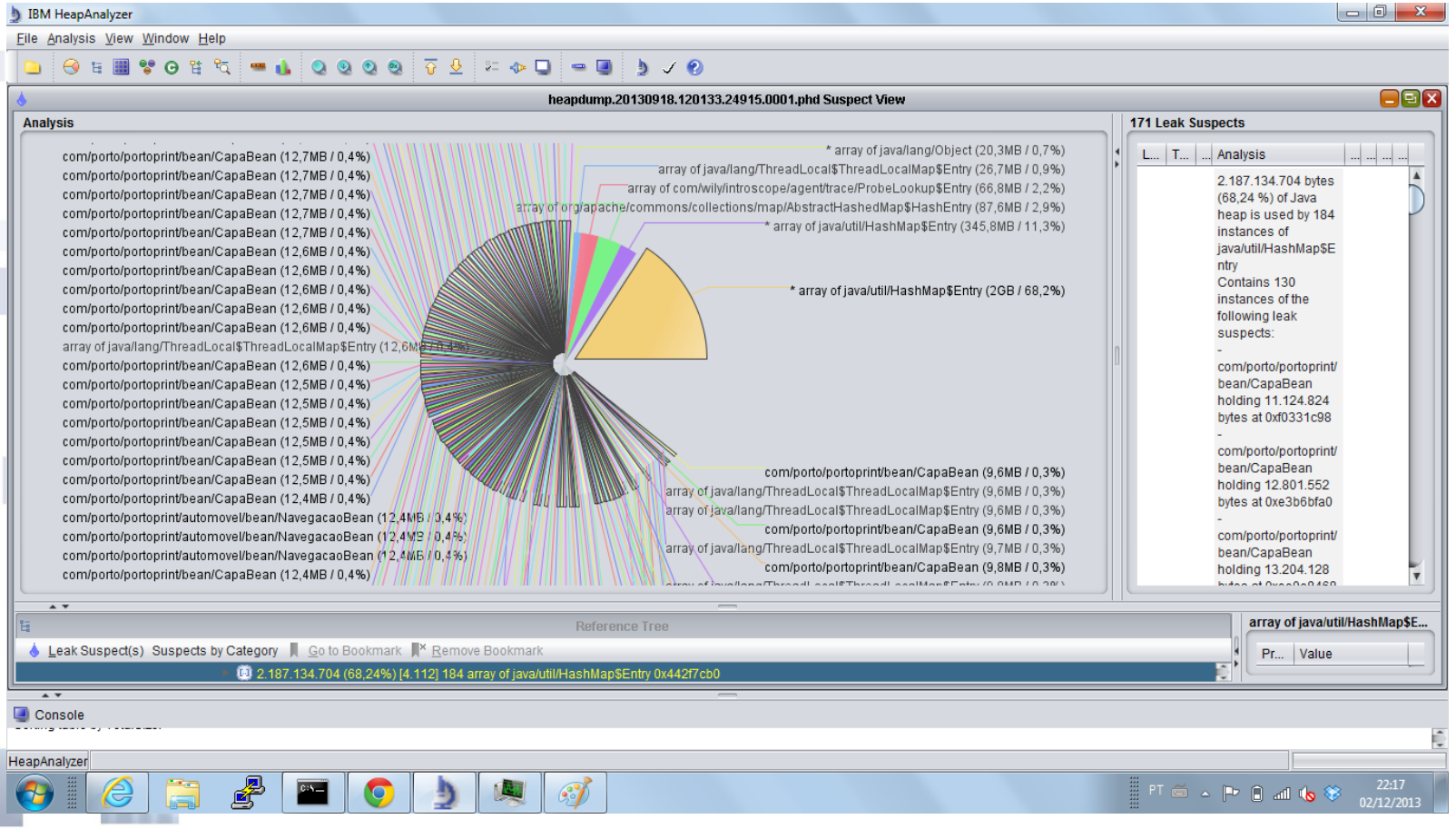
Análise de JVM

Resultado do Heapdump – Identificando Leak Suspects 2.2



Análise de JVM

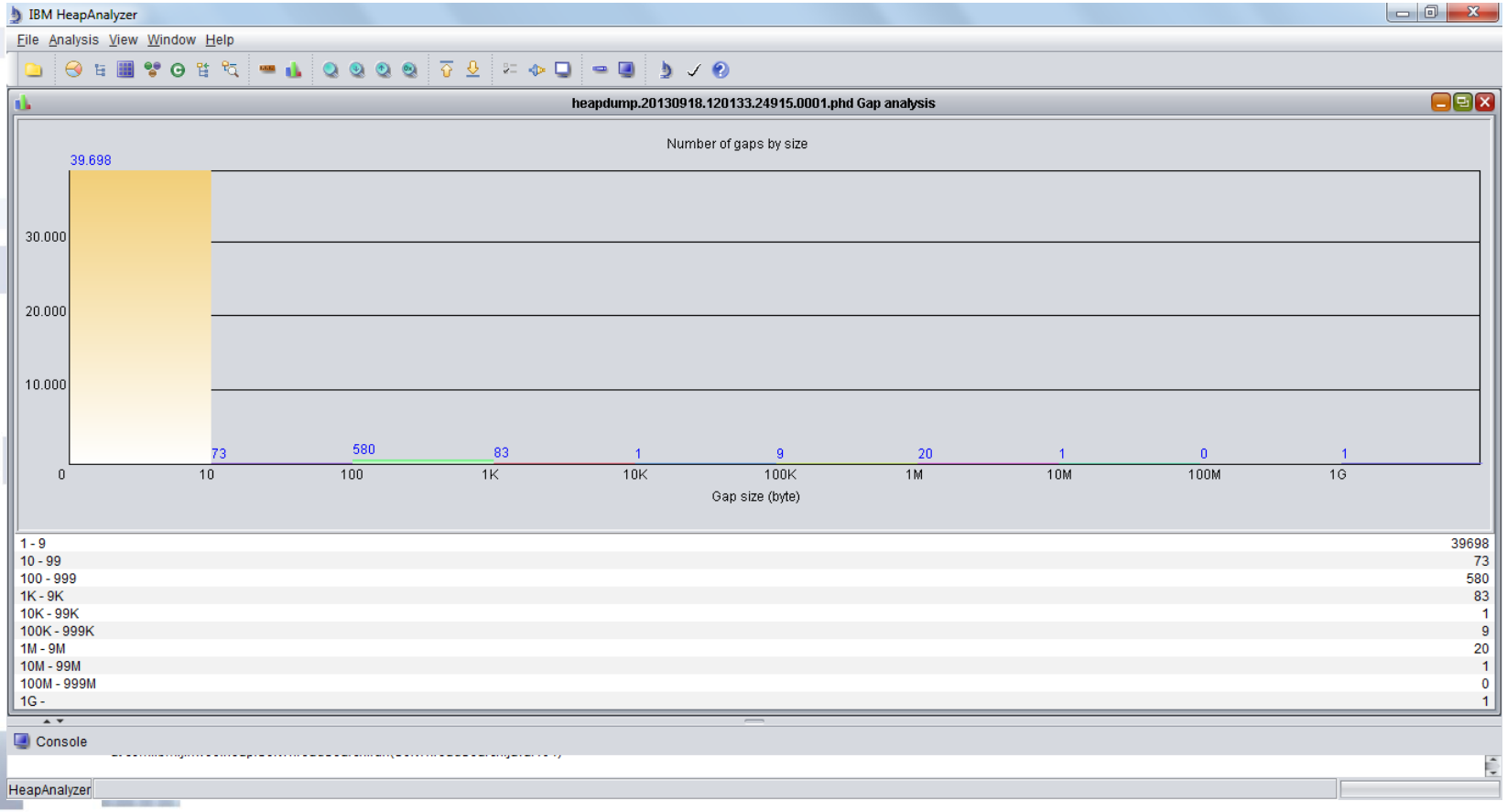
Resultado do Heapdump – Leak Suspect Pie Chart



Apontamento gráfico de todos os Leak Suspects

Análise de JVM

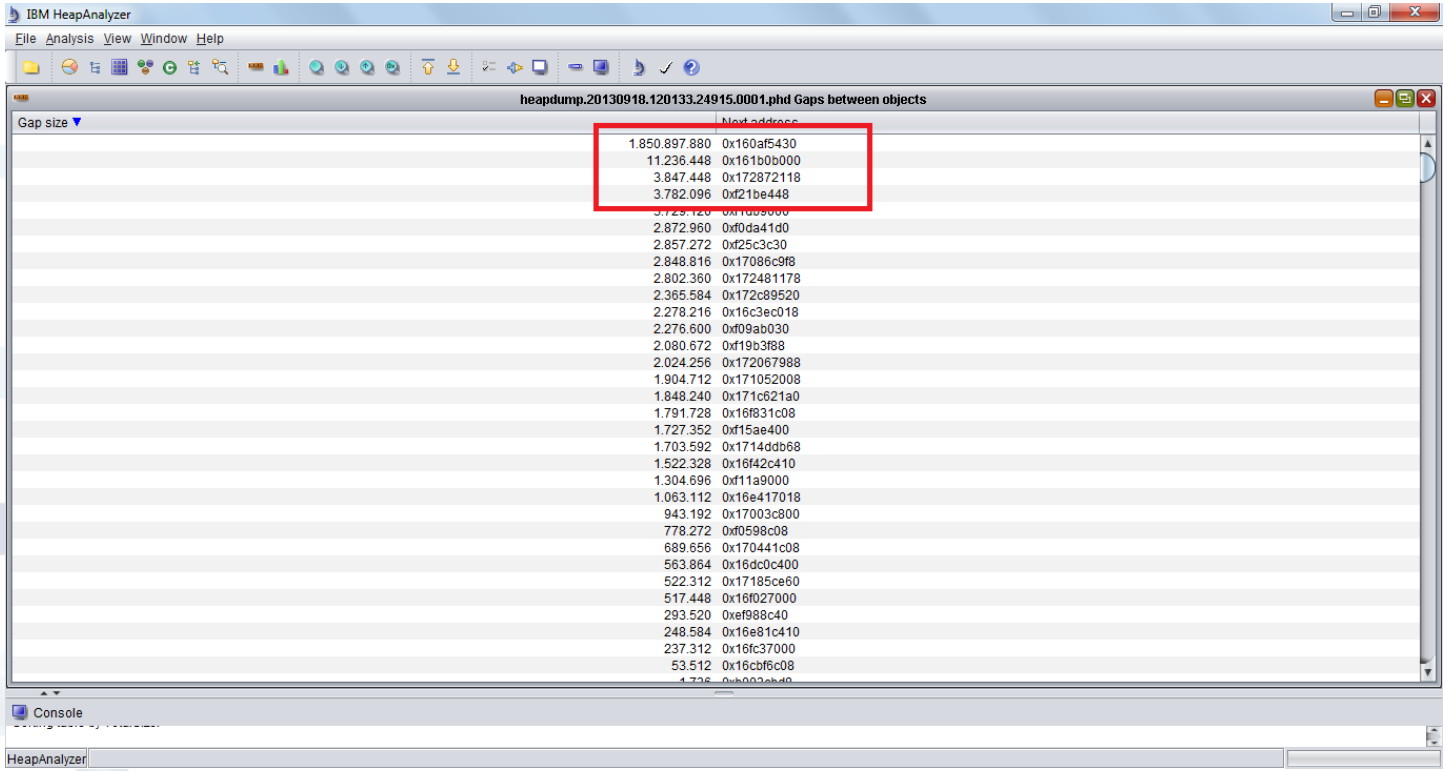
Resultado do Heapdump – Gap Statistics



Tamanho dos objetos alocados na JVM

Análise de JVM

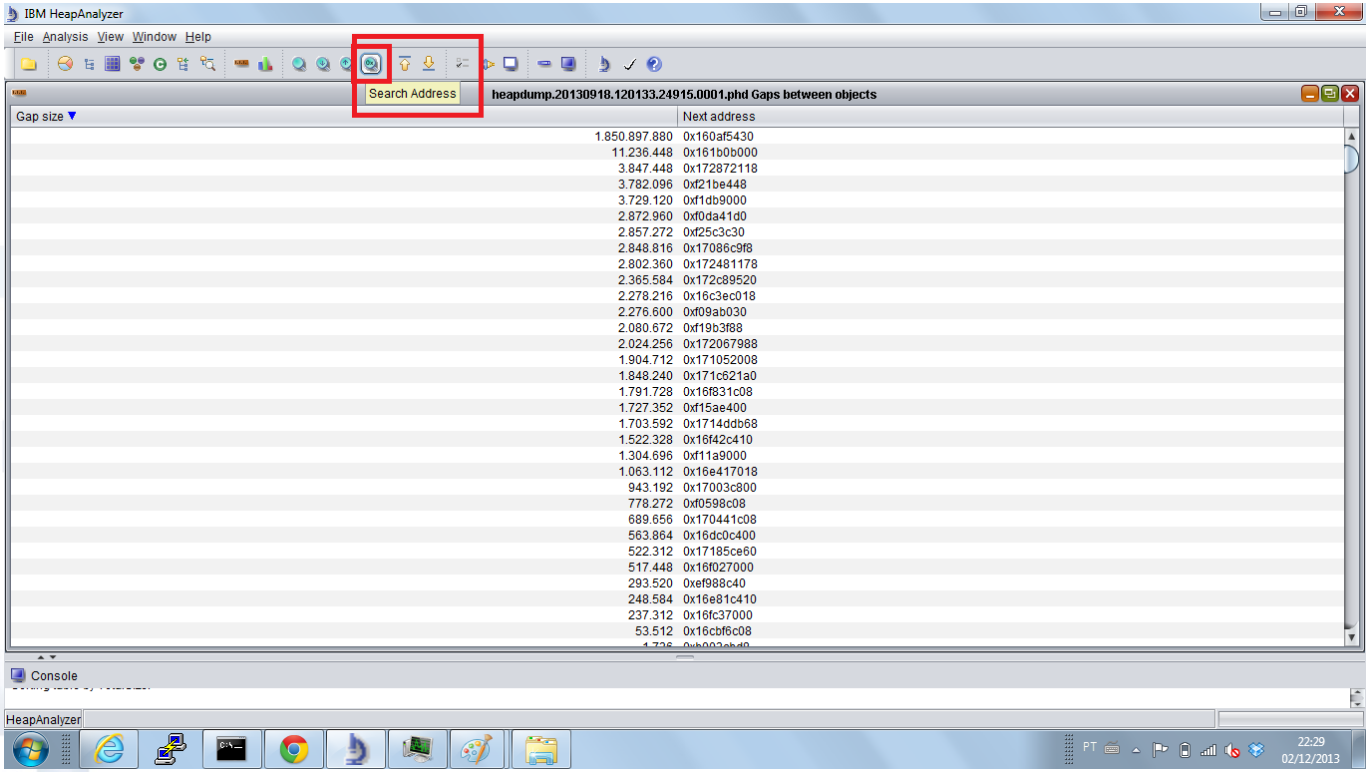
Resultado do Heapdump – Gap By Size 1



Lista de todos os objetos alocados na heap. Nessa lista é possível copiarmos o endereço de memória e fazer um drill down ou drill up até identificar o objeto que esta causando a alocação de todo este espaço

Análise de JVM

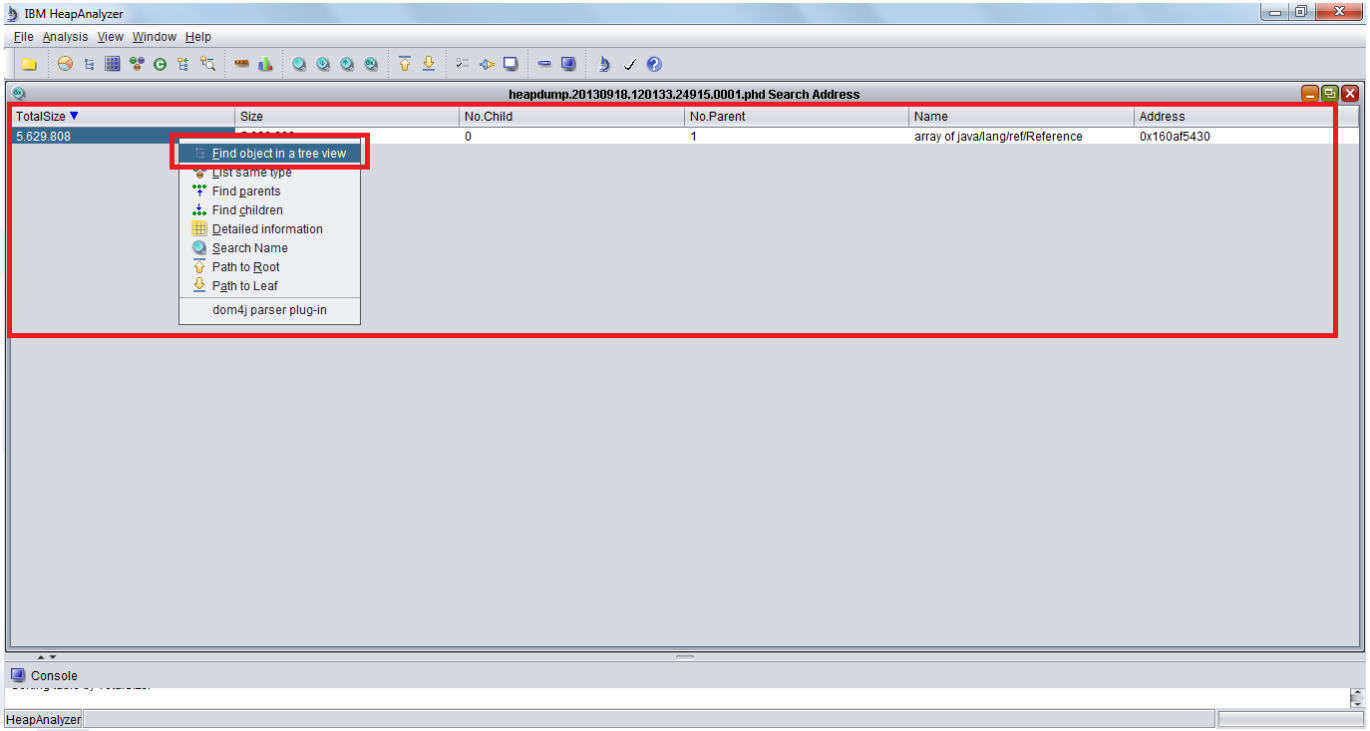
Resultado do Heapdump – Gap By Size 2



Lista de todos os objetos alocados na heap. Nessa lista é possível copiarmos o endereço de memória e fazer um drill down ou drill up até identificar o objeto que esta causando a alocação de todo este espaço

Análise de JVM

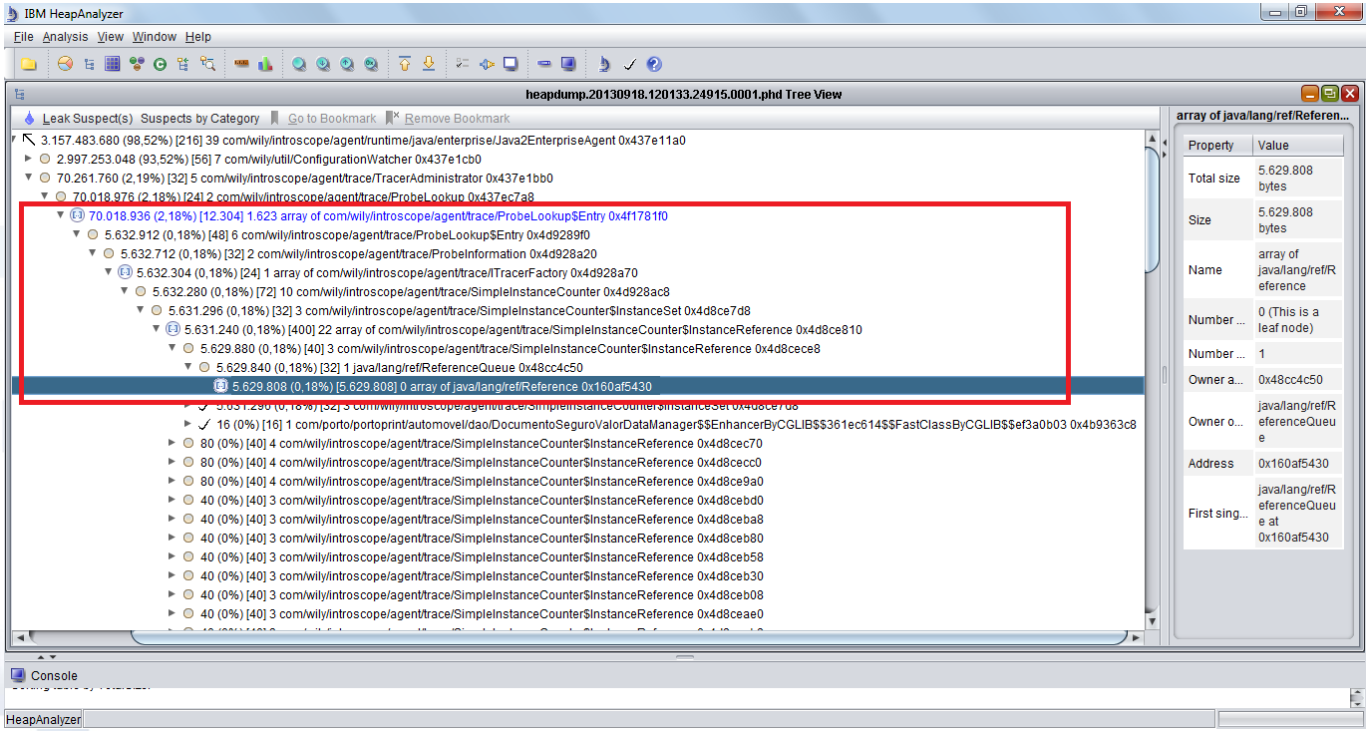
Resultado do Heapdump – Gap By Size 3



Ainda podemos clicar com o botão direito em cima do resultado da pesquisa e procurar o objeto referente a este endereço de memória

Análise de JVM

Resultado do Heapdump – Gap By Size 4

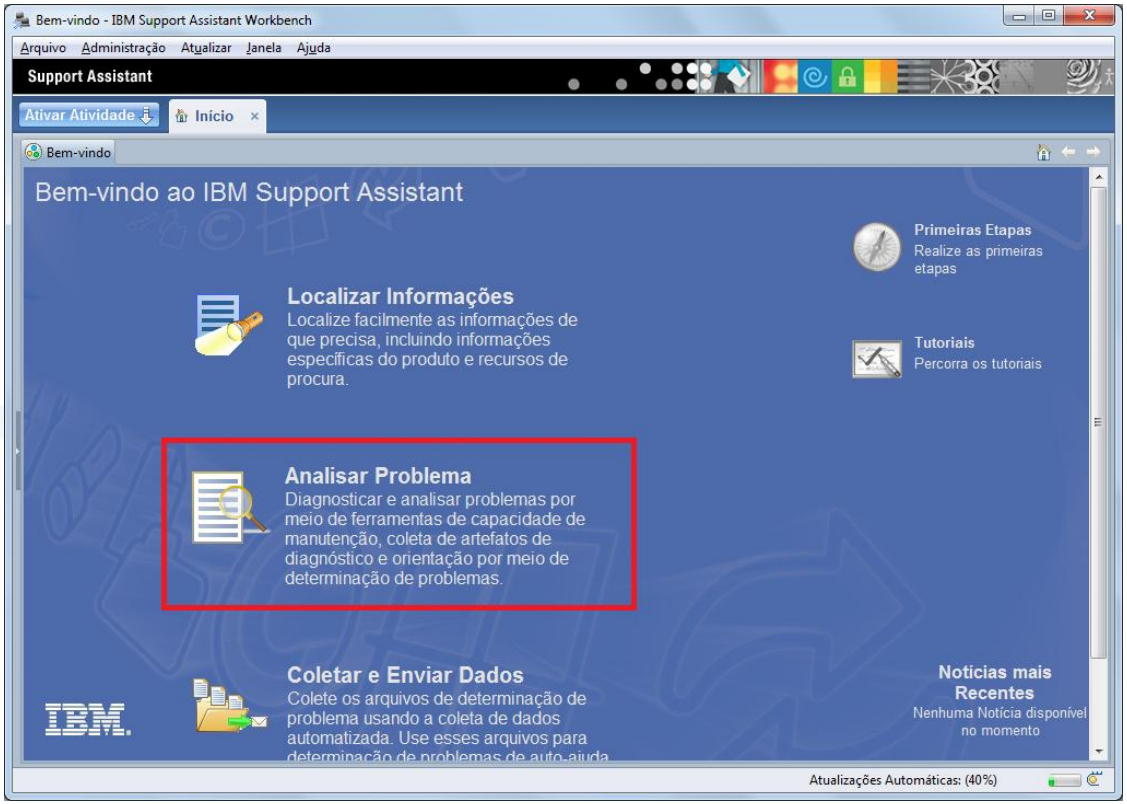


Este é o objeto que esta causando a alocação de todo aquele espaço encontrado anteriormente nas analises.

Análise de Garbage Collection

Acessando a ferramenta GCMV

Passo 1 – Abrindo a ferramenta através do ISA

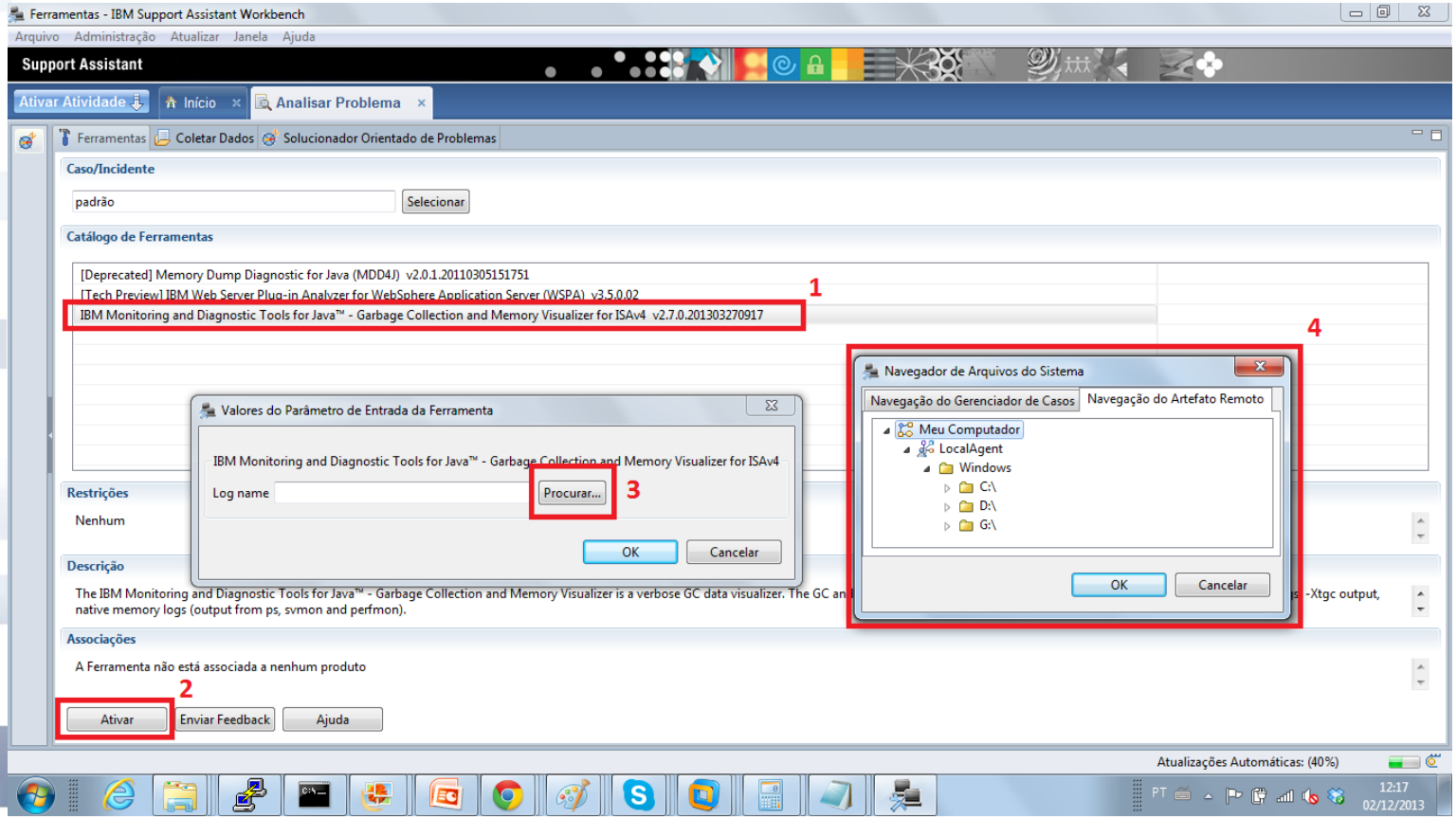


Análise de Garbage Collection

Acessando os logs

MAGNASISTEMAS

Passo 1 – Abrindo o native_stderr.log(informações doGC)



Análise de Garbage Collection

Pontos que usaremos para análise

- **Memória da JVM -** Tamanho da JVM, Uso da JVM, restarts de JVM

-**JVM Objects Size -** Tamanho dos objetos trafegados na JVM

-**Tempo do GC(Pause time)** – Tempo que o GC demora para ser executado

- **Intervalos entre GC** – Tempo entre os GCs

- **Advisor** – Relatório automático gerado com sugestões de melhoria ao ambiente

Análise de Garbage Collection

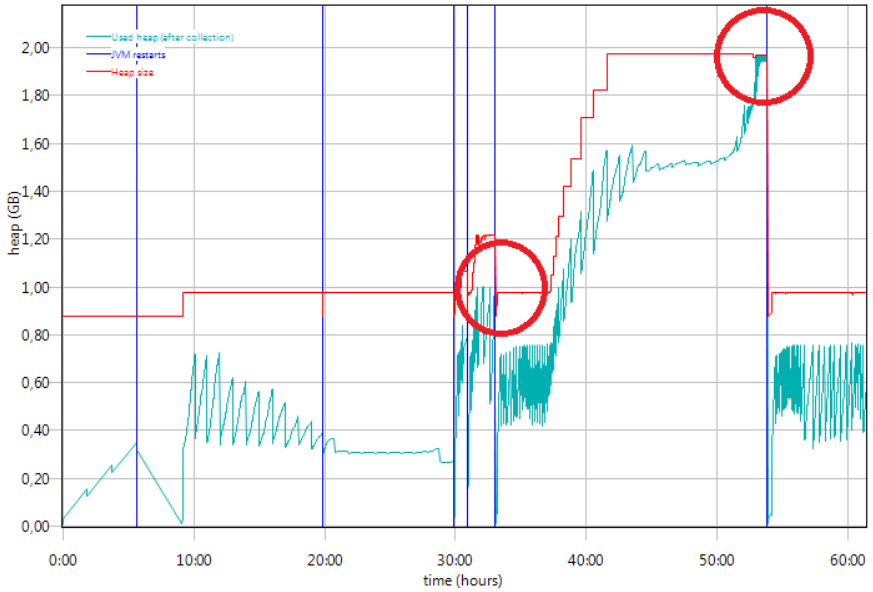
Memória da JVM – Analise básica

Pontos Importantes para análise:

Heap Size:
O tamanho da JVM atual

Used Heap(after Collection)
Uso real da JVM – após GC

JVM Restarts
Processo de restart manual ou devido a algum problema

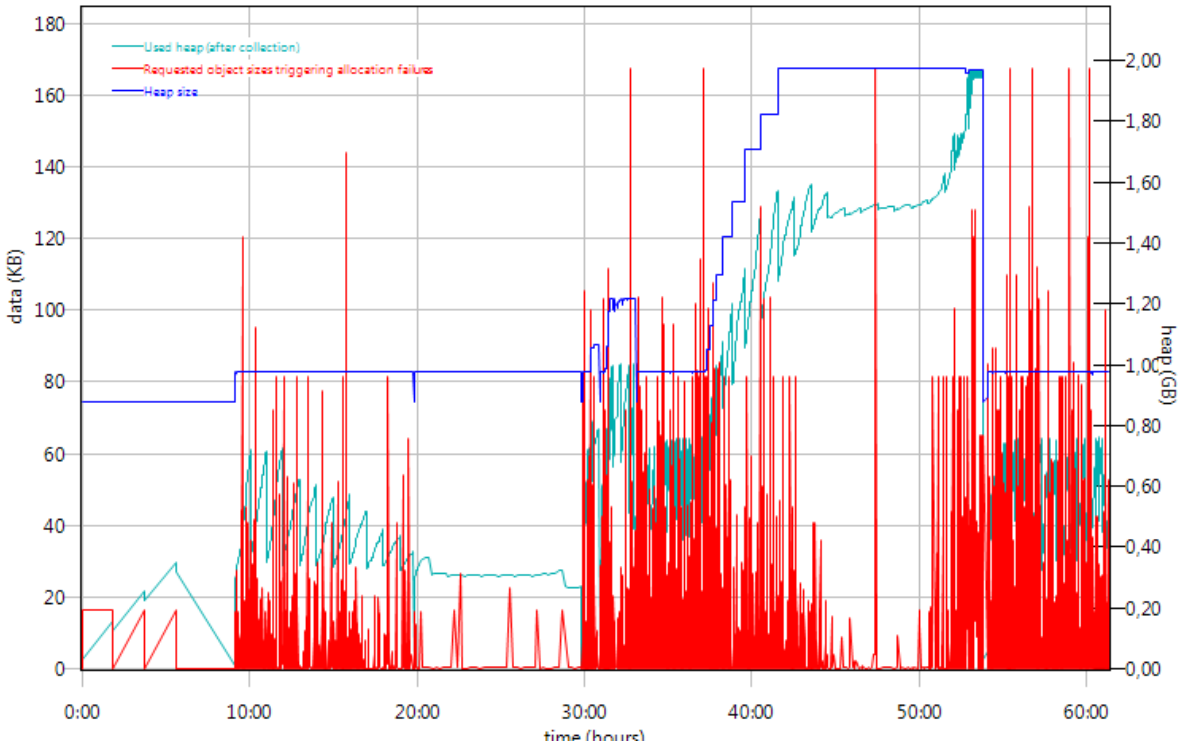


Identificando Problemas

Quando o used heap encosta na linha do Heap Size, pode haver algum problema de Memory Leak, ou ajuste no GC, pois o GC não esta conseguindo excluir os objetos instanciados(sem/com referências) em memória corretamente
Nos pontos em vermelho as linhas se encostam e ocorre um restart por falha

Análise de Garbage Collection

JVM objects size – Analise básica



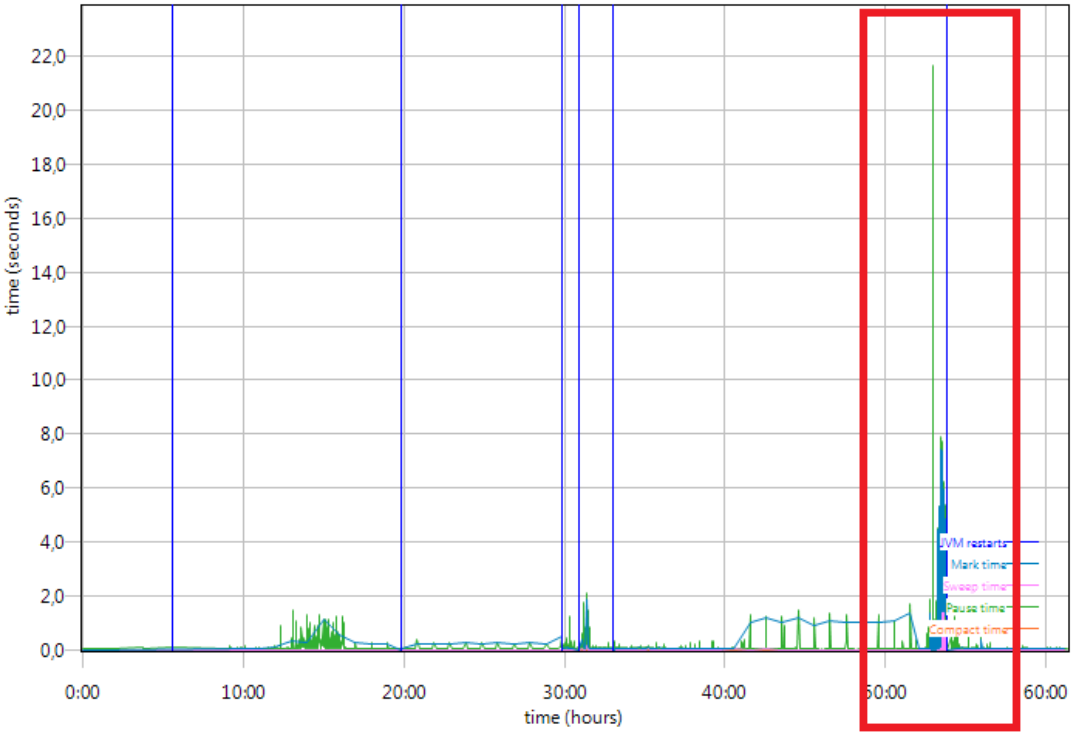
Pontos Importantes para análise:

Requested object sizes triggering allocation failures

Tamanho de objetos que geraram allocation failures

Análise de Garbage Collection

Pause time – Analise básica



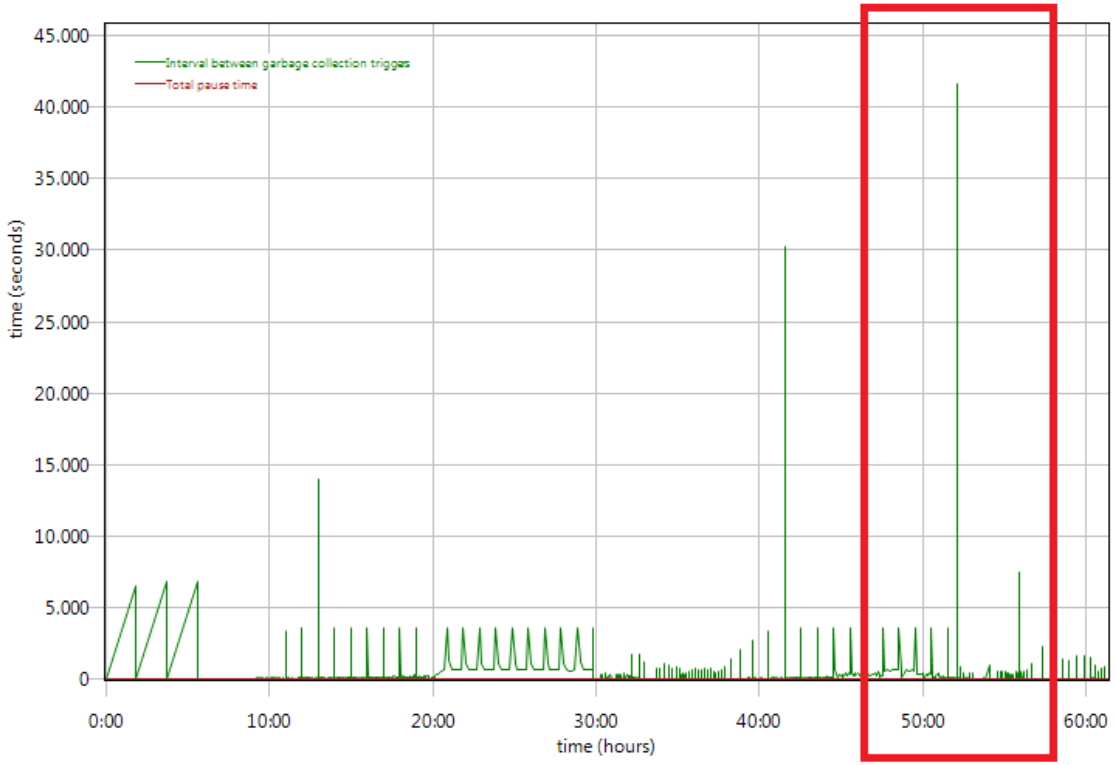
Pontos Importantes para análise:

Pause Time

Tempo que o GC demora para ser executado.

Análise de Garbage Collection

Interval between garbage collection triggers



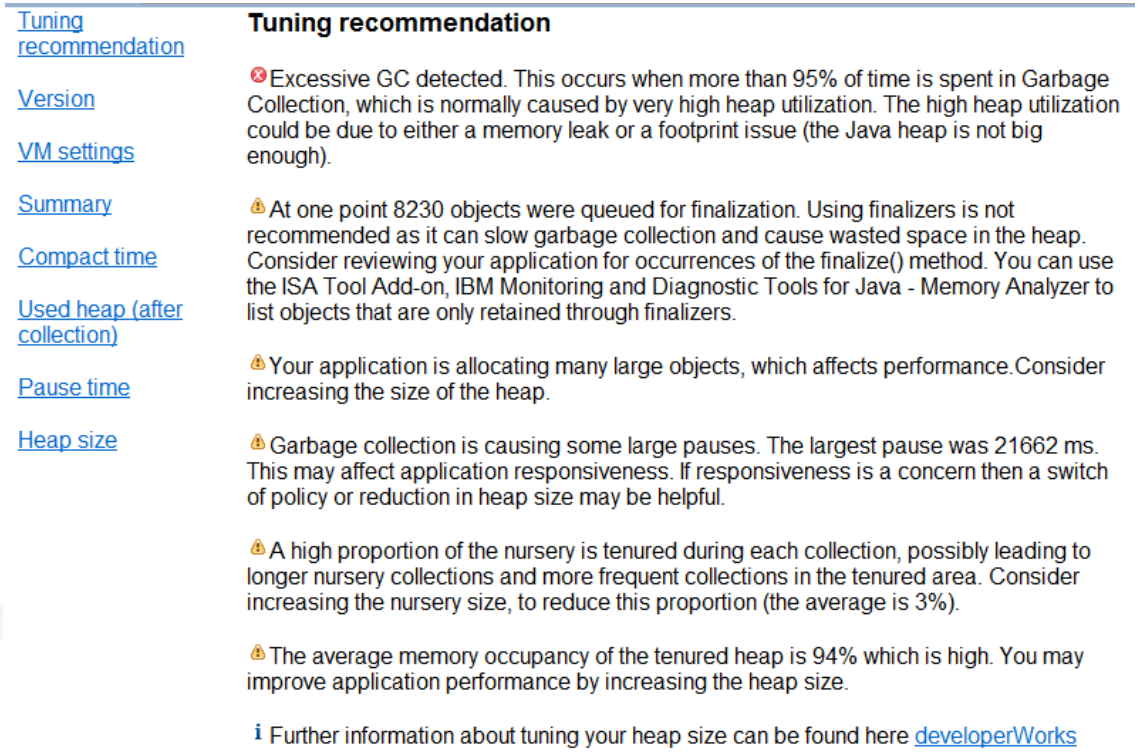
Pontos Importantes para análise:

Interval between garbage collection triggers

Tempo entre os GCs

Análise de Garbage Collection

Relatório - Advisors



Tuning recommendation

Version

VM settings

Summary

Compact time

Used heap (after collection)

Pause time

Heap size

Tuning recommendation

Excessive GC detected. This occurs when more than 95% of time is spent in Garbage Collection, which is normally caused by very high heap utilization. The high heap utilization could be due to either a memory leak or a footprint issue (the Java heap is not big enough).

At one point 8230 objects were queued for finalization. Using finalizers is not recommended as it can slow garbage collection and cause wasted space in the heap. Consider reviewing your application for occurrences of the finalize() method. You can use the ISA Tool Add-on, IBM Monitoring and Diagnostic Tools for Java - Memory Analyzer to list objects that are only retained through finalizers.

Your application is allocating many large objects, which affects performance.Consider increasing the size of the heap.

Garbage collection is causing some large pauses. The largest pause was 21662 ms. This may affect application responsiveness. If responsiveness is a concern then a switch of policy or reduction in heap size may be helpful.

A high proportion of the nursery is tenured during each collection, possibly leading to longer nursery collections and more frequent collections in the tenured area. Consider increasing the nursery size, to reduce this proportion (the average is 3%).

The average memory occupancy of the tenured heap is 94% which is high. You may improve application performance by increasing the heap size.

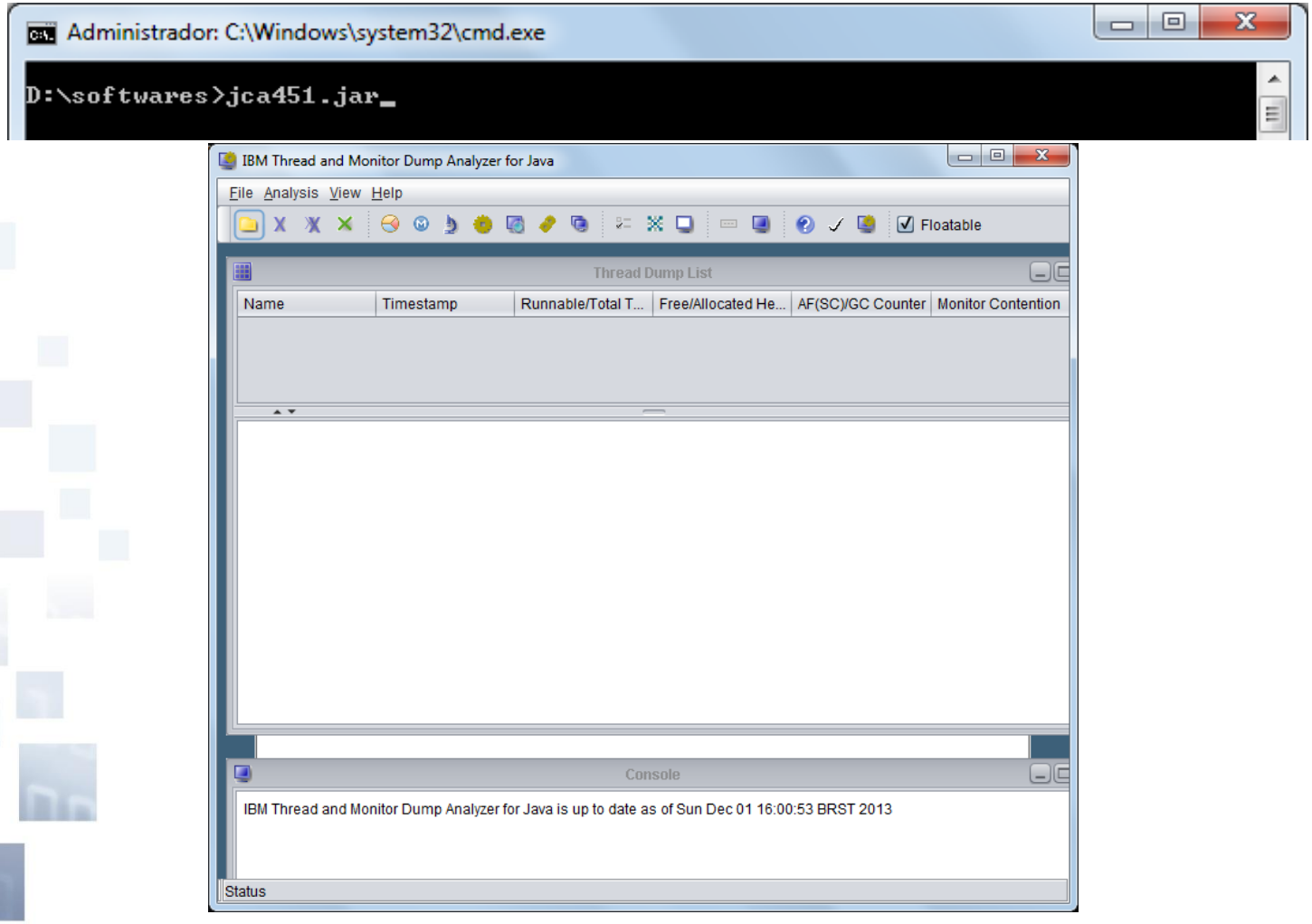
i Further information about tuning your heap size can be found here developerWorks

Tuning Recommendation

No tuning recommendation a prória ferramenta logo após de realizar o parse do arquivo, faz uma análise e a partir disso ela fornece algumas sugestões que podem ser feitas para melhorar o desempenho do servidor/aplicação.

Análise de Thread Dump

Abrindo a ferramenta



Análise de Thread Dump

Pontos importantes de analise

MAGNASISTEMAS

Thread que analisaremos

Deadlock – Threads em deadlock

Runnable – Threads em execução

Waiting on condition – Thread em wait. Ex:sleep()

Blocked – A thread esta bloqueada(depende de outra)

Status	Number of Threads : 232	Percentage
Deadlock	0	
Runnable	61	
Waiting on condition	158	
Waiting on monitor	0	
Suspended	0	
Object.wait()	0	
Blocked	11	
Parked	2	

1 – Deadlock – Caso haja qualquer deadlock, abrir a thread específica e analisar qual objeto específico esta causando este lock

2 – Blocked – Pode ser um comportamento normal poucos blocked, mas se o número estiver muito alto, vale analisar o que esta acontecendo

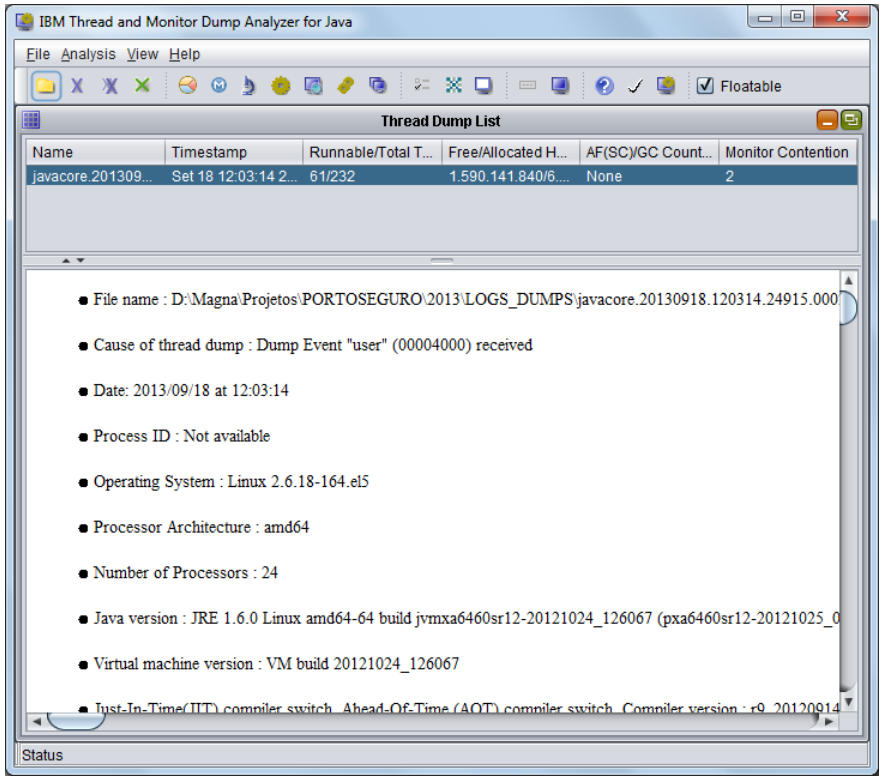
Análise de Thread Dump

Informações

Outras informações

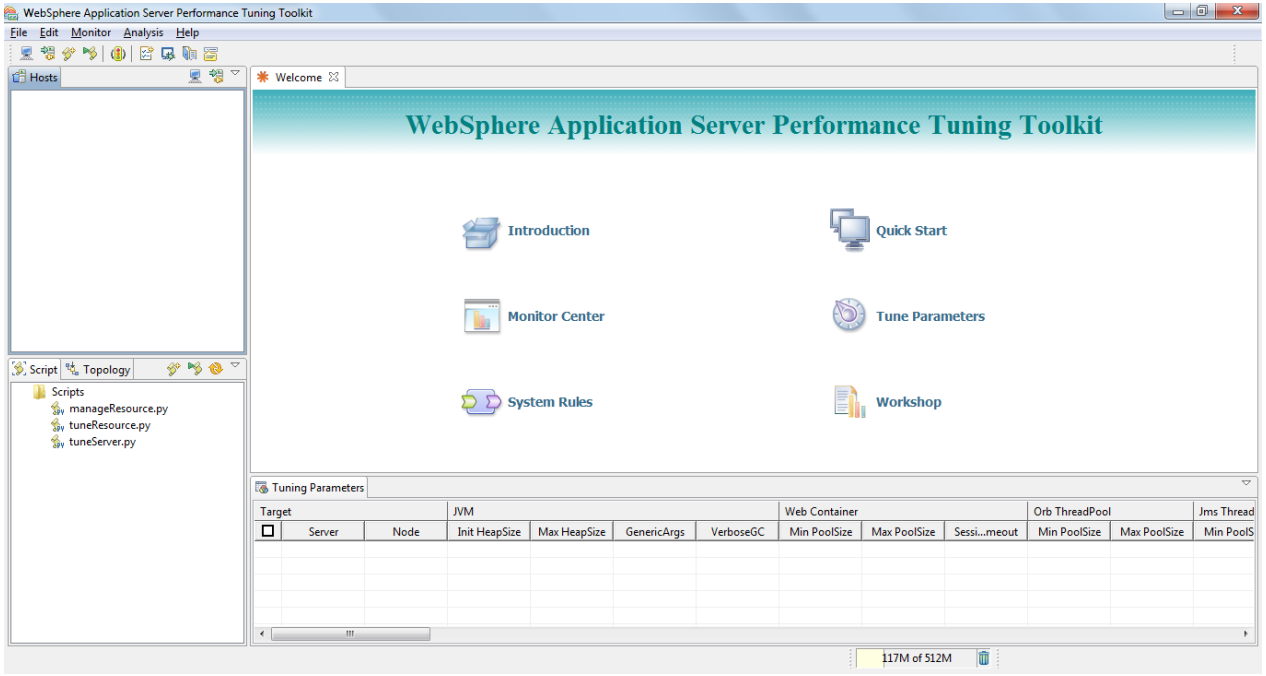
Quando abrimos o log na ferramenta a tela inicial nos mostra diversas informações da JVM e do servidor.

Isso nos ajuda a ver qual arquitetura em relação a servidor e também ao java no qual estamos utilizando



Análise Runtime

IBM WebSphere Performance Tuning Toolkit(IWPTT)



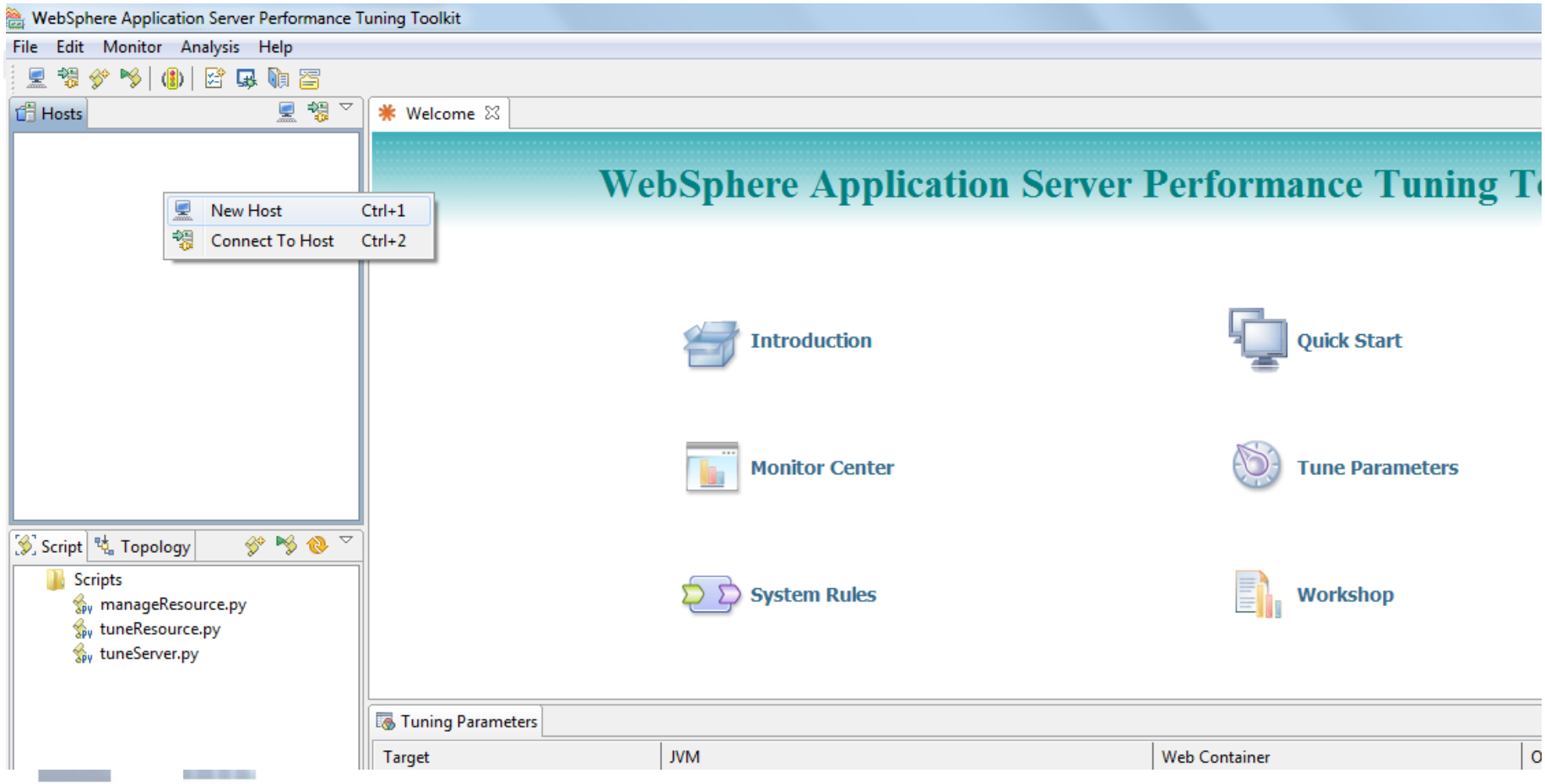
Informações importantes

Para funcionar o PMI deve estar habilitado no server, conforme demonstrado em slides anteriores

O servidor que executar a ferramenta necessita ter acesso ao servidor na porta SOAP(default 8879)

Análise Runtime

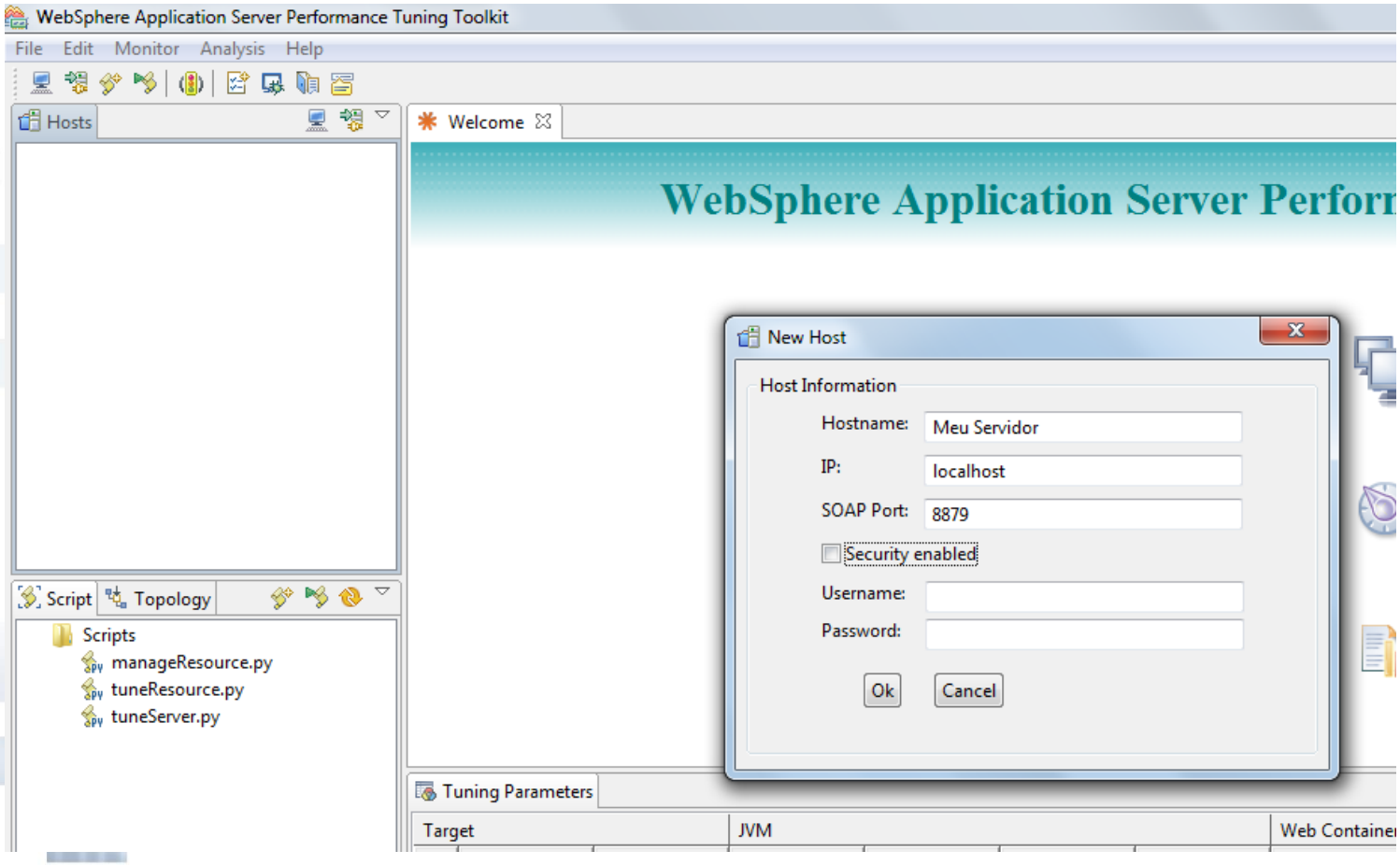
IWPTT – Conectando no Application Server



No painel Hosts, clicar com o botão direito e depois em New Host

Análise Runtime

IWPTT – Conectando no Application Server



Prrencha os dados solicitados conforme o exemplo acima

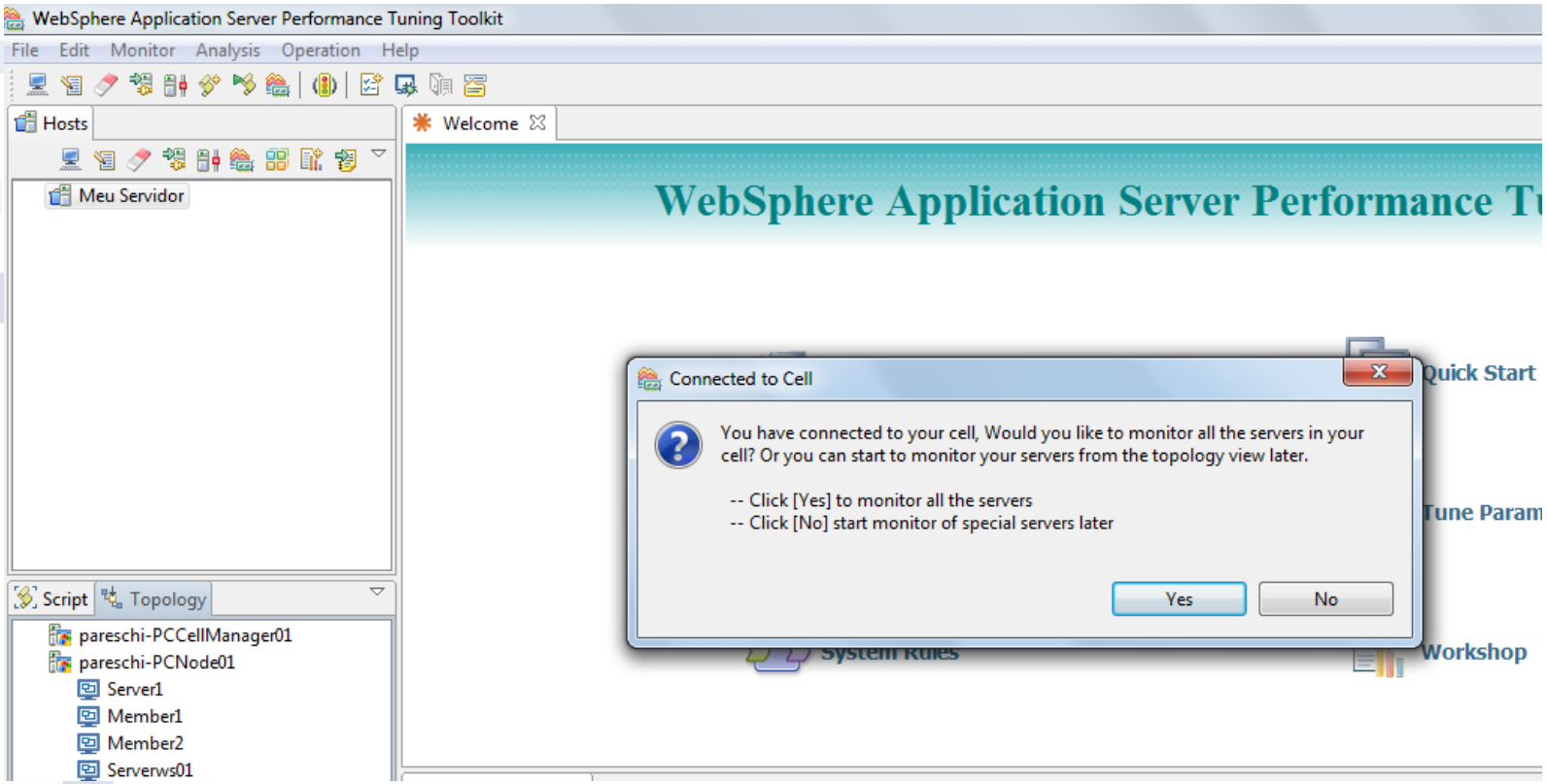
Análise Runtime

IWPTT – Conectando no Application Server

No servidor criado no painel hosts clicar com o botão direito em Connect to host

Análise Runtime

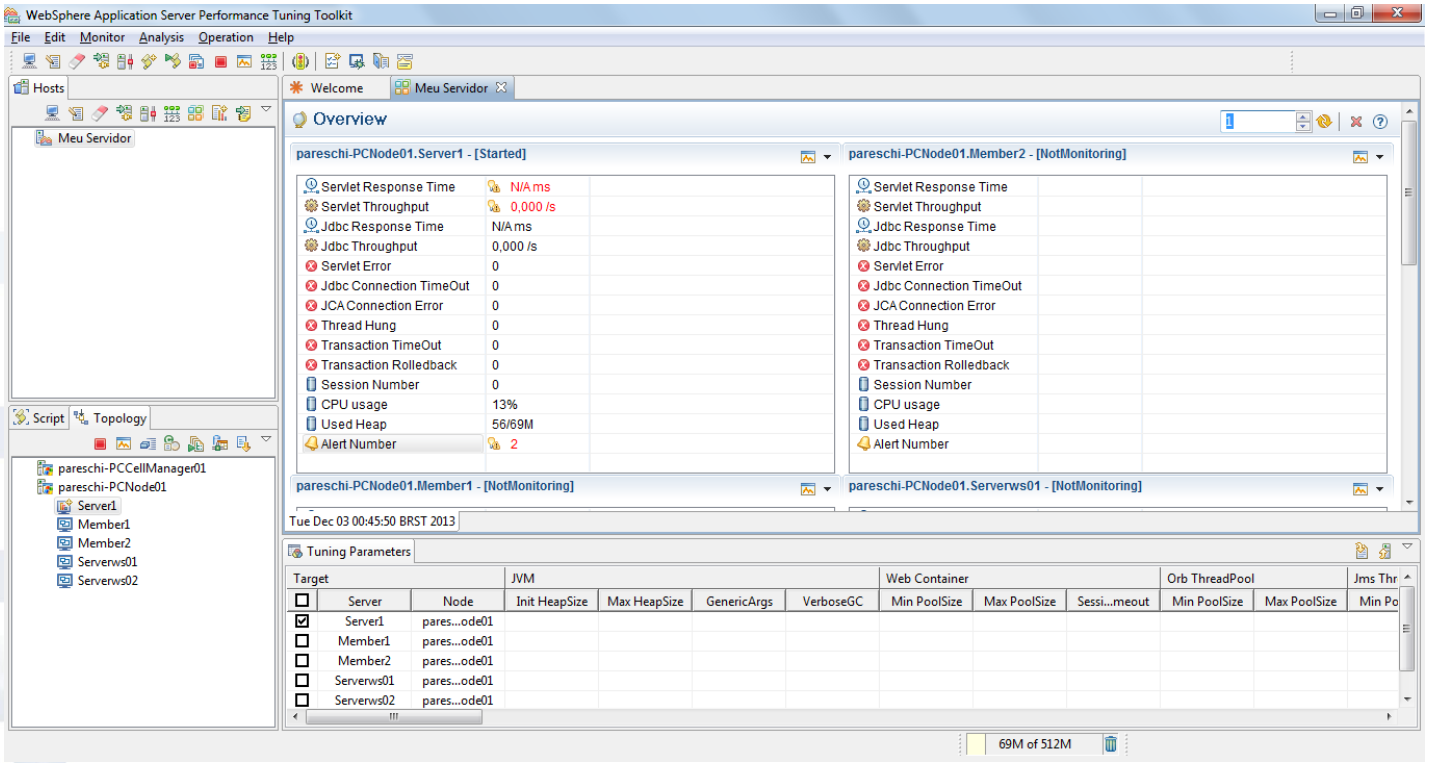
IWPTT – Conectando no Application Server



A ferramenta identifica se o servidor faz parte de uma célula, e pergunta se quer monitorar todos os application servers

Análise Runtime

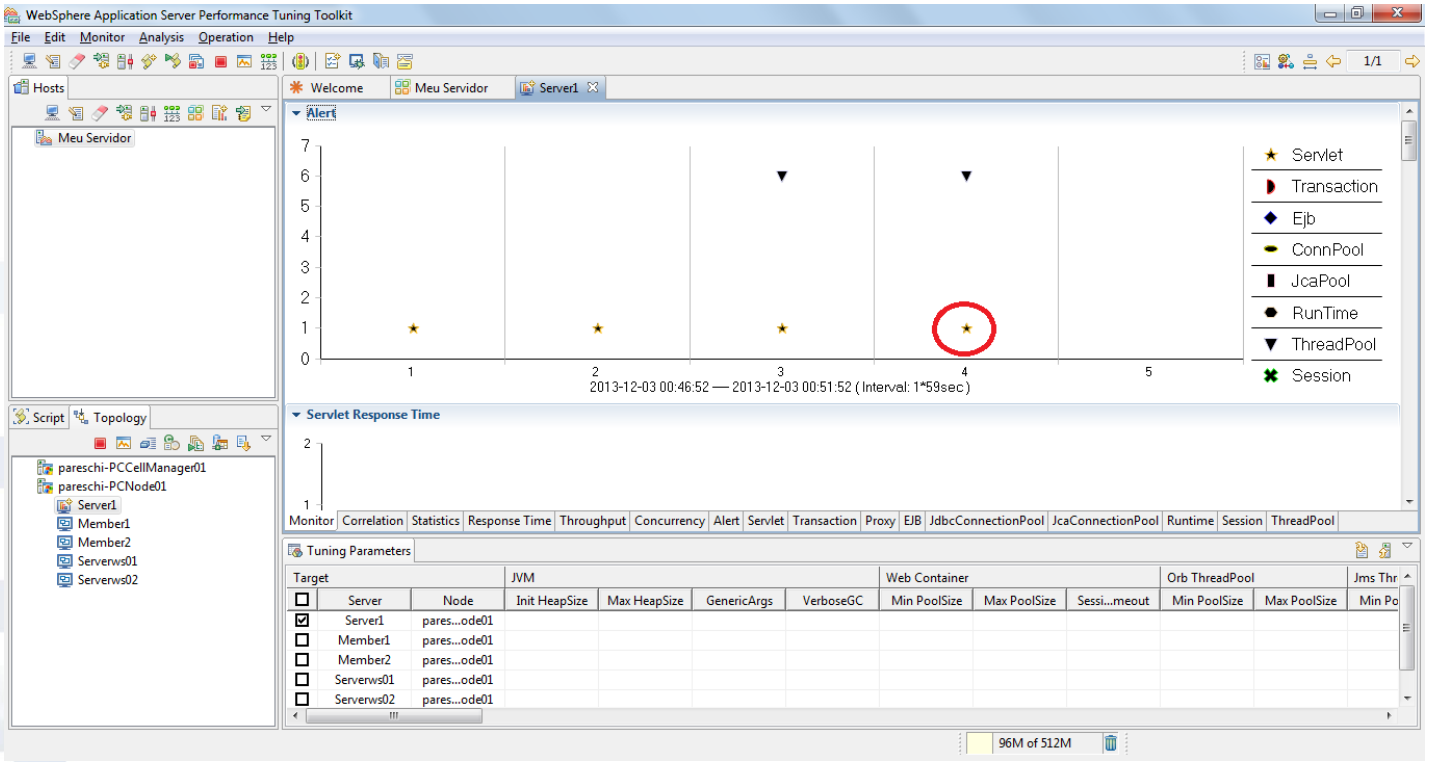
IWPTT – Visão resumida



Assim que conectado temos um dashboard apresentando os principais pontos de monitoria da aplicação e alertas padrões do próprio produto, por exemplo se o servidor atinge 90% do Pool de threads

Análise Runtime

IWPTT – Alertas 1

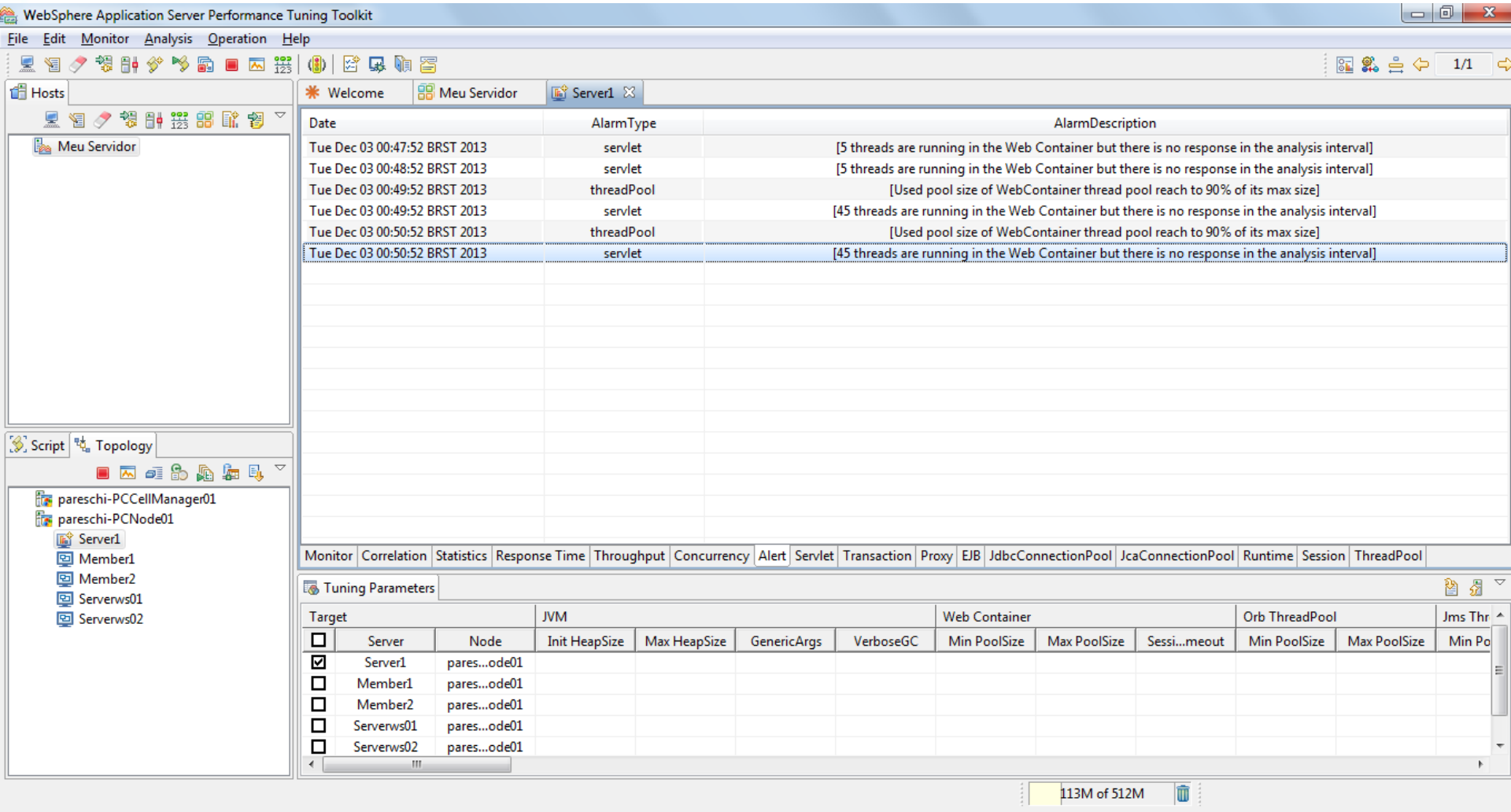


Clicando em alertas no slide anterior vamos para os gráficos de alerta, onde contemplam diversos itens tais como: Servlets, Transaction, ThreadPool, JDBC Pool e etc

No exemplo a seguir clicaremos no icone referente a ThreadPool

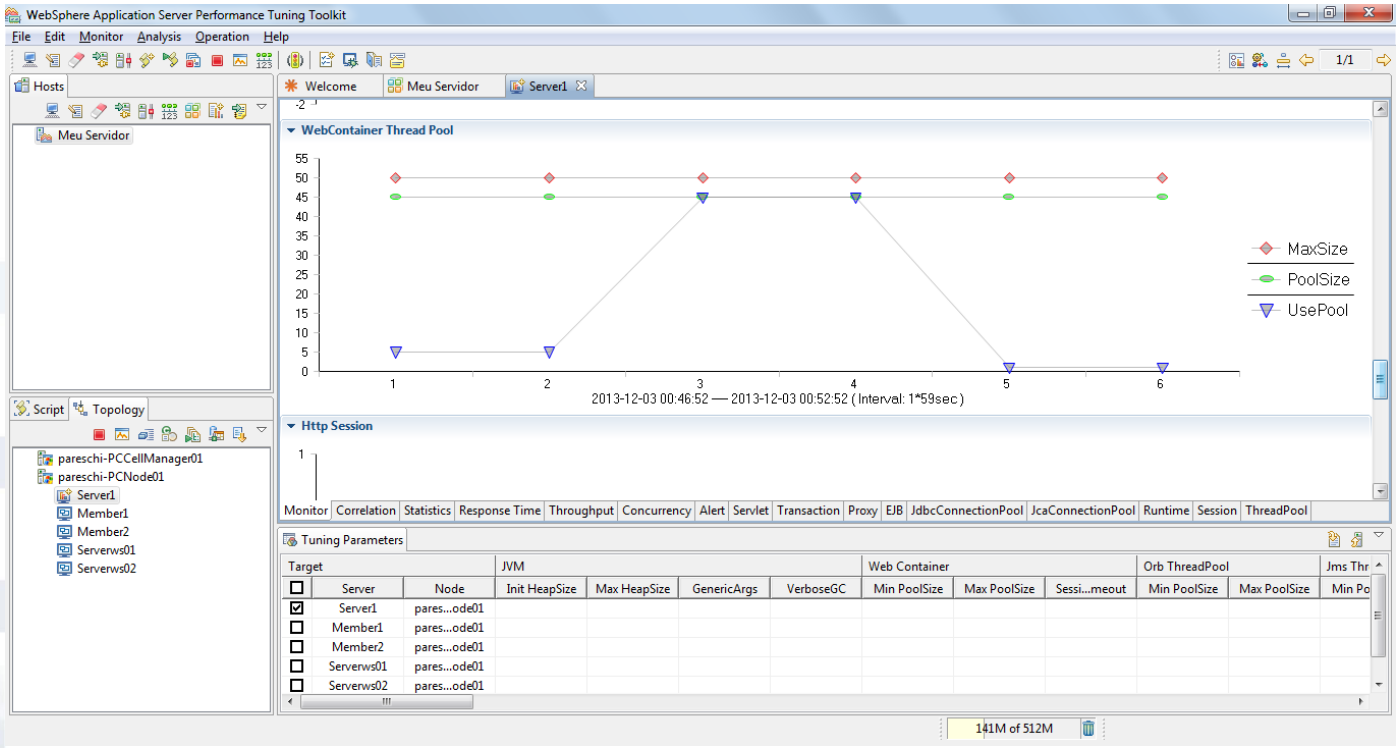
Análise Runtime

IWPTT – Alertas 2



Análise Runtime

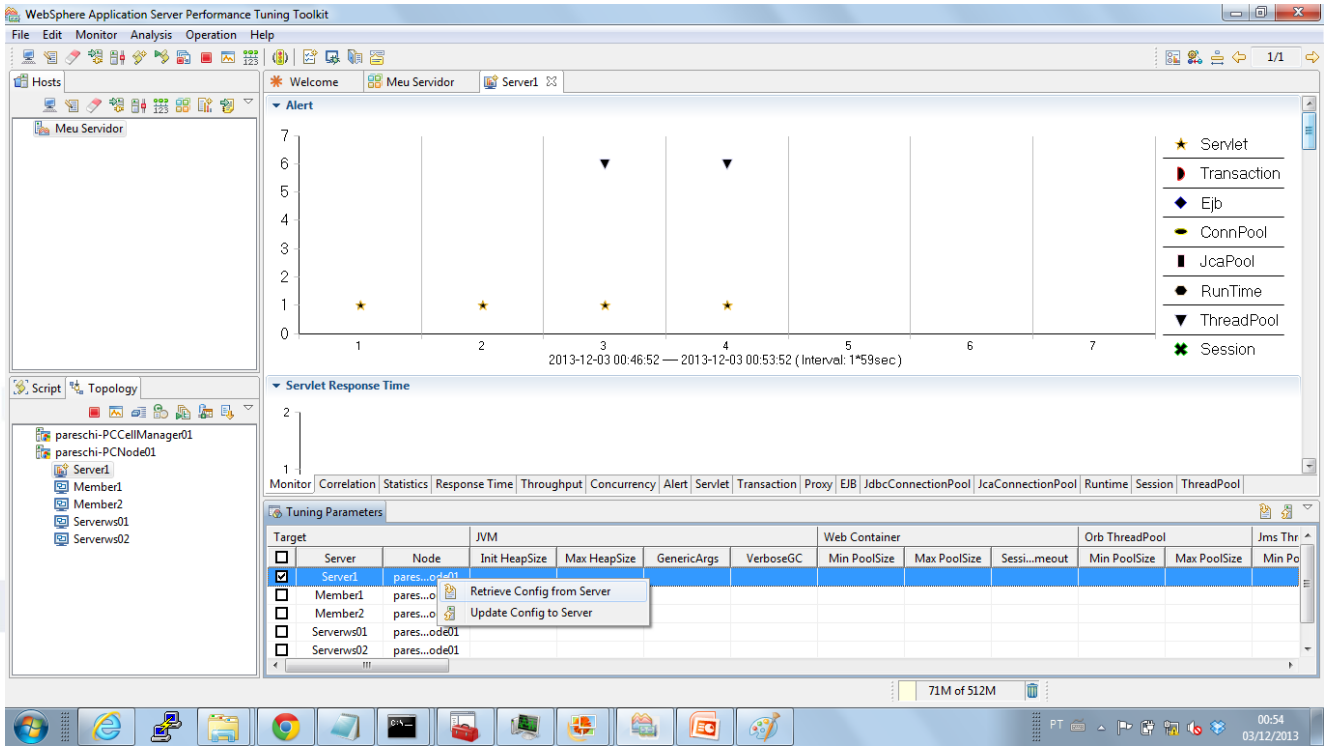
IWPTT – Monitorações



A analise que a ferramenta disponibiliza é muito mais intuitiva do que no PMI. Existem também configurações de alertas, regras entre outros que facilita a monitoração do ambiente.

Análise Runtime

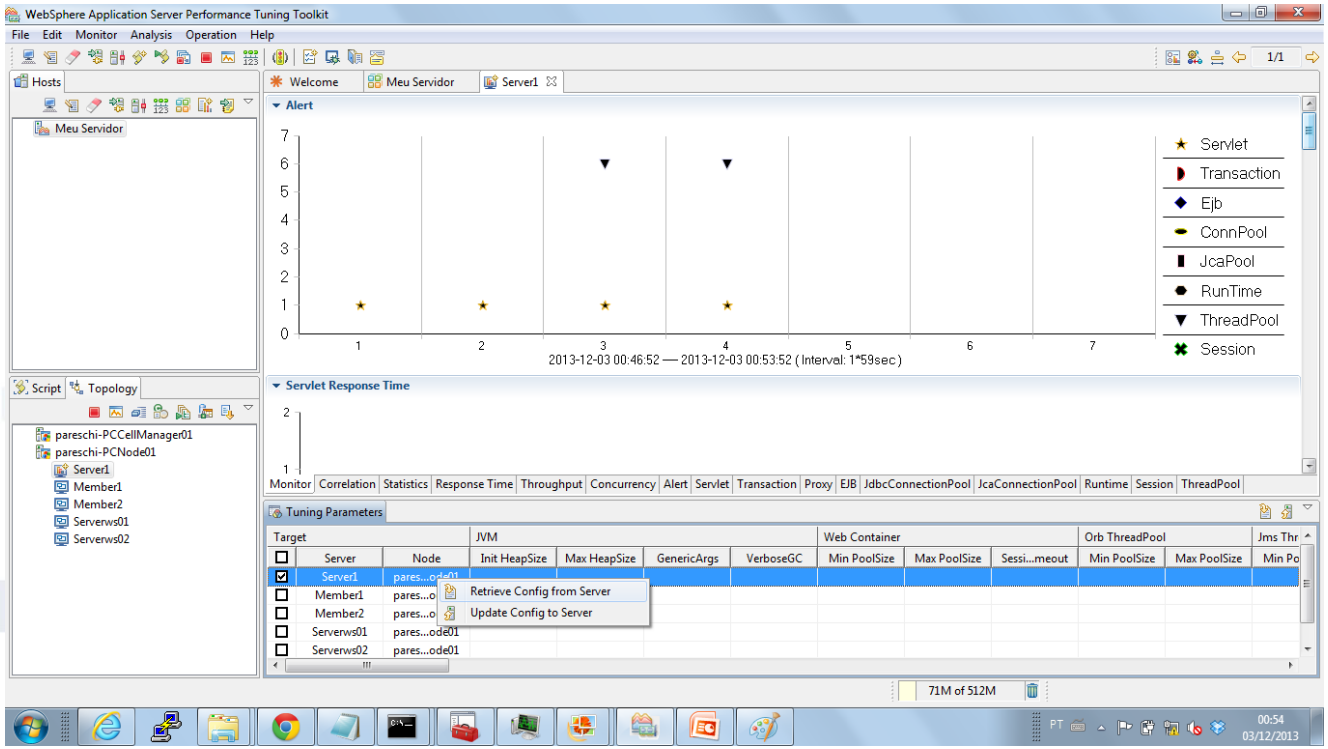
IWPTT – Retrieve de configurações



Com a ferramenta também podemos trazer todas as configurações do servidor e atualiza-las diretamente via ferramenta sem a necessidade de abrir o console adminsitrativo.

Análise Runtime

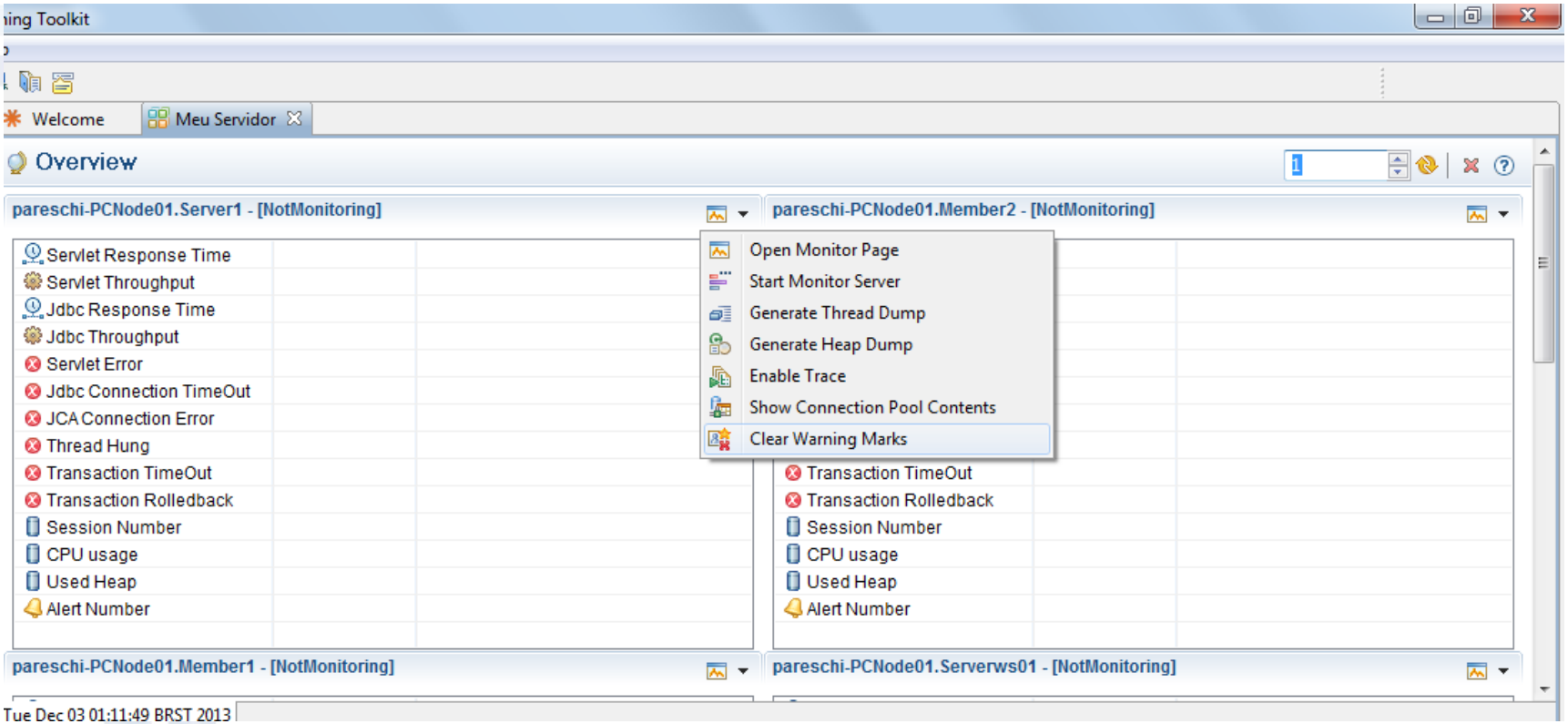
IWPTT – Atualização de configurações



Da mesma forma que executamos o retrieve, podemos alterar qualquer valor e executar um update da configuração, tal configuração será efetiva somente após o restart da JVM

Análise Runtime

IWPTT – Atualização de configurações



Com a ferramenta podemos gerar de forma rápida e sem uso de scripts os heapdumps e threaddump usados nas ferramentas de monitoração da IBM que vimos anteriormente

Melhores práticas

Desenvolvimento/Infraestrutura

MAGNASISTEMAS

- **Segregar conteúdo estático no webserver e a parte dinâmica no Application Server**
- **Não utilizar método finalizer() na aplicação, isso aumenta o tempo dos GCs na JVM**
- **Separar aplicações em clusters diferentes para utilizar os recursos de forma mais eficiente**
- **Logo após que criar o application server, setar algum parâmetro básico de tamanho de heap**
- **Caso não necessite monitorar desabilitar todos os PMI´s chega a 6% de degradação**
- **Se possível deixar o DMGR em um servidor excluiso**
- **Na arquitetura de ambientes sempre definir o conceito de three tier(Web -> WAS -> Banco)**
- **Sempre realizar um tuning básico nos ambientes(dev, hml e produção)**

MAGNASISTEMAS

Obrigado

CONTATO

Luiz Pareschi
Coordenador de Middleware
lpareschi@gmail.com