

MONITORING APPLICATION PERFORMANCE

with AppDynamics

FABIAN AND CODECENTRIC





**DEVOPS
DELIVERS
VALUE FAST**

**BUT DEVOPS
NEEDS MONITORING**





**MANY MORE THINGS TO
MANAGE**



FREQUENT CHANGE

SCALE ON DEMAND

A perspective view of a server room with rows of blue server racks on both sides. The floor is a light-colored grid that reflects the overhead lights. The ceiling has several horizontal light strips. In the distance, two double doors are visible at the end of the aisle. The overall atmosphere is high-tech and clean.

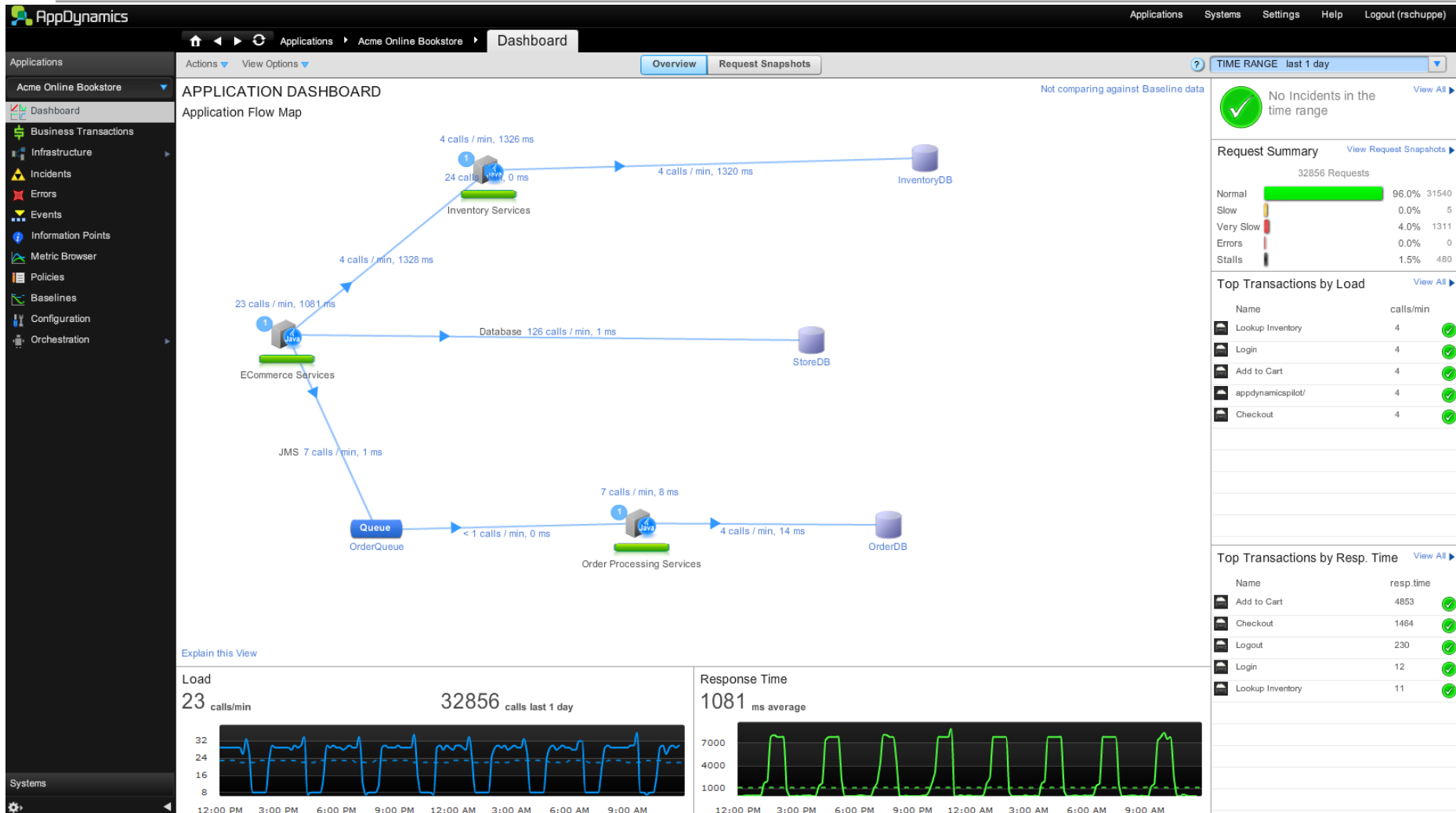


THE SOLUTION



AUTOMATIC DISCOVERY AND CONFIGURATION

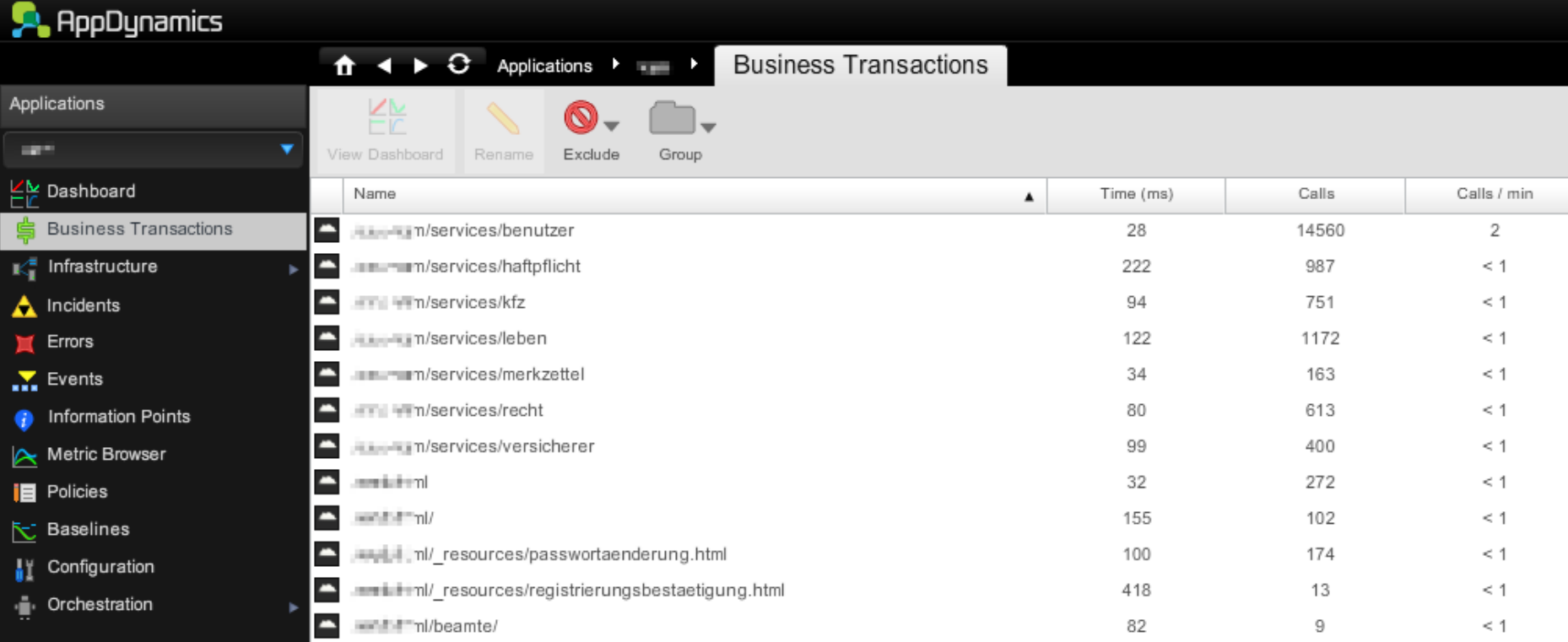
DISCOVER THE INTERNALS OF YOUR APPLICATION





TROUBLESHOOTING – SPOT ON

ALL YOUR TRANSACTIONS AT A GLANCE



The screenshot displays the AppDynamics interface for Business Transactions. The left sidebar contains navigation links for Applications, Dashboard, Business Transactions (selected), Infrastructure, Incidents, Errors, Events, Information Points, Metric Browser, Policies, Baselines, Configuration, and Orchestration. The top navigation bar includes a home icon, back/forward arrows, a refresh icon, and a breadcrumb trail: Applications > Business Transactions. Below the breadcrumb, there are action buttons: View Dashboard, Rename, Exclude, and Group. The main content area is a table listing various transactions.

Name	Time (ms)	Calls	Calls / min
./api/v1/services/benutzer	28	14560	2
./api/v1/services/haftpflicht	222	987	< 1
./api/v1/services/kfz	94	751	< 1
./api/v1/services/leben	122	1172	< 1
./api/v1/services/merkzettel	34	163	< 1
./api/v1/services/recht	80	613	< 1
./api/v1/services/versicherer	99	400	< 1
./api/v1/	32	272	< 1
./api/v1/_resources/passwortaenderung.html	155	102	< 1
./api/v1/_resources/passwortaenderung.html	100	174	< 1
./api/v1/_resources/registrierungsbestaetigung.html	418	13	< 1
./api/v1/beamte/	82	9	< 1

SPOT ON ANSWER

Request Snapshot Viewer - [http://www.services/haftpflicht](#)

Slow Request

Has Call Graph

REQUEST EXECUTION TIME
823 ms

REQUEST TIMESTAMP
09/15/10 11:41:45 AM

Diagnostic Session after - [5] % were slow requests out of [133] for minute ending [Wed Sep 15 10:59:35 CEST 2010]

SUMMARY

PHYSICAL VIEW

CALL GRAPH

HOT SPOTS

SQL CALLS

HTTP PARAMS

COOKIES

USER DATA

JVM / HARDWARE

Call Graph Selector

Multiple Call Graphs are available for this Tier.

View Selected Call Graph

Type	Call Originating From	Summary	Details	Time Taken	Start Time
▼ >=	(end user)		More Details	823 ms	09/15/10 11:41:45 AM
▶ Web Service	jboss	Service = TarifrechnerServiceService, Operation = plausibilisieren	More Details	44 ms	09/15/10 11:41:45 AM
▶ Web Service	jboss	Service = TarifrechnerServiceService, Operation = rechnen	More Details	103 ms	09/15/10 11:41:45 AM
▶ Web Service	jboss	Service = TarifrechnerServiceService, Operation = plausibilisieren	More Details	168 ms	09/15/10 11:41:45 AM
▶ Web Service	jboss	Service = TarifrechnerServiceService, Operation = plausibilisieren	More Details	47 ms	09/15/10 11:41:46 AM
▶ Web Service	jboss	Service = TarifrechnerServiceService, Operation = plausibilisieren	More Details	50 ms	09/15/10 11:41:46 AM
▶ Web Service	jboss	Service = TarifrechnerServiceService, Operation = plausibilisieren	More Details	68 ms	09/15/10 11:41:46 AM
▶ Web Service	jboss	Service = TarifrechnerServiceService, Operation = plausibilisieren	More Details	69 ms	09/15/10 11:41:46 AM
▶ Web Service	jboss	Service = TarifrechnerServiceService, Operation = rechnen	More Details	159 ms	09/15/10 11:41:46 AM

GAIN DEEP VISIBILITY



INVESTIGATE WITH FULL CALL STACK VISIBILITY

Request Snapshot Viewer - **transfer-area/secure/loeschen.do**

➔ Error Request

REQUEST EXECUTION TIME
30087 ms

REQUEST TIMESTAMP
09/13/10 02:36:49 PM

Request took more than slow request static threshold of [300]

SUMMARY

PHYSICAL VIEW

CALL GRAPH

HOT SPOTS

SQL CALLS

HTTP PARAMS

COOKIES

USER DATA

ERROR DETAILS

JVM / HARDWARE

Execution Time: **30087 ms**. Node **testsrv.codecentric.local-opt/tomcat/appdynamics/agent** Timestamp: **09/13/10 02:36:49 PM**.

[Callgraph navigation help](#) [This Call Graph was collected after a 605 ms delay](#)

Show Filters

Name	Time (ms)	Ext...
▼ Proxy For Spring Bean - springSecurityFilterChain:doFilter:390	0 ms (self) 0 %	
▼ Spring Bean - filterSecurity:doFilter:83	0 ms (self) 0 %	
▼ Spring Bean - filterSecurity:invoke:109	0 ms (self) 0 %	
▼ Proxy For Spring Bean - springSecurityFilterChain:doFilter:378	0 ms (self) 0 %	
▼ HTTPServlet:service:803	0 ms (self) 0 %	
▼ HTTPServlet:service:710	0 ms (self) 0 %	
▼ Servlet - InternalDispatcher:doService:807	0 ms (self) 0 %	
▼ Servlet - InternalDispatcher:doDispatch:875	0 ms (self) 0 %	
▼ Spring Bean - org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter:handle:313	0 ms (self) 0 %	
▼ Spring Bean - org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter:invokeHandlerMethod:326	0 ms (self) 0 %	
▼ Spring Bean - dateiLoeschenController:dateiLoschen:64	0 ms (self) 0 %	
▼ Spring Bean - dateiOperationService:dateiLoeschenKunde:160	0 ms (self) 0 %	
▼ Spring Bean - dateiOperationService:dateiLoschenUndArchivieren:496	0 ms (self) 0 %	
▼ Spring Bean - dateiKundeDAO:loescheDatei:395	0 ms (self) 0 %	
▼ Spring Bean - transactionTemplate:execute:128	0 ms (self) 0 %	
▼ Proxy For Spring Bean - dateiKundeDAO:doInTransaction:398	0 ms (self) 0 %	
▼ Spring Bean - simpleJdbcTemplate:update:237	0 ms (self) 0 %	
▼ Spring Bean - jsoJdbcTemplate:update:858	0 ms (self) 0 %	
▼ Spring Bean - jsoJdbcTemplate:update:850	0 ms (self) 0 %	
▼ Spring Bean - jsoJdbcTemplate:update:792	0 ms (self) 0 %	
▼ Spring Bean - jsoJdbcTemplate:execute:591	0 ms (self) 0 %	
▼ Proxy For Spring Bean - jsoJdbcTemplate:doInPreparedStatement:798	30081 ms (self) 100 % JDBC	

(close)

SEE ALL EXPENSIVE METHODS AND SQL QUERIES

Expensive Methods and SQL Statements

Most Expensive Methods / SQL Statements

Display up to the top 25 most expensive methods and SQL statements from the following snapshots:

	Time	Exe Time (ms)	Summary	Business Transaction	Node	* ...
▶▶	09/15/10 01:41:12 PM	10064	Response Time > moving avg for last 120 minutes:1335 + deviation range :1687 ms (4.0x of standard deviation)	ViewCart.sendItems	ec-prod-01	
▶▶	09/15/10 01:27:05 PM	9003	Response Time > moving avg for last 120 minutes:1328 + deviation range :1787 ms (4.0x of standard deviation)	ViewCart.sendItems	ec-prod-01	
▶▶	09/15/10 01:16:56 PM	7925	Response Time > moving avg for last 120 minutes:1141 + deviation range :1593 ms (4.0x of standard deviation)	ViewCart.sendItems	ec-prod-01	
▶▶	09/15/10 01:02:19 PM	14549	Response Time > moving avg for last 120 minutes:1099 + deviation range :1506 ms (4.0x of standard deviation)	ViewCart.sendItems	ec-prod-01	

Expensive Methods (28)

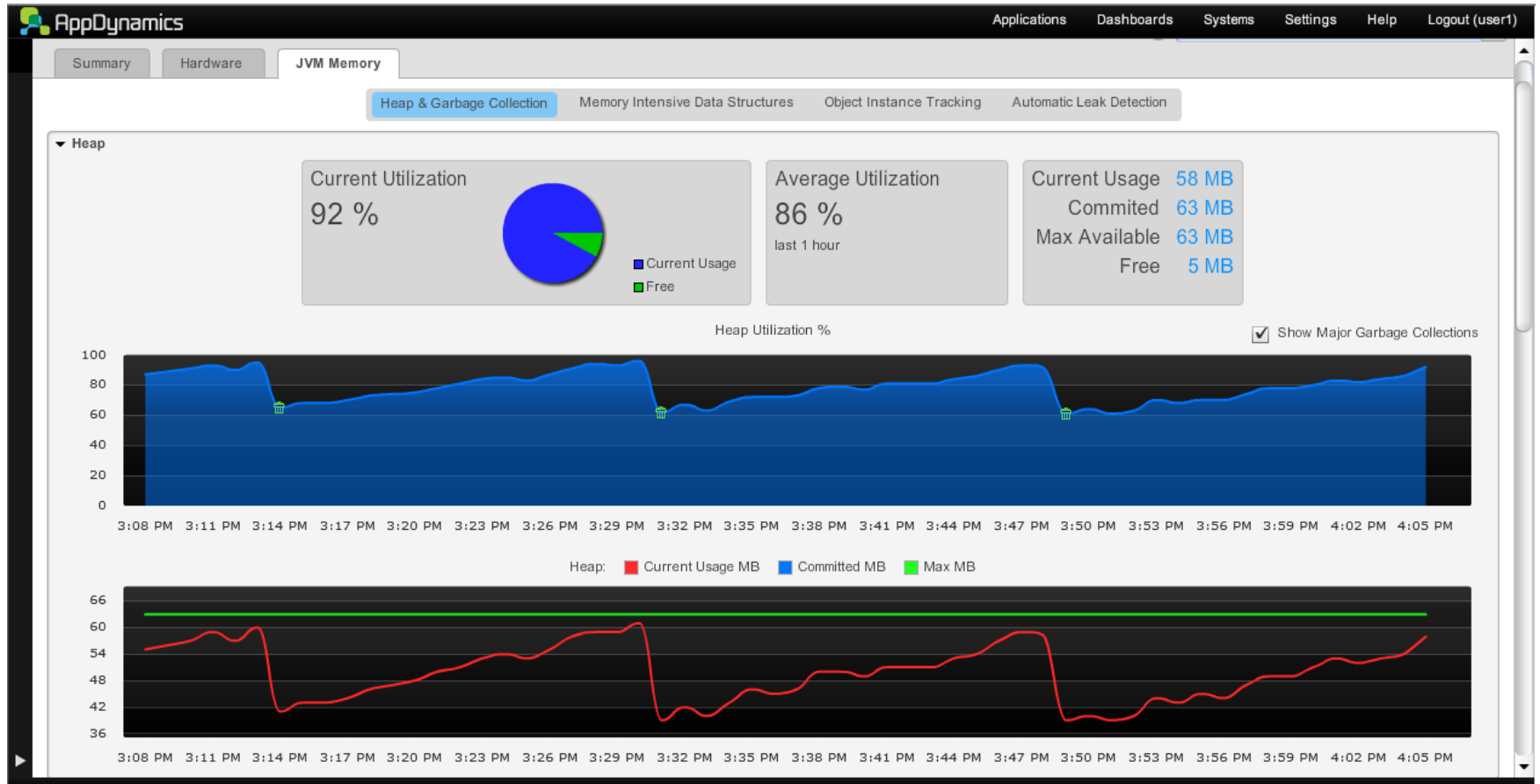
Expensive SQL (47)

The following methods were the most expensive methods invoked by the selected snapshots.

Class and Method	Total Exe. Time(ms)	Avg. Exe. Time(ms)	Call Count
StrutsActionProxy:execute	40488	10122.0	4
Struts Action - ViewCart.sendItems:124	40431	10107.8	4
Spring Bean - dataSource.getConnectionFromDriverManager:255	39023	1300.8	30
Proxy For Spring Bean - cartServiceTarget:invoke	26391	1256.7	21
Spring Bean - proucerJmsTemplate:send:469	18556	1159.8	16
Spring Bean - dataSource.getConnection:236	13110	1311.0	10
Spring Bean - cartPersistence.getCartSize:41	12958	1619.8	8
Proxy For Spring Bean - cartServiceTarget:checkOut	12764	1160.4	11
Spring Bean - cartServiceTarget:checkOut:40	10598	1059.8	10
Proxy For Spring Bean - proucerJmsTemplate:doInJms:479	9278	1159.8	8
Spring Bean - messageProducer.sendMessageWithOrderId:30	9131	2282.8	4
com.appdynamics.inventory.OrderServiceSOAP11BindingStub:createOrder:158	7066	706.6	10

(close)

MONITOR JVM HEAP SIZE



**ALL WITHOUT
EFFORT**





FREE VERSION AVAILABLE

WANT TO LEARN MORE?

come to the codecentric booth