

Monitoring von Java Webanwendungen

Peter Roßbach
pr@objektpark.de



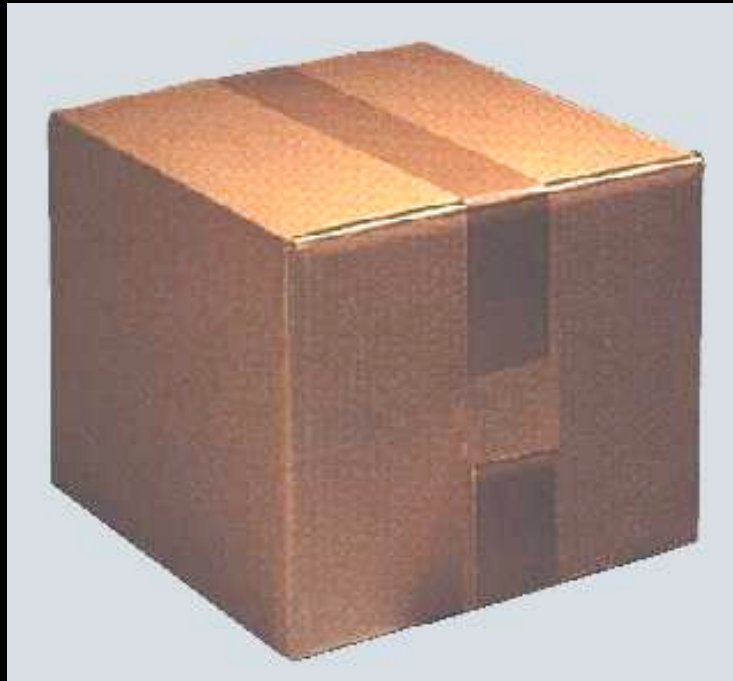
Peters Rucksack

- Peter Roßbach
 - Freiberuflicher IT-Systemarchitekt, Berater, Trainer und Autor
 - Entwickler im Apache Tomcat Projekt
 - Mitglied der Apache Software Foundation
 - Webexperte
 - Autor der TomC@Kolumne im Java Magazin
 - Autor
 - <http://tomcat.objektpark.org/>
 - Tomcat 4x <http://www.tom4.de>
 - Java Server und Servlets
 - E-Mail: pr@objektpark.de
 - Fachautor und Speaker auf deutschen Java Konferenzen

*Java*magazin













Warum Monitoring?

- Wie gut funktioniert mein System eigentlich?
- Schnell Probleme & Ursache erkennen
 - Benachrichtigung
- Zustandsanalyse
 - Post Mortem
 - Kapazitätsplanung

Was ist “Monitoring”?

- Information über ein System bereitstellen
 - Messwerte
 - Konfiguration
 - Daten (Logs, Schwellwerte, usw.)
- Reaktionsregeln anwenden
- Informationen auswerten und visualisieren

Das perfekte Monitoring-Tool

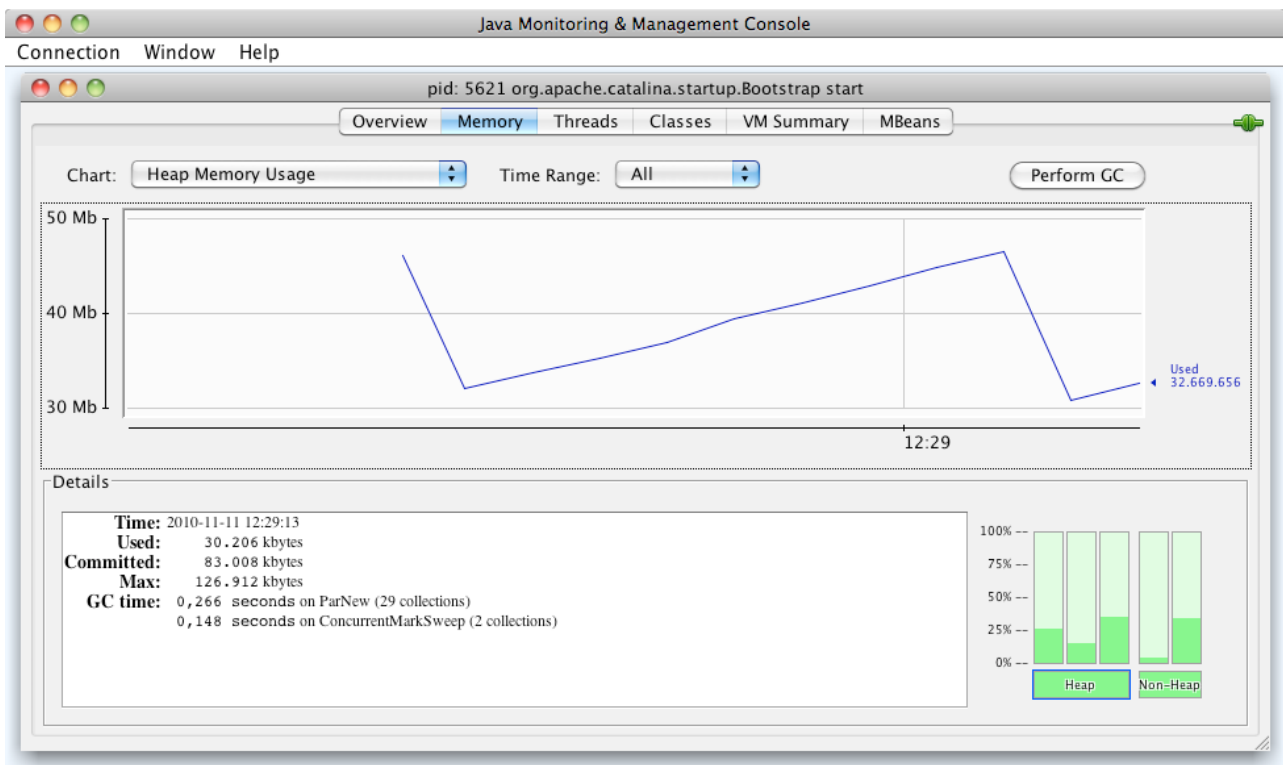
- sofort produktiv
 - einfach zu installieren
- nicht störend
- konfigurierbar
- erweiterbar
- skalierbar
- Ergonomisch
- Reaktion ermöglichen
- Daten exportierbar

Erste Tools

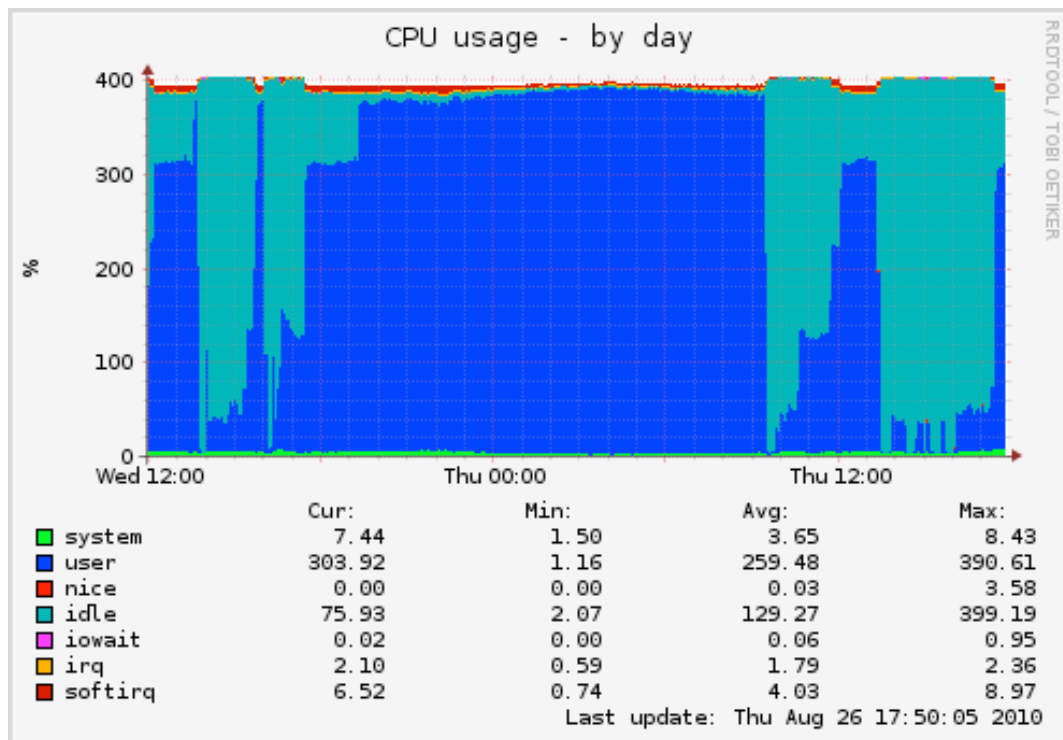
- kill -3
- jconsole
- jvisualvm



jconsole

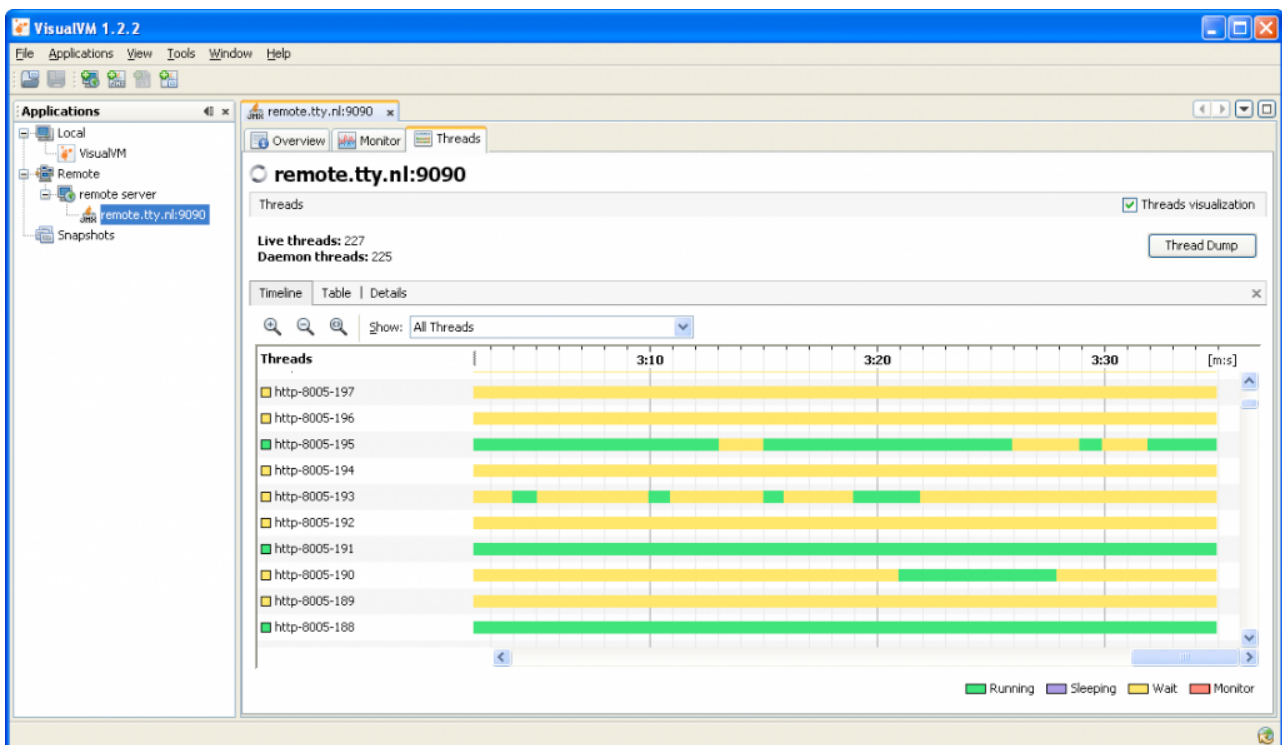


Situation



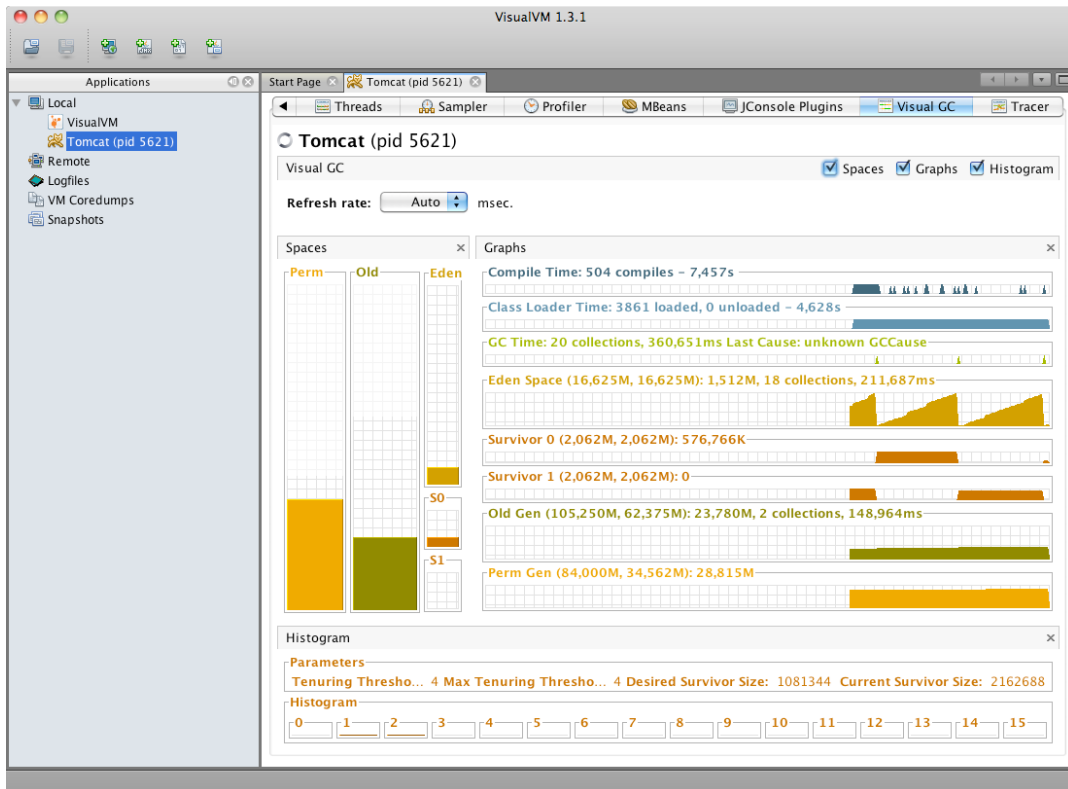
Aus <http://blog.tty.nl/2010/09/03/using-visualvm-to-fix-live-tomcat-and-jvm-problems/>

Visualvm Thread view



Aus <http://blog.tty.nl/2010/09/03/using-visualvm-to-fix-live-tomcat-and-jvm-problems/>

VisualVM:Visual GC

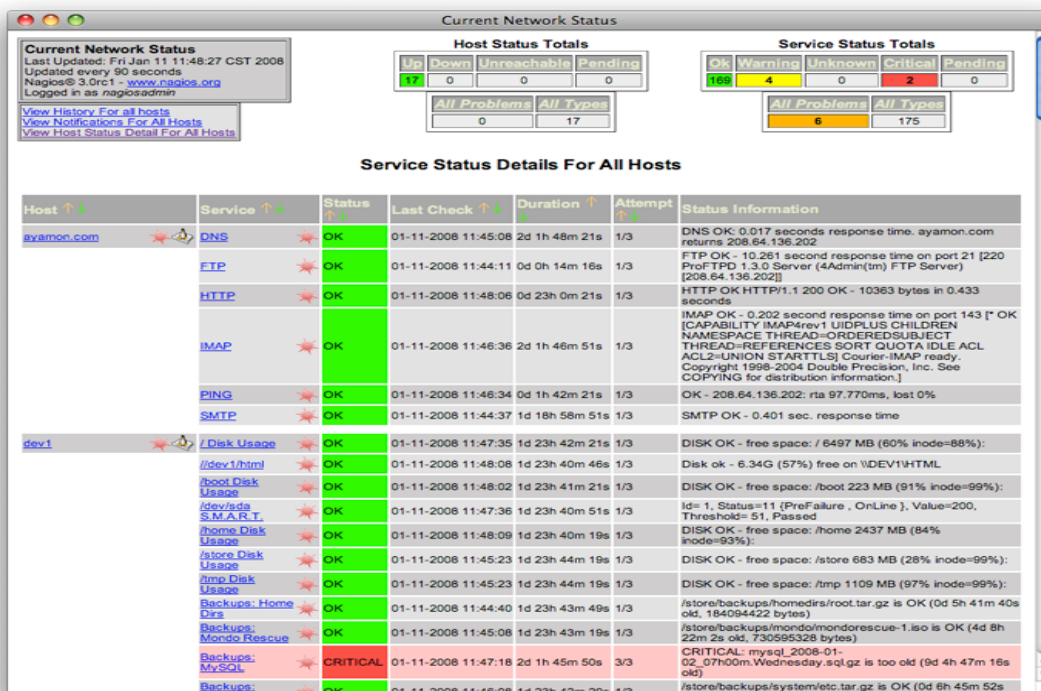


System Monitoring Tools

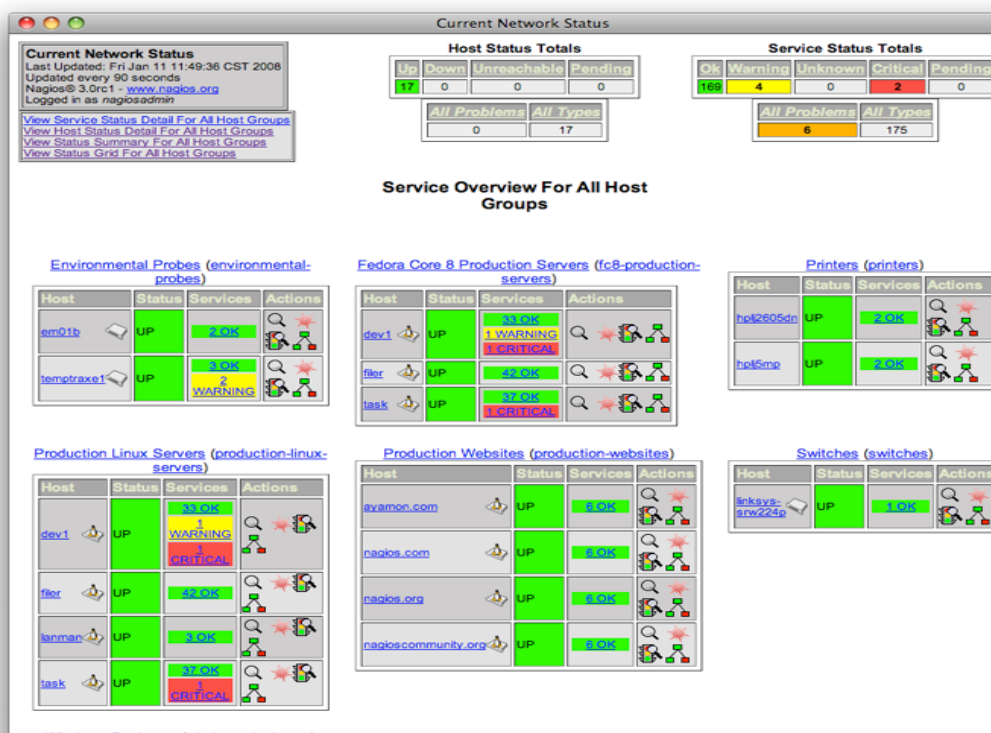
- Munin
- Nagios
- Cacti
- Zabbix
- OpenNMS
- ...



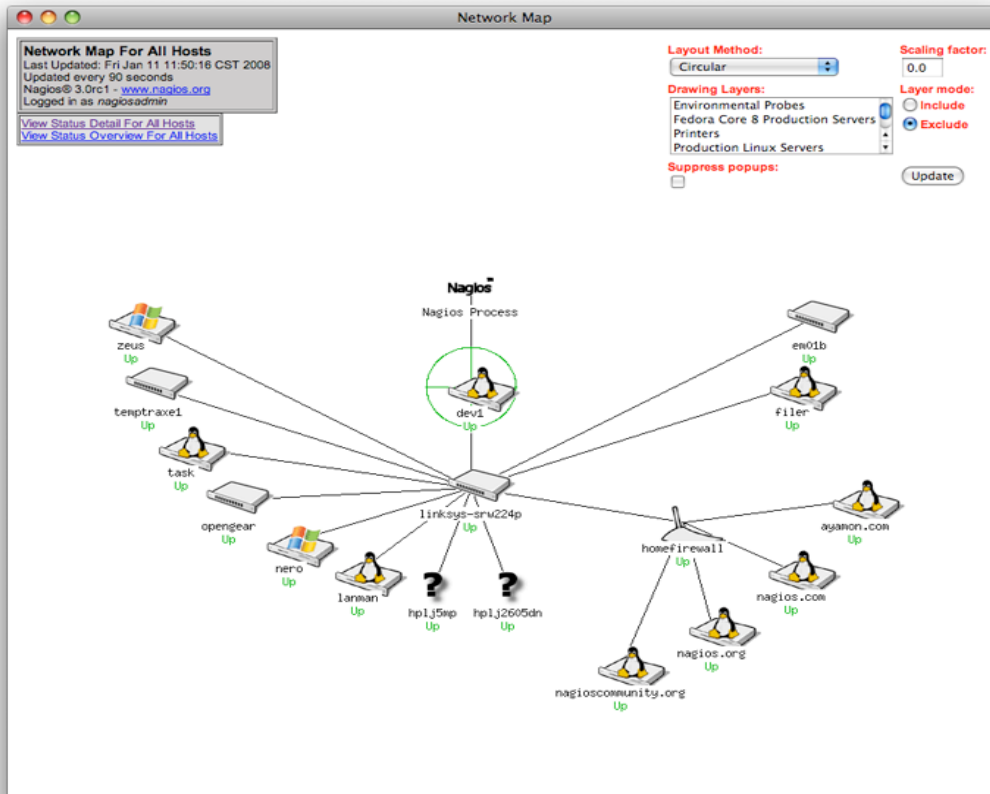
Nagios



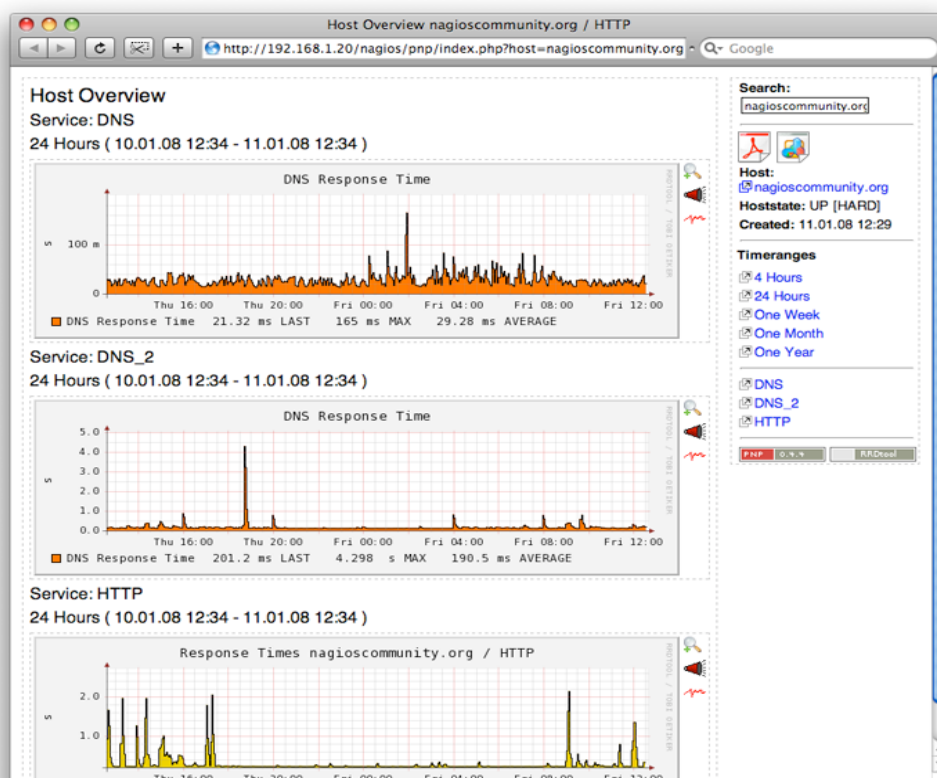
Nagios



Nagios



Nagios



Nagios & JMX

- check_jmx
 - http://www.monitoringexchange.org/inventory/Check-Plugins/Software/Java/check_jmx
 - <http://code.google.com/p/jmxquery/>
 - http://exchange.nagios.org/directory/Plugins/Java-Applications-and-Servers/check_jmx/details
- Jsend_nsca
 - <http://code.google.com/p/jsendnsca/>
- NetwaysGrapherV2
 - http://www.netways.de/de/produkte/nagios_addons/netwaysgrapherv2/
- Jolokia
 - <http://www.jolokia.org/>
- Tomcat JMXServlet

Entwickler Tools

- JMX per Hand
- Perf4J
- JETM
- Epsilon
- JaMon
- Java Simon
- ...



```
public static void main(String[] args) {  
    Stopwatch stopwatch = SimonManager.getStopwatch("stopwatch");  
  
    for (int i = 0; i < 10; i++) {  
        strangeMethod();  
    }  
    System.out.println(stopwatch);  
    System.out.println(stopwatch.sample());  
}
```

```
Simon Stopwatch: [stopwatch INHERIT] total 9.54 s, counter 10, max 2.03 s, min 16.6 ms  
StopwatchSample{total=9.54 s, counter=10, min=16.6 ms, max=2.03 s, minTimestamp=090623-102015.866,  
maxTimestamp=090623-102012.967, active=0, maxActive=1, maxActiveTimestamp=090623-102015.866,  
last=1.76 s, mean=954 ms, standardDeviation=679 ms,  
variance=5.1276613691815206E17, varianceN=4.614895232263369E17, note=null}
```

Klassische Meßpunkte

- Interceptor
 - Spring / AOP / EJB3
- JDBC Treiber

```
jdbc:jamon:sybase:Tds:localhost:2638?  
jamonrealdriver=com.sybase.jdbc2.jdbc.SybDriver
```

- Logging (log4j appender)
- HTTP (requests / responses)

JaMon

JAMon Action	Mon Proxy Action	Output	Range/Units	Display Columns	Cell Format	Filter (optional)	Highlight (optional)	(optional)
Refresh	No Action (currently=TTTF)	HTML	AllMonitors	Basic Cols Only	#####	http		

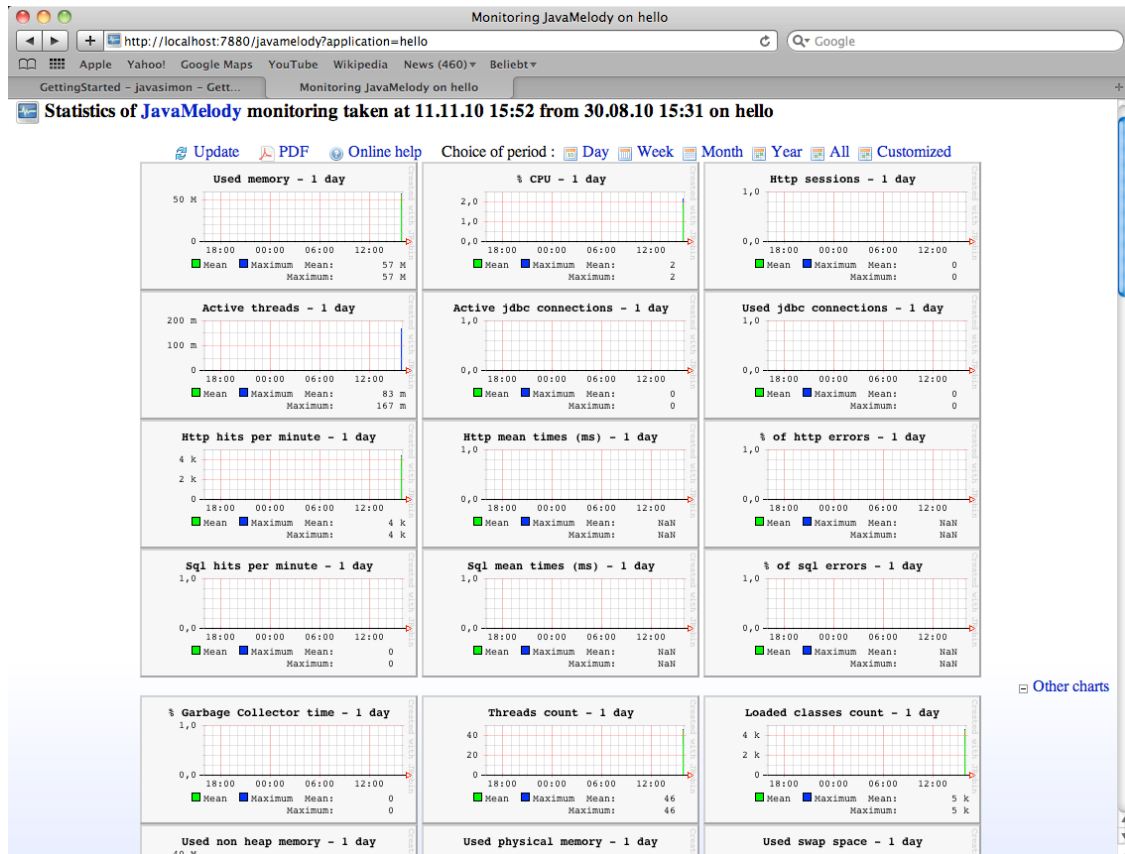
Modify	Label	Hits	Avg	Total	StdDev	LastValue	Min	Max	Active	AvgActive	MaxActive	FirstAccess	LastAccess	Enabled	Primary	HasListeners
☐	com.jamonapi.http.JAMonTomcatValve.request.allPages.ms.	24	42	998	139	15	0	672	1	1	1	9/18/07 9:19:52 PM	9/18/07 9:20:37 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.contextpath: .ms.	4	0	0	0	0	0	0	0	1	1	9/18/07 9:20:26 PM	9/18/07 9:20:26 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.contextpath: /jamon.ms.	14	55	765	178	31	0	672	1	1	1	9/18/07 9:19:52 PM	9/18/07 9:20:00 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.contextpath: /manager.ms.	6	39	233	62	15	0	156	0	1	1	9/18/07 9:20:28 PM	9/18/07 9:20:37 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /.ms.	1	0	0	0	0	0	0	0	1	1	9/18/07 9:20:26 PM	9/18/07 9:20:26 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /asf-logo-wide.gif.ms.	1	0	0	0	0	0	0	0	1	1	9/18/07 9:20:26 PM	9/18/07 9:20:26 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /jamon/css/JAMonStyles.css.ms.	2	0	0	0	0	0	0	0	1	1	9/18/07 9:19:52 PM	9/18/07 9:19:53 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /jamon/images/dsco.gif.ms.	2	0	0	0	0	0	0	0	1	1	9/18/07 9:19:52 PM	9/18/07 9:19:54 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /jamon/images/fds_logo_small.jpg.ms.	2	0	0	0	0	0	0	0	1	1	9/18/07 9:19:52 PM	9/18/07 9:19:54 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /jamon/images/jamon_small.jpg.ms.	2	0	0	0	0	0	0	0	1	1	9/18/07 9:19:52 PM	9/18/07 9:19:54 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /jamon/images/monmodify.gif.ms.	2	0	0	0	0	0	0	0	1	1	9/18/07 9:19:52 PM	9/18/07 9:19:54 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /jamon/jamonadmin.jsp.ms.	4	191	765	320	31	31	672	1	1	1	9/18/07 9:19:52 PM	9/18/07 9:20:00 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /manager/html.ms.	2	109	218	66	62	62	156	0	1	1	9/18/07 9:20:28 PM	9/18/07 9:20:30 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /manager/html/list.ms.	1	0	0	0	0	0	0	0	1	1	9/18/07 9:20:34 PM	9/18/07 9:20:34 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /manager/images/asf-logo.gif.ms.	1	0	0	0	0	0	0	0	1	1	9/18/07 9:20:30 PM	9/18/07 9:20:30 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /manager/images/tomcat.gif.ms.	1	0	0	0	0	0	0	0	1	1	9/18/07 9:20:30 PM	9/18/07 9:20:30 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /manager/status.ms.	1	15	15	0	15	15	15	0	1	1	9/18/07 9:20:37 PM	9/18/07 9:20:37 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /tomcat-power.gif.ms.	1	0	0	0	0	0	0	0	1	1	9/18/07 9:20:28 PM	9/18/07 9:20:28 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.request.page.value: /tomcat.gif.ms.	1	0	0	0	0	0	0	0	1	1	9/18/07 9:20:26 PM	9/18/07 9:20:26 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.response.getCount().bytes	24	4,796	115,110	5,778	5,446	0	15,118	0	0	0	9/18/07 9:19:52 PM	9/18/07 9:20:37 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.response.getStatus().value: 200, httpStatus	13	200	2,600	0	200	200	200	0	0	0	9/18/07 9:19:52 PM	9/18/07 9:20:37 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.response.getStatus().value: 304, httpStatus	10	304	3,040	0	304	304	304	0	0	0	9/18/07 9:19:52 PM	9/18/07 9:19:54 PM	true	false	false
☐	com.jamonapi.http.JAMonTomcatValve.response.getStatus().value: 401, httpStatus	1	401	401	0	401	401	401	0	0	0	9/18/07 9:20:28 PM	9/18/07 9:20:28 PM	true	false	false

OS Java EE monitoring tools

- Lambda-Probe / Psi-Probe
- InfraRed
- IxoraRMS
- JavaMelody
- Java Monitor
- ...



JavaMelody



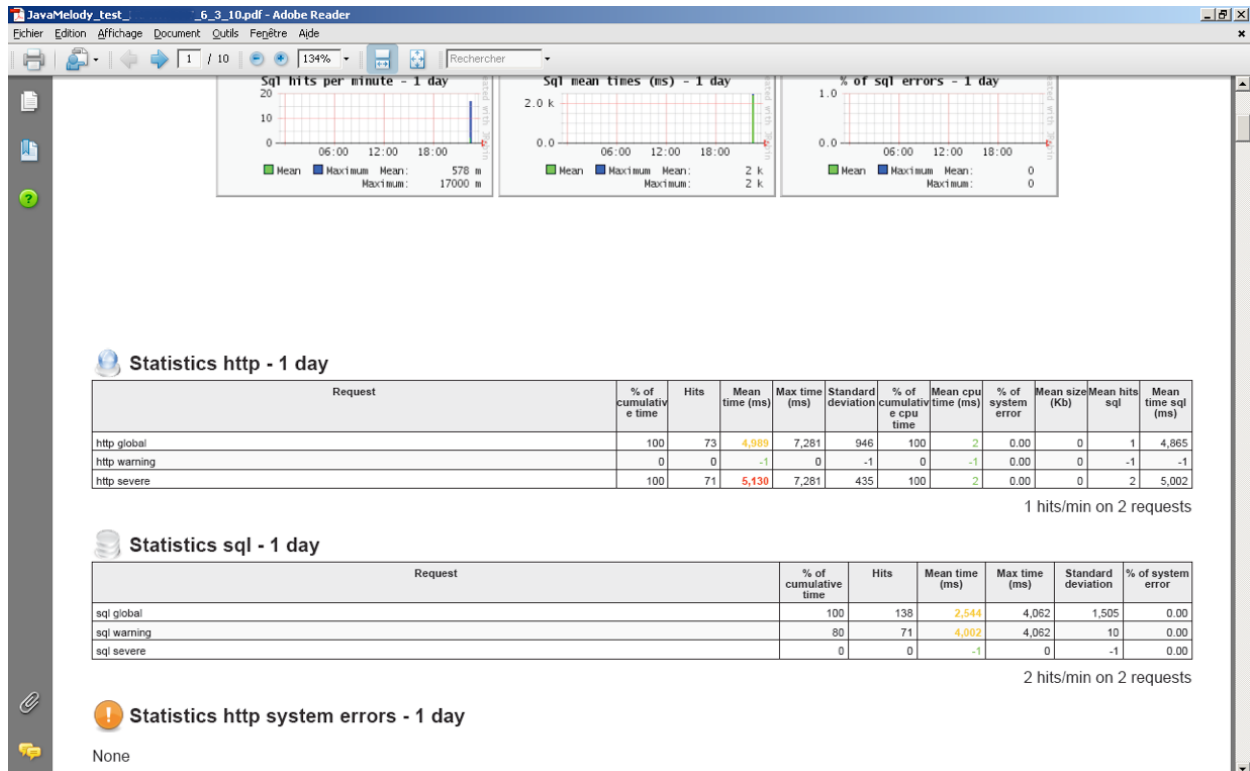
JaMon SQL??

Retour Actualiser Temps en ms des requêtes les plus longues

Base de données : Temps en ms des requêtes les plus longues

PCT NUMBER(22)	ELAPSED_TIME NUMBER(22)	MS BY EXEC NUMBER(22)	CPU TIME NUMBER(22)	TOTAL_COST NUMBER(22)	ELEM_COST NUMBER(22)	EXEC NUMBER(22)	NB_ROWS NUMBER(22)	SQL_TEXT VARCHAR2(1000)
6.75	3.15	70	.12	440	10	45	140	select /*+ index(idl_ub1\$ i_idl_ub11) */ piece#,length,piece from idl_ub1\$ where obj# =
4.71	2.2	2200	.16	1809	1809	1	0	call dbms_scheduler.auto_purge()
4	1.87	1869	.09	1403	1403	1	1	DECLARE job BINARY_INTEGER := job; next_date DATE := mydate; broken BOOLE/
3.67	1.71	156	.03	399	36	11	11	declare vsn varchar2(20); begin vsn := dbms_rcvman.getPackageVersion; pkg_vsn.pkg

JavaMelody Day Stats



Jolokia

<http://www.jolokia.org>

Einfacher HTTP Restful Zugriff statt JSR 160

Zugriff auf die JMX Daten via

Java

Perl/Shell

JavaScript

Proxy Support

Deployment einer WebApp, JVM6 Agent, Mule und OSGI Bundle

Operationen

read, write, exec, list, search, version

Sicherheit (WEB-INF/classes/jolokia-access.xml)

<deny> or <allow>

<mbean>

<name>java.lang:type=Memory</name>

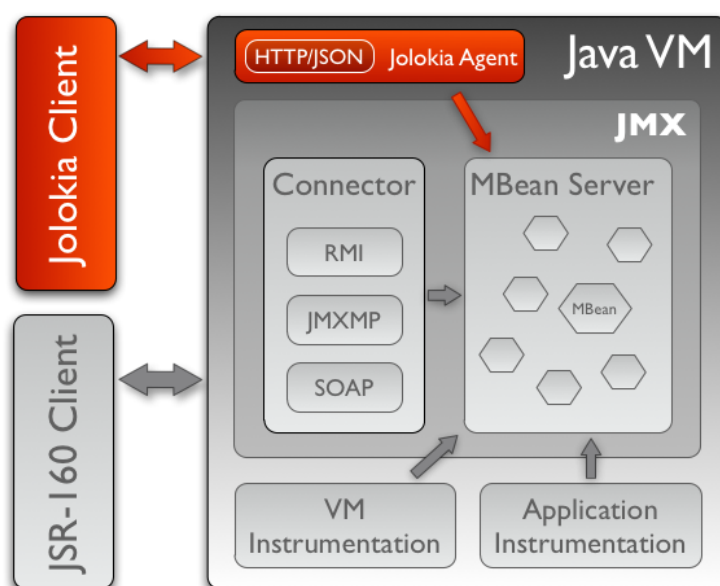
<attribute>*Memory*</attribute>

<attribute mode="read">Verbose</attribute>

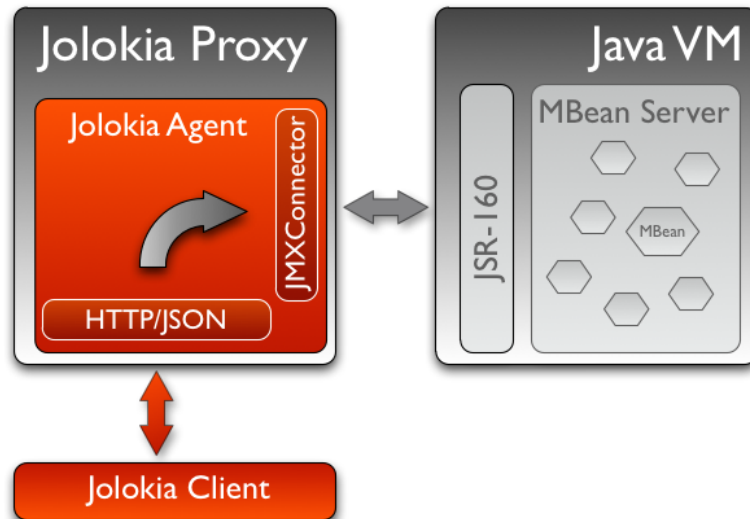
<operation>gc</operation>

</mbean>

Jolokia



Jolokia Proxy



Commercial Tools

- RHQ (Jboss On OpenSource)
- Hyperic HQ
- dynaTrace
- TCat Server
- AppDynamics
- Introscope
- NewRelic RPM
- ...



Spaßbremse?



- “low overhead”?



Overhead

- Negativen Einfluss auf:
 - CPU
 - Speicher
 - I/O
- Seiteneffekte
 - Timing verändert sich stark
 - Versteckte Synchronisation

Fangfrage...

**Wer will nun in sein
Monitoringsystem
investieren?
(Know - How, Geld)**

Fragen zur Lösung

- Vorgefertigt oder umfangreich zu konfigurieren?
- Welche Daten sollen gehalten werden?
 - Volume / Granularität
- Standard Werte und eigene Kennzahlen?
- Darf mein Code instrumentiert werden?
- Sind Erweiterungen möglich?
- Akzeptierbare Geschwindigkeitseinbußen?
- Software as a Service?
- Wer soll es einführen und verwenden?

Vielen Dank

- Kommen Sie zu unseren Sessions und Workshop
 - WJAX München 2011
 - Apache Tomcat 7
 - JEE Day
 - Persistence
 - Monitoring
 - Java EE Summit in Berlin Dezember 2011
 - OOP 2012
- Beratung, Schulung, Workshops und Tomcat Support



Peter Roßbach
pr@objektpark.de



Hilfe ist nah...



LinuxHotel.de -- Lernen in angenehmer Atmosphäre mit Experten

Schulungen für Administratoren und Entwickler

Alles im Bereich Linux, Netzwerk und OS Services

Groovy & Grails (Dierk König & Marc Guillemot)

Osgi (Gerd Wütherich)

Apache Tomcat (Peter Roßbach)

Apache httpd (Michael Wandel & Peter Roßbach)

Webcontainer Firewalls (Christian Schneider & Thomas Krautgartner)

Java für Architekten (Torsten Friebe)

bald mehr...



Links (1)

- Munin <http://munin-monitoring.org/>
- Cacti <http://www.cacti.net/>
- Nagios <http://www.nagios.org/>
- Zabbix <http://www.zabbix.com/>
- OpenNMS <http://www.opennms.org/>
- Perf4J <http://perf4j.codehaus.org/>
- JaMon <http://jamonapi.sourceforge.net/>
- Java Simon <http://code.google.com/p/javasimon/>
- JETM <http://jetm.void.fm>
- InfraRED <http://infrared.sourceforge.net/>
- Java Monitor <http://java-monitor.com>

Links (2)

- Psi Probe <http://code.google.com/p/psi-probe/>
- Lambda Probe <http://www.lambdaprobe.org>
- Hyperic HQ <http://www.hyperic.com>
- Epsilon <http://www.obix-labs.com/wiki/bin/view/Epsilon/>
- IxoraRMS <http://www.ixorarms.com>
- NewRelic RPM <http://www.newrelic.com/>
- JavaMelody <http://code.google.com/p/javamelody/>
- TCatServer <http://www.mulesoft.com/tcat-server-enterprise-tomcat-application-server>
- Blog Post “The Java Application Performance Management Vendor Showdown”<http://williamlouth.wordpress.com/2010/05/25/the-java-application-performance-management-vendor-showdown/>