

Application Performance Monitoring of a scalable Java web-application in a cloud infrastructure

Final Presentation
August 5, 2013

Student:	Michael Rose
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Alexander Schneider Dr. Thomas Büchner

Agenda

- Application Performance Monitoring
- APM for *Tricia*
- Evaluation
- Further Work
- Summary

Application Performance Monitoring (APM)

„Monitoring is the process of maintaining surveillance over the existence and magnitude of state change and data flow in a system.“¹

- How fast is my application?
- ... If the application is not fast ...
 - **When** is it slow? everytime, on a certain time of the day, ...
 - **What** is slow? everything, a certain type of operation, a single request, ...
 - **Why** is it slow? bug in the application, insufficient resources, wrong configuration ,...

➔ Solve the problem

¹ L. Slawek, *Effective monitoring and alerting*

APM - Stakeholders

- Customers
 - improve their business
 - **application must be available and usable**
- Developers
 - Are there any Problems?
 - If so:
 - **What are they?**
 - **How can they be solved?**
- Cloud Operations
 - Is there need for action? (more resources, more servers, ...)
 - require application-level information
 - **standardized interface**

APM – Key Metrics

- Overall response time
- Requests per Minute
- Database operation execution time / operations per Minute
- Elasticsearch operation execution time / operations per Minute
- System Load
- Process Load
- Memory Consumption

APM – Means of Evaluation

Checking for problems → Performance Snapshot

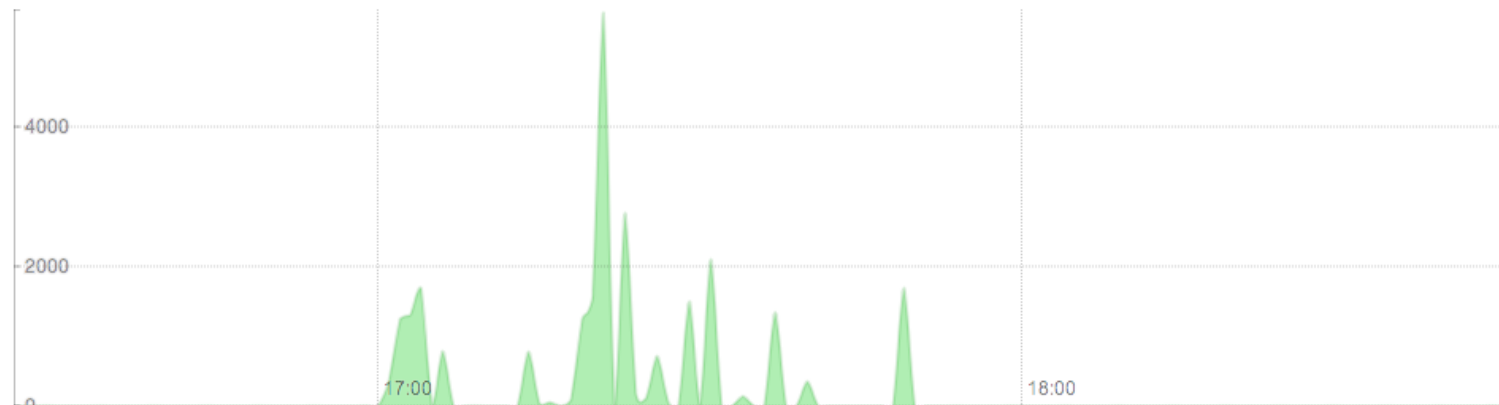
- created at a regular interval (e.g. 2 hours)
- detailed information for all monitors in the system
- use as indicators → maximum execution time

Name	Avg	Total	Max
Response Time	712.7 ms	177 920.4 ms	6 942.5 ms
Elasticsearch Op	230.0 ms	78 036.2 ms	5 391.2 ms
Database Op	0.9 ms	56 493.5 ms	1.2 ms

APM – Means of Evaluation

Verifying a Problem → Graphical Visualization

- create time series
- see if just a singular event
- detect dependencies



APM – Means of Evaluation

Tracking a Problem → Performance Trace

- created in case of a problem
- gather **all** monitors used in the executed operation
- print detailed information

→ Find the monitor responsible for the slowdown

APM – Means of Evaluation

Tracking a Problem → Performance Trace

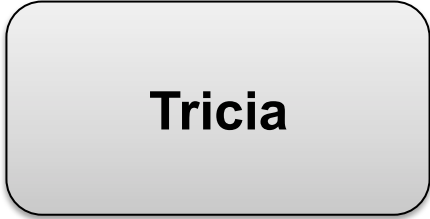
5 200 ms for /action/execute

Name	Hits	Total	Avg
responseTime	1	5 200.4 ms	5 200.4 ms
handler	1	5 000.0 ms	5 000.0 ms
utilities.parseUrl	5	4 505.0 ms	901.0 ms
es	10	350.7 ms	35.1 ms
db	22	26.2 ms	1.2 ms

APM for Tricia

- Existing Solutions / Libraries
 - New Relic
 - Kieker
 - Java Simon
 - RRDTTool
- Existing Standards
 - Java Management Extensions (JMX)
 - Nagios
 - collectd, ganglia, ...

APM for Tricia - Architecture



Tricia

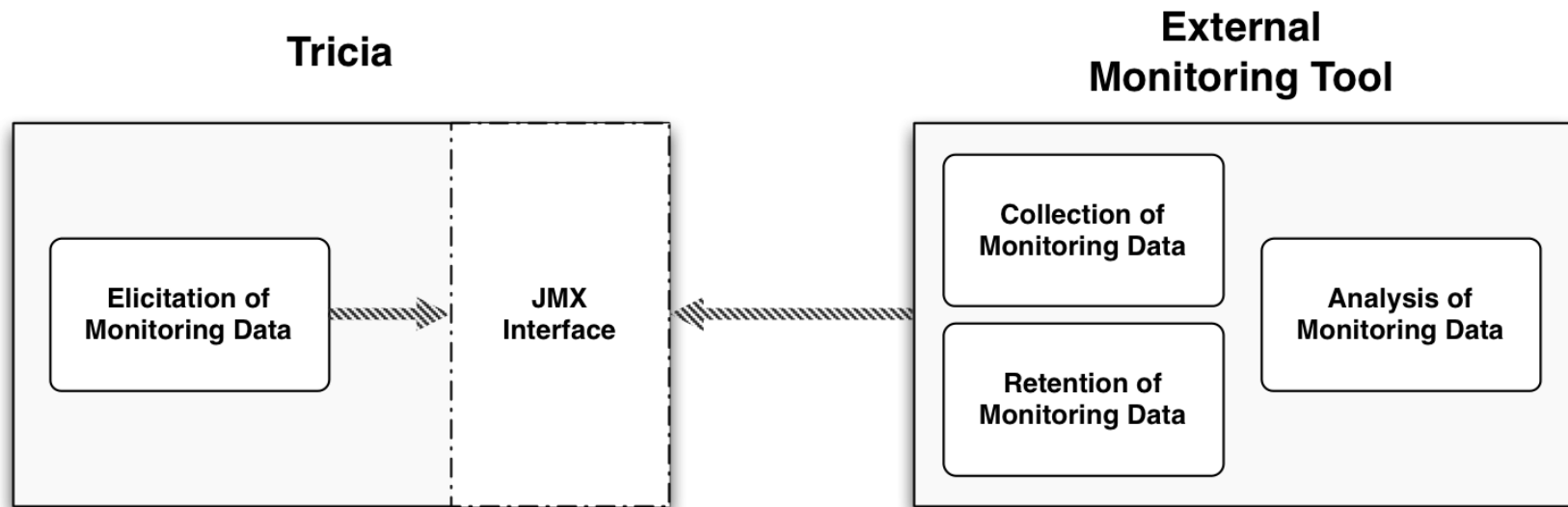
- Performance Snapshots
- Performance Traces
- Detection
- Tracking



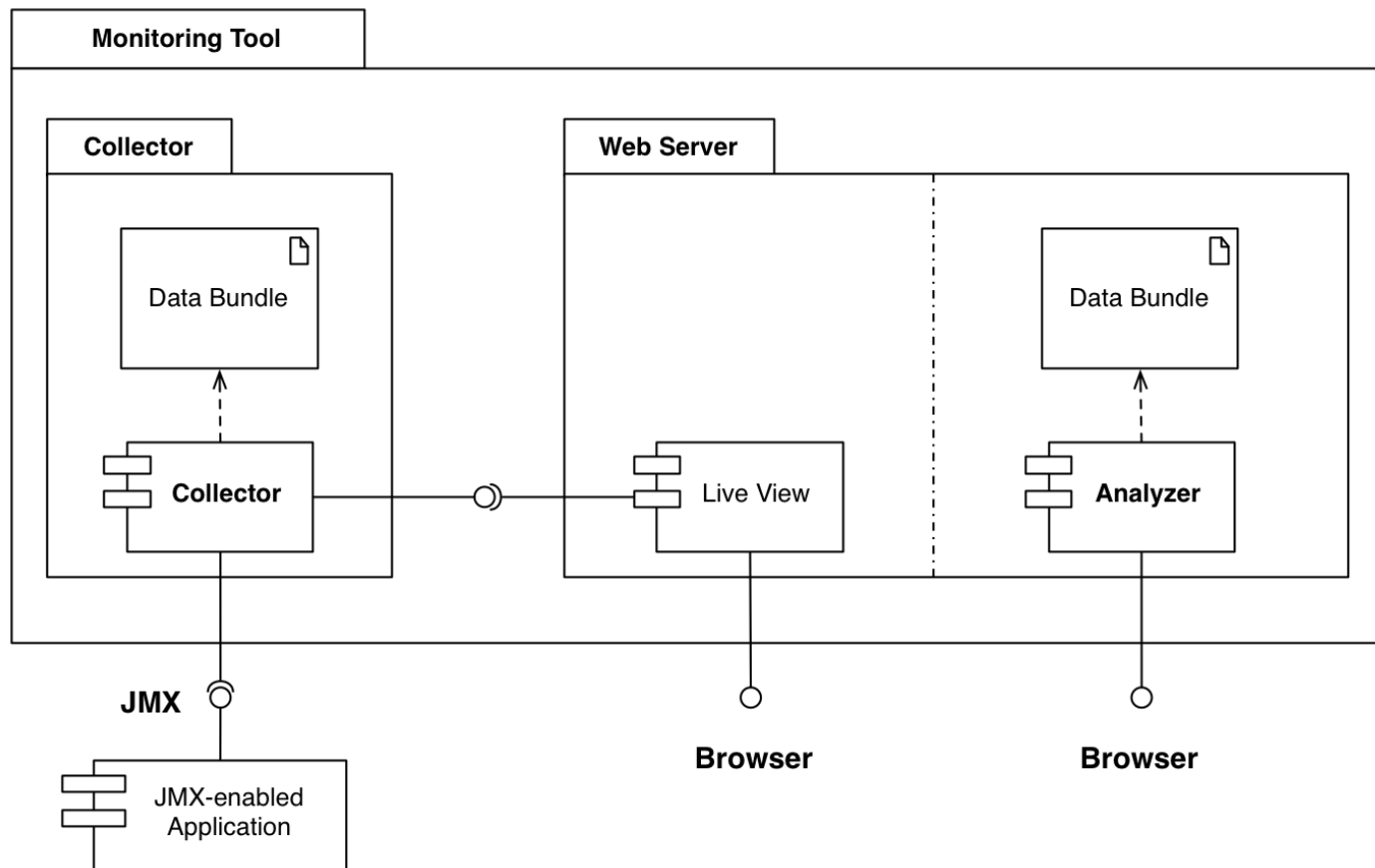
Monitoring Tool

- Graphical Visualization
- Detection
- Verification

APM for Tricia - Architecture



APM for Tricia - Architecture



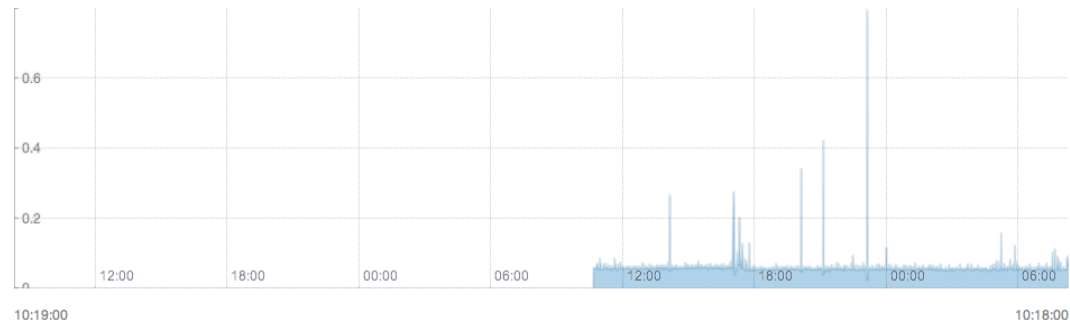
APM for Tricia - Architecture

Graphs

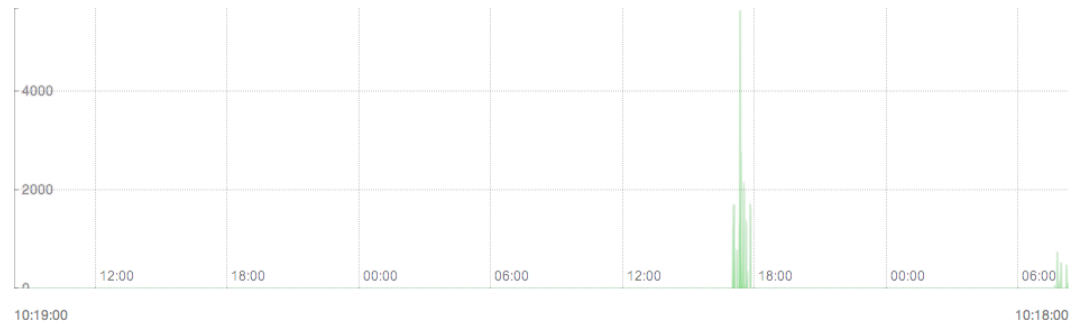
Selected time range: 2013-07-22 10:19:00 - 2013-07-24 10:18:00



Graph de.infoasset.monitoring.monitors.db.Mean



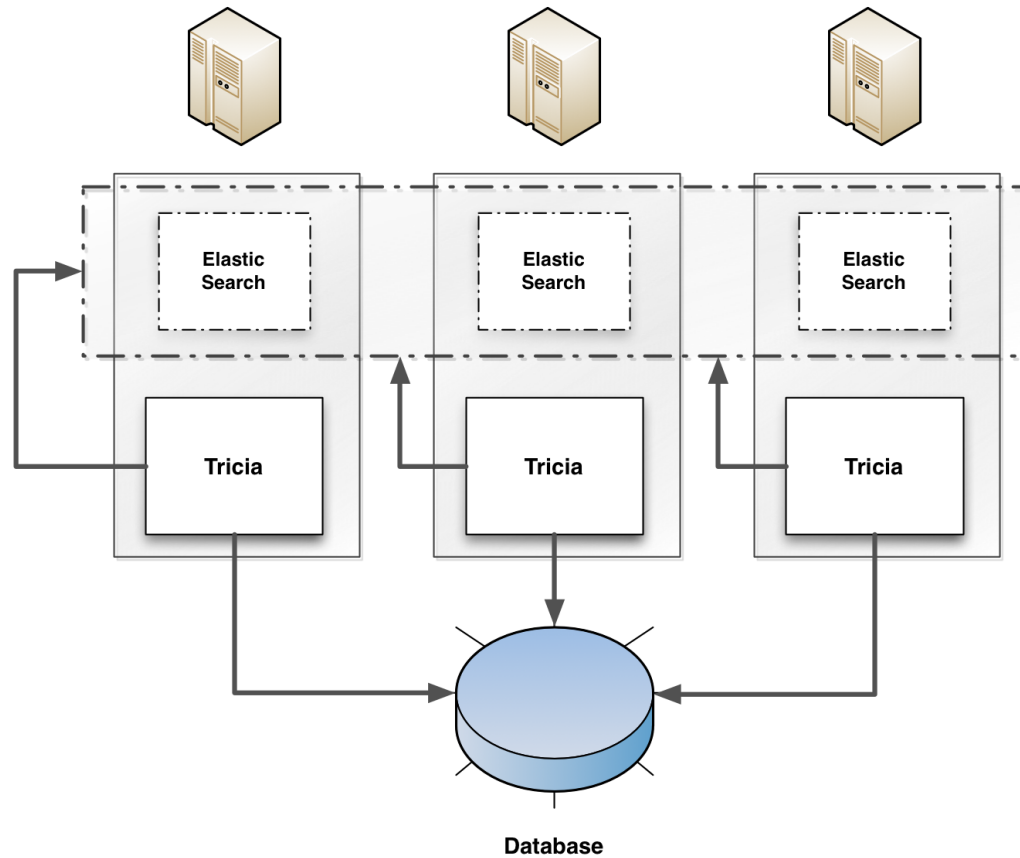
Graph de.infoasset.monitoring.monitors.es.Mean



Evaluation – Context

- Done in cooperation with *infoAsset*
- Customers reported slow performance
- Specific tasks took around **10 seconds** to complete
 - Creation of a workspace
 - No errors visible on UI
- Problem was reproducible
- *Business Marketplace* of DTAG
- Cloud Environment
 - Multiple servers working together
 - One shared database
 - One shared Elasticsearch cluster

Evaluation - Context



Evaluation – Checking for a problem

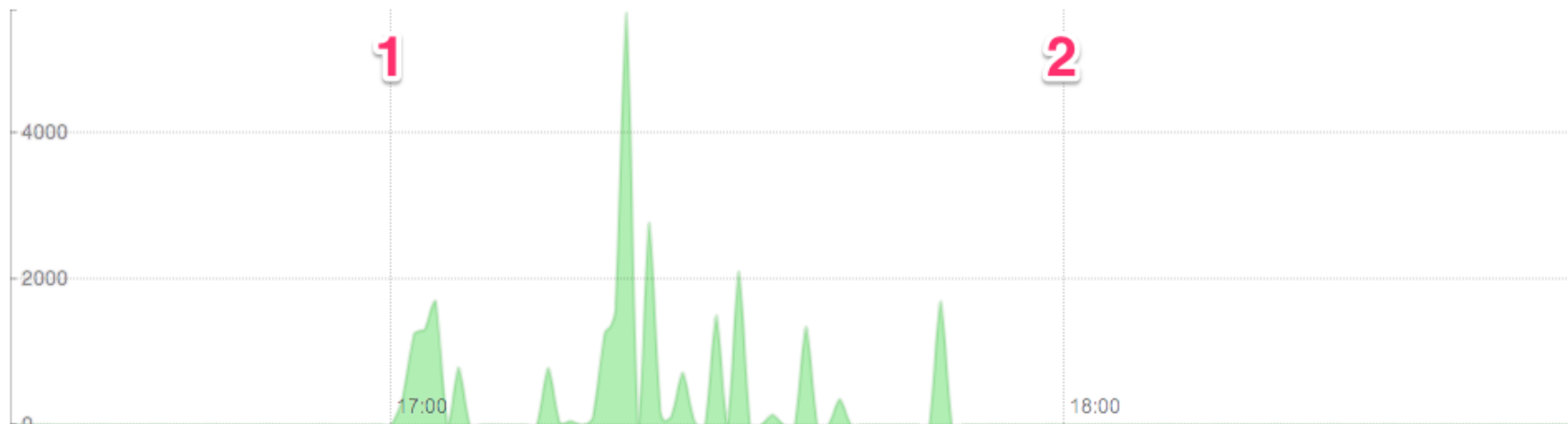
- Look at *Performance Snapshots*

Name	Hits	Avg	Total	Max
handler	608	292	177 920	85 877
entity.persist	1 541	1 660	2 559 448	154 076
es	1 886	580	1 094 084	85 545
es.searchables_production_multitenancy.commit	196	1 161	227 713	84 974
db	221 487	1	317 457	60 022

Evaluation – Verification

- Graphical visualization

Graph de.infoasset.monitoring.monitors.es.Mean



Evaluation – Tracking Down

- Perform slow request
- Analyze *Performance Trace*

9 607ms for /space/submit

Name	Hits	Total
webserver.handlerRequests	1	9 607
handler	1	9 605
es	10	8 788
es.searchables_production_multitenancy.commit	1	8 686

Evaluation – Tracking Down

- Commit operation performed by a single method
- Method is basic ES library call
- Implementation according to documentation
- Research on the internet produced no results
- Only environment with this problem

Evaluation – Results

- Low-level analysis revealed slow file deletion
- No bug in *Tricia*
- Issue caused by target infrastructure
- Provider was notified, hard drive problem solved

Further Work

- Higher degree of automatization
 - Extract data from log files
 - Alerts & Notifications
- Even more data via *JMX*
 - Performance Snapshots
 - Performance Trace

Summary

- Identification of key metrics
- Examination of existing solutions & standards
- Successfully employed APM in *Tricia*
- Evaluation of developed solution
- How-To for using the solution

Questions & Discussion

Thank you for your attention.

Michael Rose