A short introduction to

Allmon

a generic performance and availability monitoring system

Tomasz Sikora – London 2009

List of topics

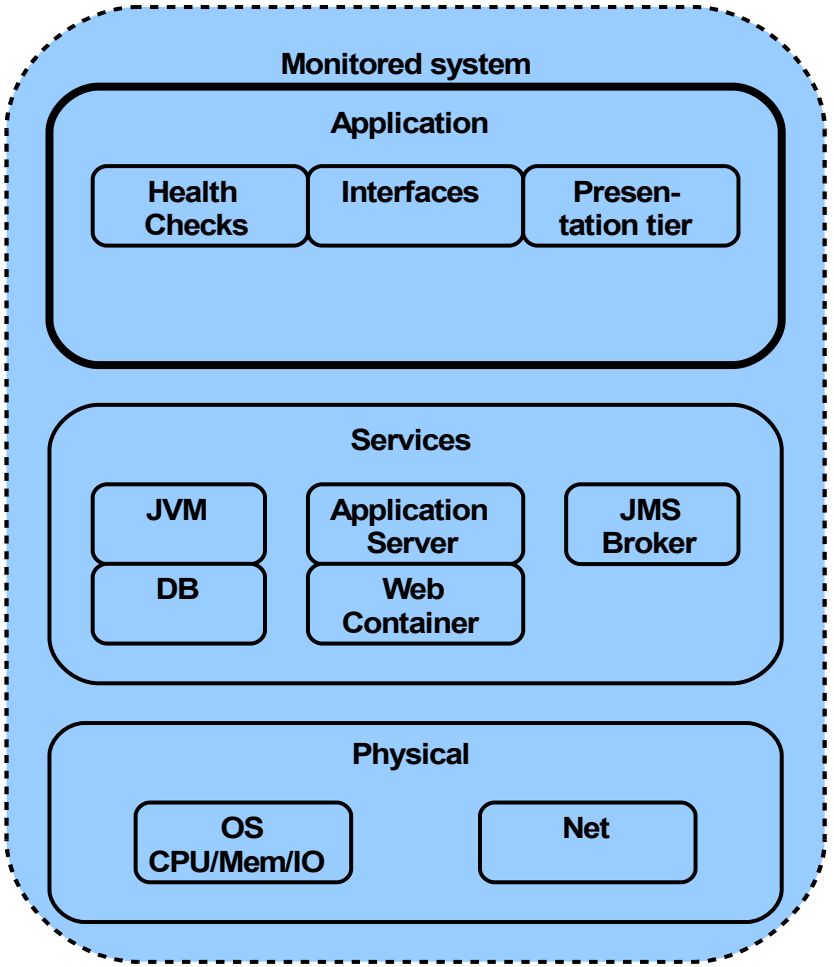
- Source of performance problems
- Continuous monitoring and metrics acquisition
- Allmon architecture (scalability, messaging)
- Deployment (distributed system)
- Configuration
- Analysis (use cases)
- Questions

Source of performance problems

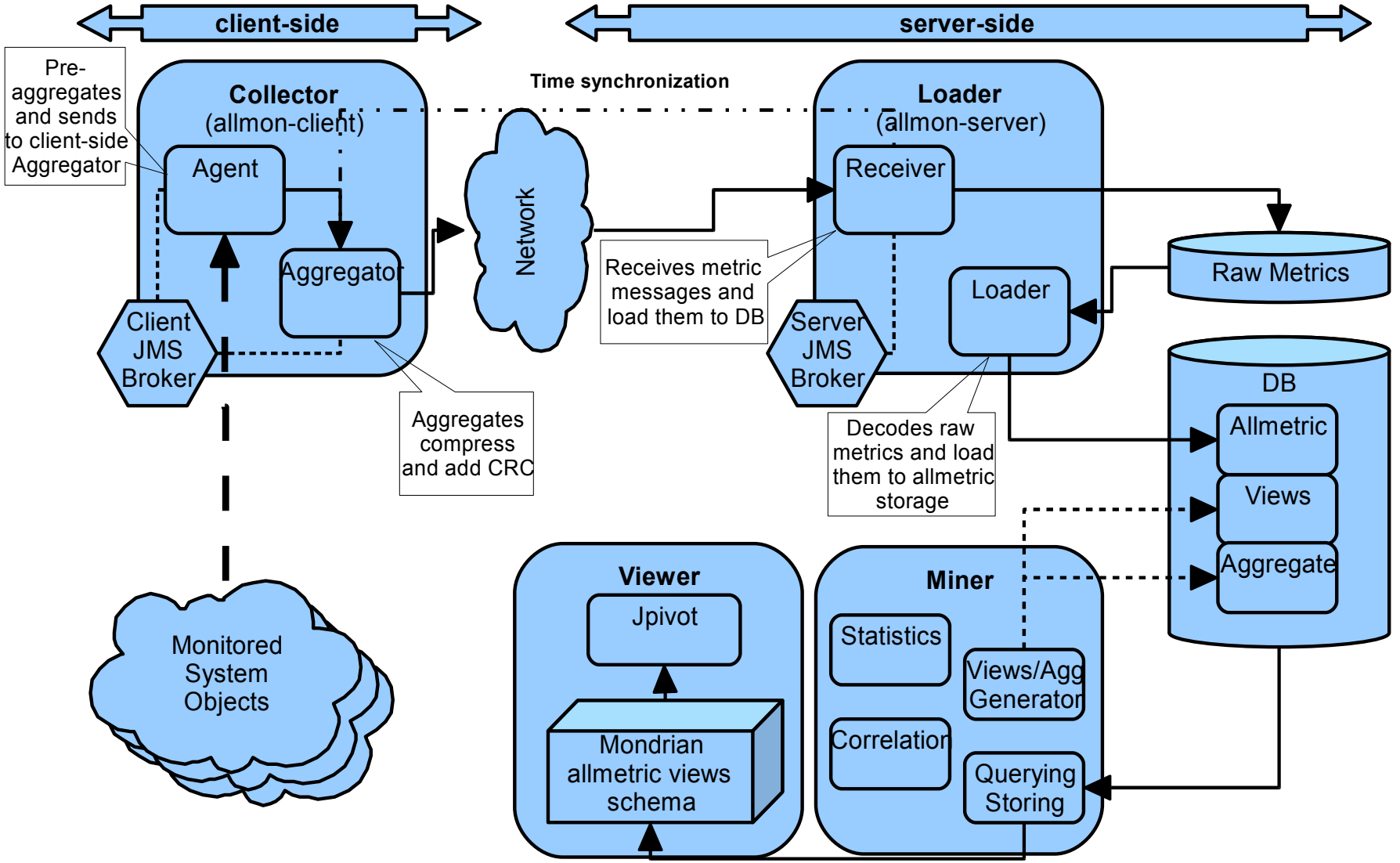
- **Lack of understanding (knowledge)**
  - Not well defined requirements (functional/non-functional)
  - Not experienced developers
  - Not well trained users
- **Lack of visibility**
  - Developers have to understand operational aspects
  - Multi-layer system monitoring is essential
- **Not enought testing**
  - Load tests (integration, regression) + Soak tests
  - Monitoring for "before-after" load test comparison and store all results

Continuous monitoring (metrics acquisition)

- **Monitoring multi-layer enterprice systems**
  - Main layers: Application, Services, Phisical
  - Service Health Checks
- **Distributed system**
  - Messaging: isolation, non-intrusive, reliable
- **What to monitor, what to analyse?**
  - Collected data can be used for correlation analysis



Allmon architecture

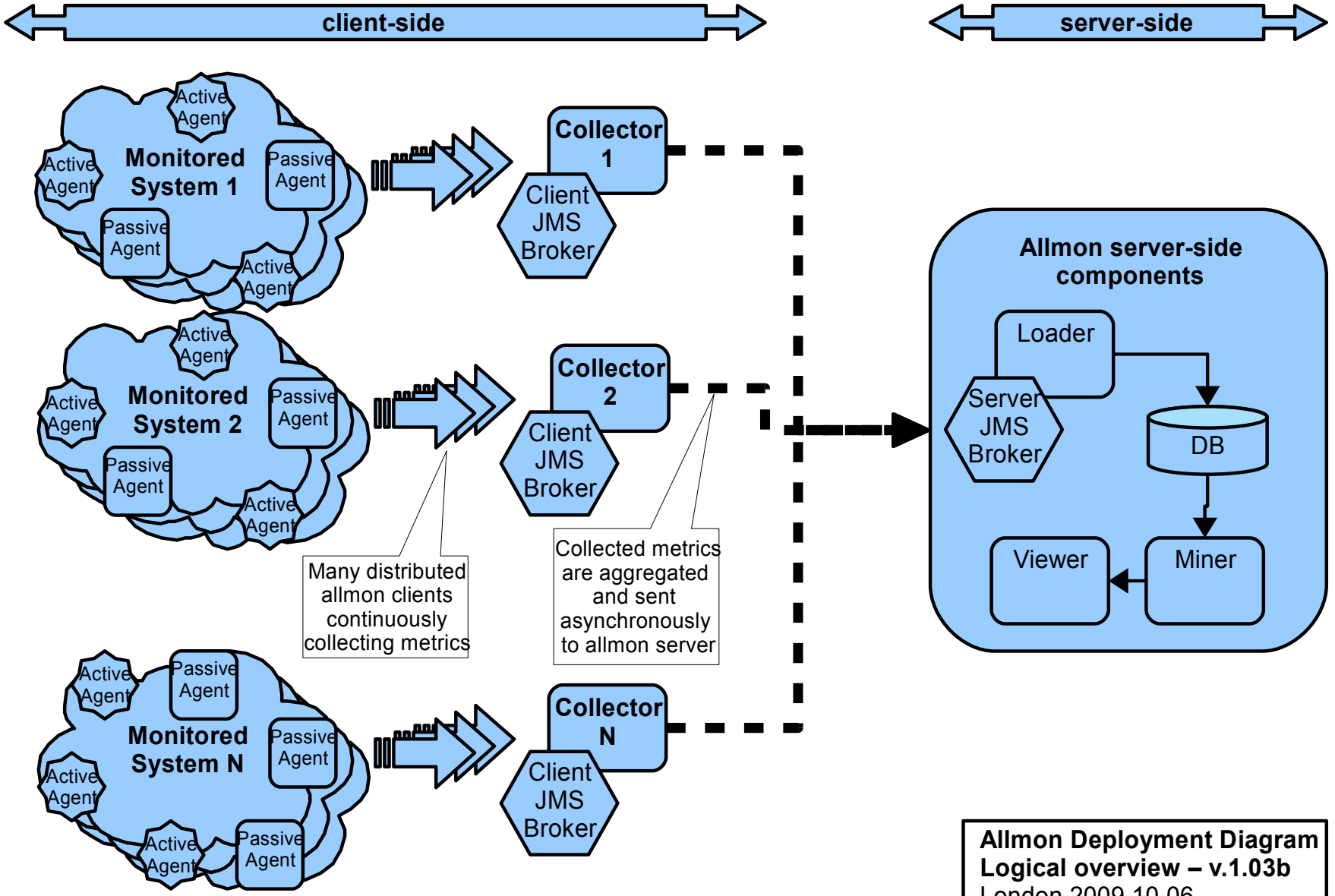


Allmon Components
Logical overview – v.1.03a
London 2009.10.06

Allmon architecture

- **Collector** (distributed client-side)
  - Agents (Passive/Active agents collecting metrics)
  - Aggregator (common pre-aggregating and sending data mechanism)
- **Loader** (centralized server-side)
  - Miner (transforming data to allmetrics, aggregating)
  - Viewer (presentation, multidimensional analysis)
- **Data storage**
  - Raw data storage (staging tier)
  - Allmetric schema (generic 3NF structure)
  - Aggregates, pre-calculated structures (access tier)

Deployment


client-side
server-side
Monitored System 1
Monitored System 2
Monitored System N
Active Agent
Passive Agent
Collector 1
Collector 2
Collector N
Client JMS Broker
Allmon server-side components
Loader
Server JMS Broker
DB
Viewer
Miner
Many distributed allmon clients continuously collecting metrics
Collected metrics are aggregated and sent asynchronously to allmon server
Allmon Deployment Diagram
Logical overview – v.1.03b
London 2009.10.06

Configuration

- **Client-side**
  - Agents configuration
    - Inedpendent configuration for different types of agents
  - Active agents scheduling
    - Based on crontab (cron4j)
- **Server-side**
  - Database conectivity
  - Loading process scheduling
  - Aggregate processes parametrisation
  - Visualization set-up

Allmon analysis - use cases

Collecting multi-tier system metrics is crucial for understanding system and finally finding performance problems (two simple examples)

- **Study case 1 -** Growing JVM memory allocation (leaks) in comparison between several releases and users activity
  - Input: JVM metrics via JMX (mem, GC), application actions stats
  - Output: differences in users activity, differences in application behaviour and allocated resources
  - Action: identifing areas in code base responsible for huge deltas in memory consumption
- **Study case 2 -** Not efficient interactions with services and databases
  - Input: database metrics, DB OS stats, application actions stats, intercepted persistence level calls
  - Output: rankings of: the longes performing application actions, the biggest product of execution count and exection times
  - Action: Easier prioritisation of areas which have to be improved

Questions

? ? ?

Allmon
2009

Allmon project page:
http://code.google.com/p/allmon/

Allmon user group:
http://groups.google.com/group/allmon

Code license: Apache License 2.0:
http://www.apache.org/licenses/LICENSE-2.0