alphadoop

Doflamingo

An light-weight monitoring system for Apache Hadoop

TITLE **Kafka/ Zookeeper Monitoring Module built for Flamingo Ecosystem**

DURATION **March 13, 2016 ~ June 8, 2016**

CLIENT **EXEM** **PRESENTER** **ALPHADOOP**

CONTENTS

_ WHAT WE WILL DO

Collect Performance Metrics,
Visualize it, and
Integrate it with Flamingo.

_ WHAT WE WILL DO

Is all system working properly?

hadoop

kafka

Doflamingo

Of Course!

Check this out!

_ WHY WE NEED THIS PROJ

1. Hard to understand Hadoop

Distributed system – not intuitive

Unable to track fluctuant mass traffic

Eyes on only the upper level
– run and hope everything goes well

_ WHY WE NEED THIS PROJ

2. The Missing Link of Flamingo

Currently flamingo is able to monitor:

- Resources
- YARN application
- Map Reduce
- Nodes

_ HOW WE DO IT

Learn from other monitoring tools

Plenty of tools exists in the field – Learn from them and try to build up similar metrics

Build it into flamingo platform

There's flamingo's way of monitoring hadoop system. Add a new task into jobscheduler.

_ HOW WE DO IT

AGILE APPROACH

1 SPRINT = 2 WEEKS

TOTAL 5 SPRINTS along the semester

_ REQUIREMENTS

1. **Built as a part of Flamingo system**
2. **Monitor and Report in Real-time**
3. **Utilize JVM ecosystem**
4. **Visualize the metrics, avoid numbers**
5. **Save metrics into Database**
6. **Special caution on log management**

_ OBJECTIVES

O1: Set up an environment for Flamingo — SPRINT 1

O2: Define Kafka measurement metrics, visualization forms

O3: Implement API server which provides collected metrics — SPRINT 2

O4: Implement charts with Sencha

O5: Integrate with Flamingo Ecosystem — SPRINT 3

O6: Define Zookeeper measurement metric, visualization — SPRINT 4

O7: Implement a Zookeeper monitoring module on Flamingo — SPRINT 5

KAFKA MODULE → M1

ZOOKEEPER MODULE → M2

_ TECHNICAL CHALLENGES

Simulate distributed environment

Kafka and zookeeper can only be tested in multiple nodes. Need to mock clustering env.

REQUEST → EXEM

Can we have sample environment or at least a tutorial that we can follow to setup distributed system?

_ TECHNICAL CHALLENGES

Selecting the important metrics

New to monitoring job and hadoop so we don’t know what are the important metrics

HOW WE WILL SOLVE THE CHALLENGE

Survey other services: what they are monitoring and ordering of metrics which implicitly denotes importance

Interview on developers – maybe EXEM engineers?

_ THE EFFECT OF OUR WORK

The ultimate control tower

Flamingo now monitors not only nodes,
but also modules that compose pipeline.

Opening up new possibility

The gathered metrics can be used for further
optimization or anomaly detection feature.

_ WHO WILL DO WHAT

TEAM _ ALPHADOOP

SEUNGHYO KANG *the hadoop master*

← **Metric Analysis**

RESTful Server →

JARYONG LEE *the spring master*

YOUNGJAE CHANG *the sencha master*

← **Visualization**

_ WE ARE RESPONSIBLE FOR:

1. Built as a open source software

Fork and request merge into flamingo

License/ Copyrights are same with flamingo

2. Bye-bye after spring semester

A/S are not supported after June 21, 2016

_ WE ONLY HAVE THESE:

LIMITED TIME: 10 WEEKS

No delay accepted – when semester ends, project should be ended

LIMITED DEVELOPERS: 3 PEOPLE

No one will help us
– no money to hire someone!

_ SAY HELLO TO MONITORING

Seeing is believing

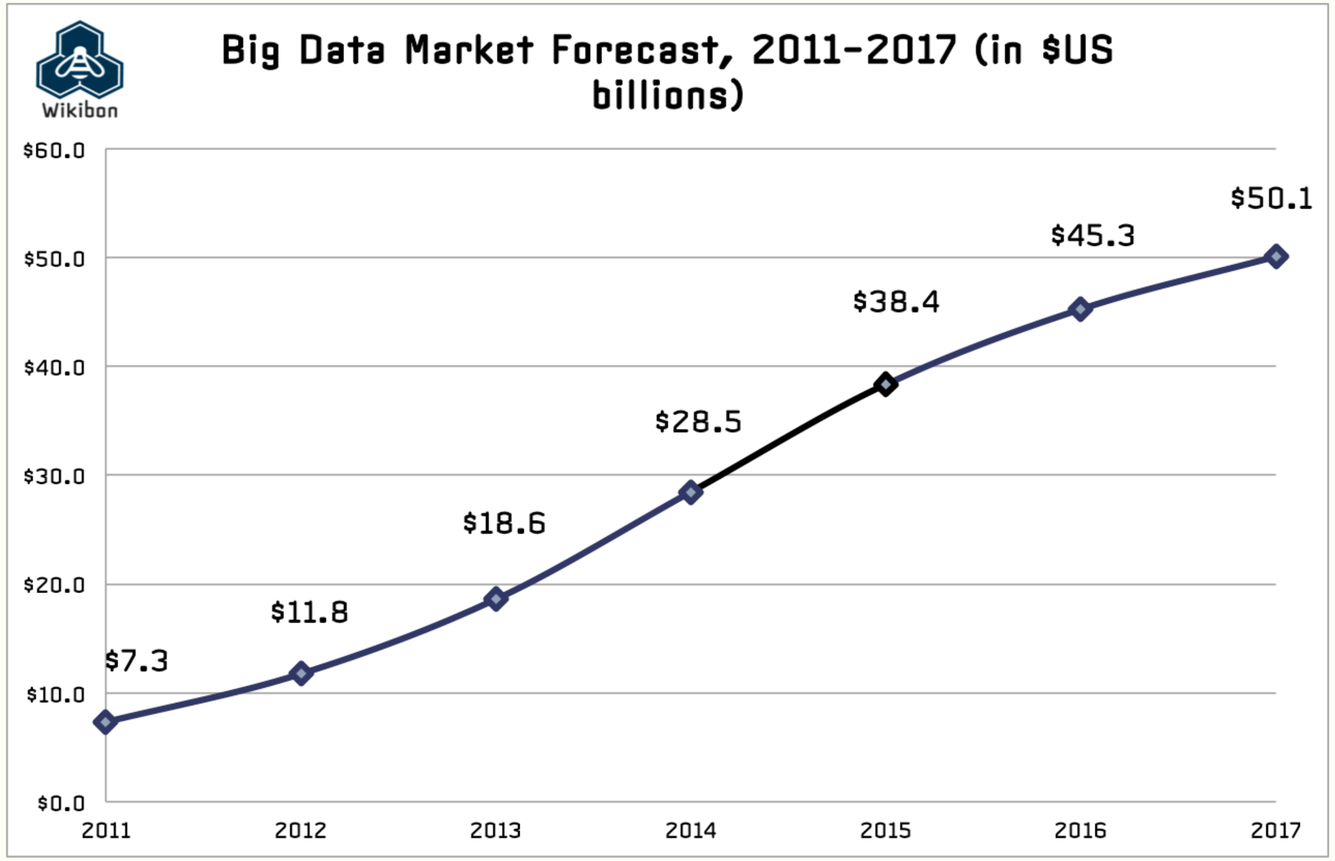
Software is intangible; so, where can we find it?

Bigdata: the buzz needs money

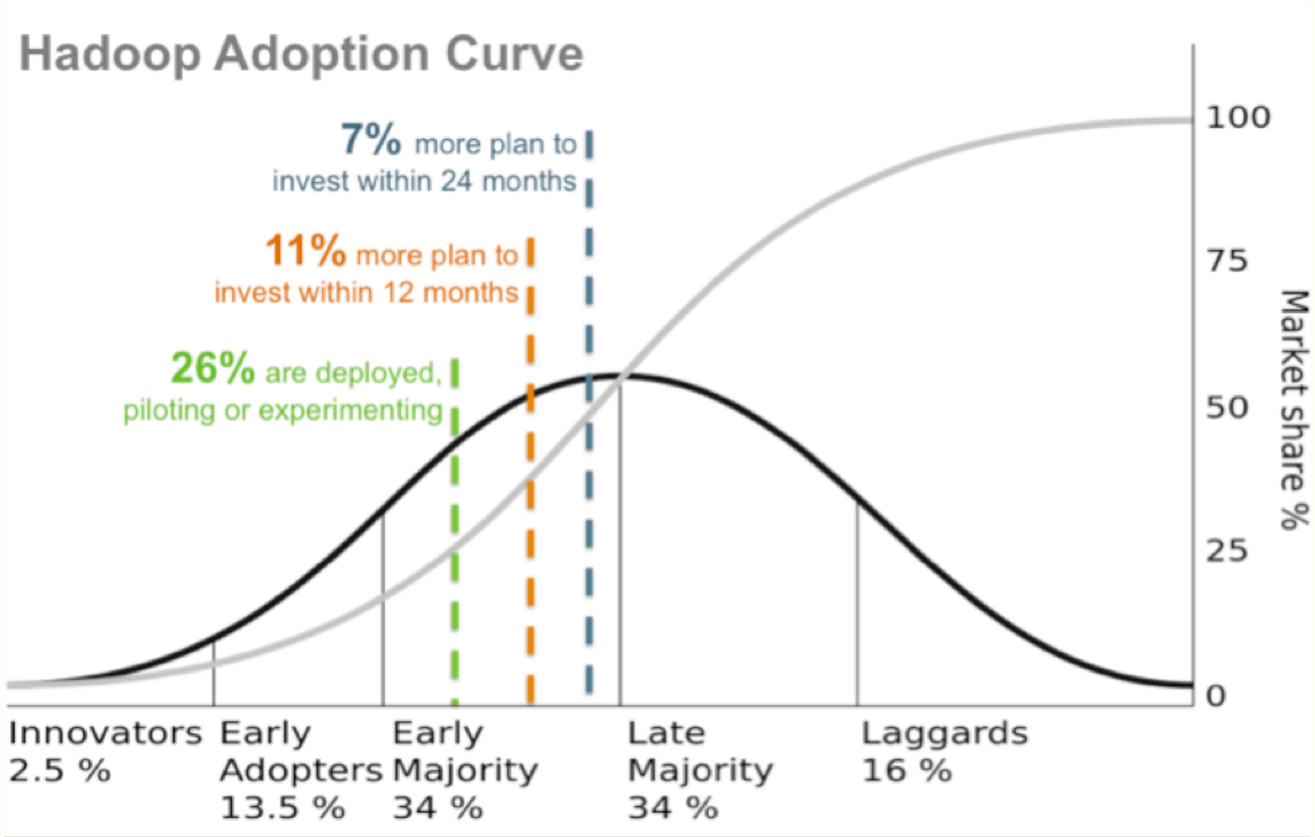
Hadoop is a money-eater:

10+ nodes, consulting, (expensive) engineers

_ SAY HELLO TO MONITORING



_ SAY HELLO TO MONITORING



_ TECHNICAL DETAILS

[A] WHAT IS KAFKA?

A high-throughput distributed messaging system

BENEFITS

- Scalable
- High-throughput
- Distributable
- Low response time
- Save on data disk

USED IN

- LinkedIn
- Twitter
- Netflix
- Tumblr
- Foursquare

_ TECHNICAL DETAILS

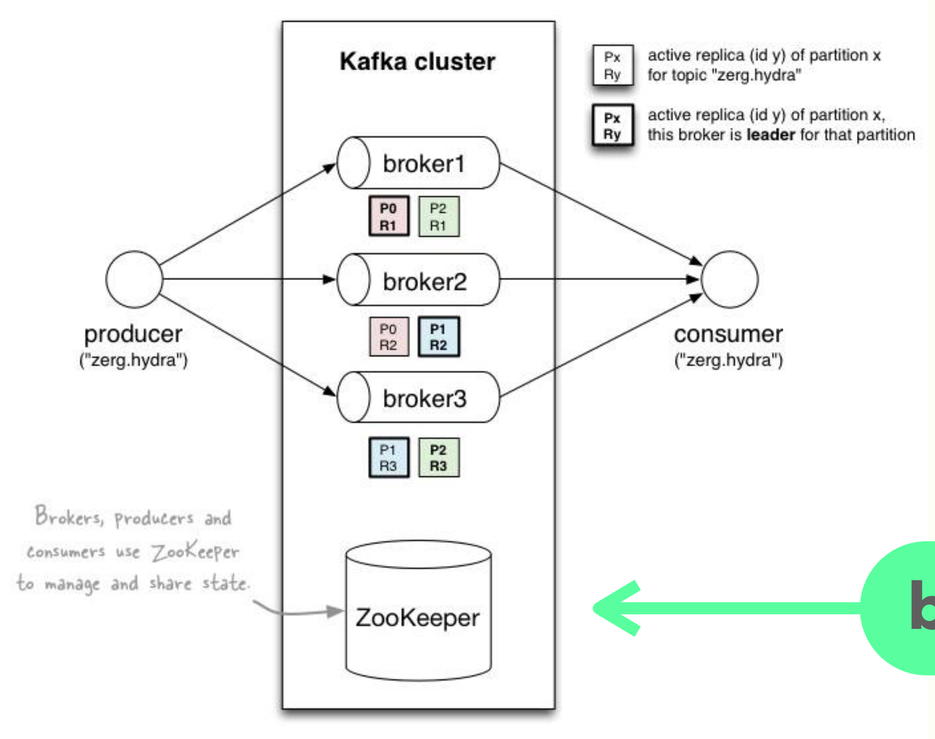
[A] WHAT IS KAFKA?

Kafka consists of producer, broker, and consumer, managed by **Zookeeper**

Producers send system messages to **brokers**
Brokers process them **distributively**
Consumers store the messages to their **disks**

_ TECHNICAL DETAILS

[A] WHAT IS KAFKA?



producer
generate message that fall into certain topics

consumer
subscribe specific topics & process the message

broker
stacks up logs based on topic

_ TECHNICAL DETAILS

[A] WHY KAFKA?

Store the messages in the **DISK**, not in the cache.

Consumers can rewind back to old data and re-consume them since they are in the disk for a certain period of time.

PULL model, not push model

consumer pull messages from broker without exceeding their limit; no drop occurs unlike producer-push model

_ TECHNICAL DETAILS

[B] WHAT IS ZOOKEEPER?

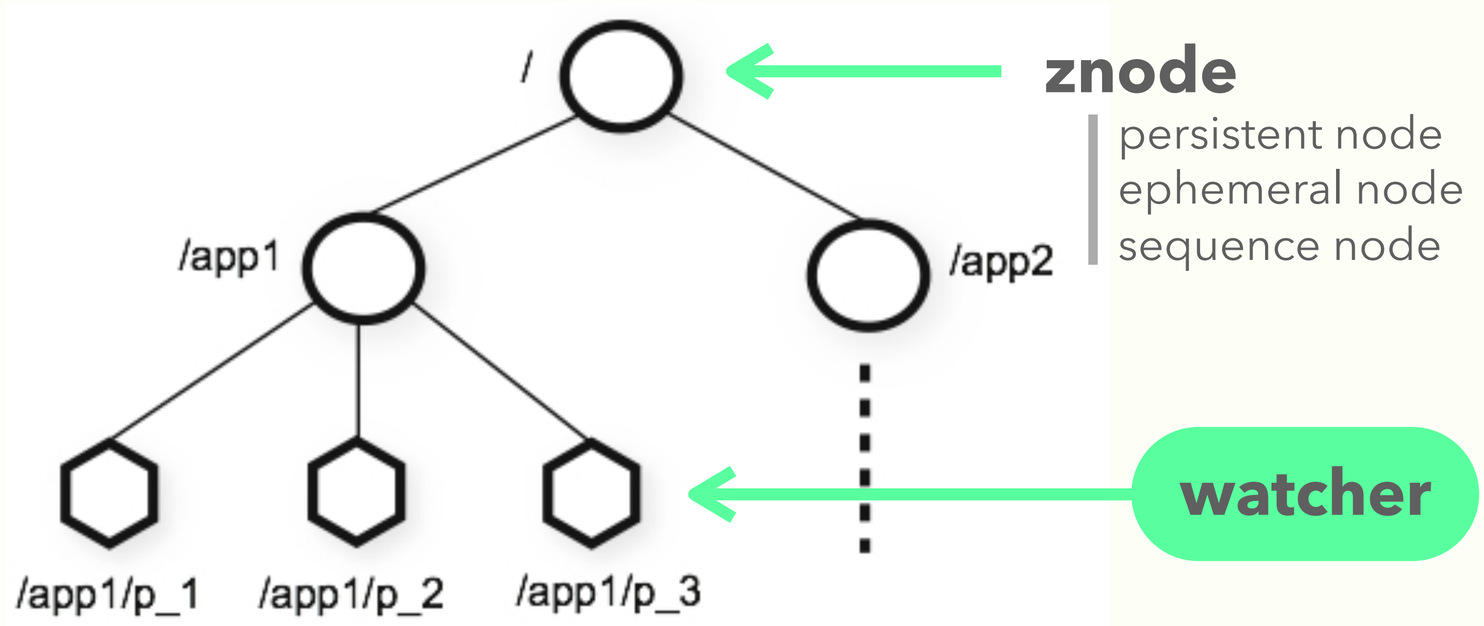
Handles various errors in distributed systems.

Four Features

Using name service to separate loads.
Using distributed lock to handle synchronization error
Error detection and recovery
Configuration management

_ TECHNICAL DETAILS

[B] WHAT IS ZOOKEEPER?



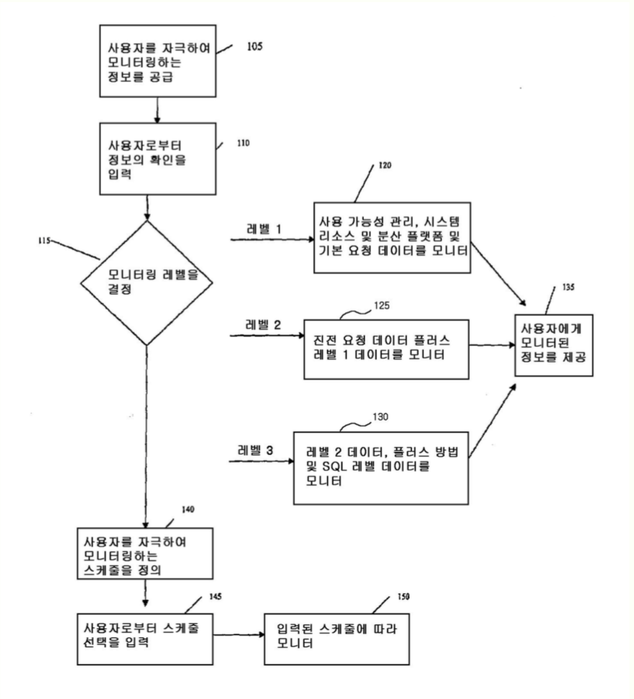
http://bcho.tistory.com/1016

_ PATENT RESEARCH

METHOD AND SYSTEM FOR MONITORING PERFORMANCE OF APPLICATIONS IN A DISTRIBUTED ENVIRONMENT

KR 0772999 B1

IBM
Assignee

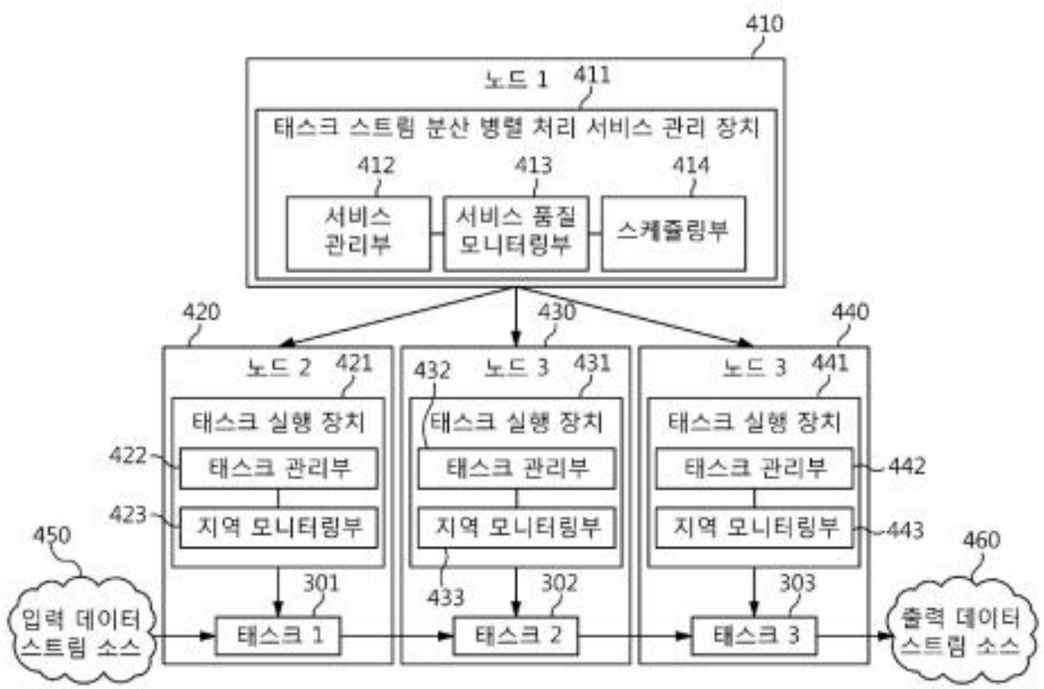


_ PATENT RESEARCH

APPARATUS AND METHOD FOR MANAGING DATA STREAM DISTRIBUTED PARALLEL PROCESSING SERVICE

KR 2013-0095910 A

ETRI
Assignee



_ PATENT RESEARCH

APPARATUS AND METHOD FOR ANALYZING BOTTLENECKS IN DATA DISTRIBUTED PROCESSING SYSTEM

KR 2015-0050689 A

SAMSUNG ELECTRONICS SEOUL NATIONAL UNIV.

Assignee

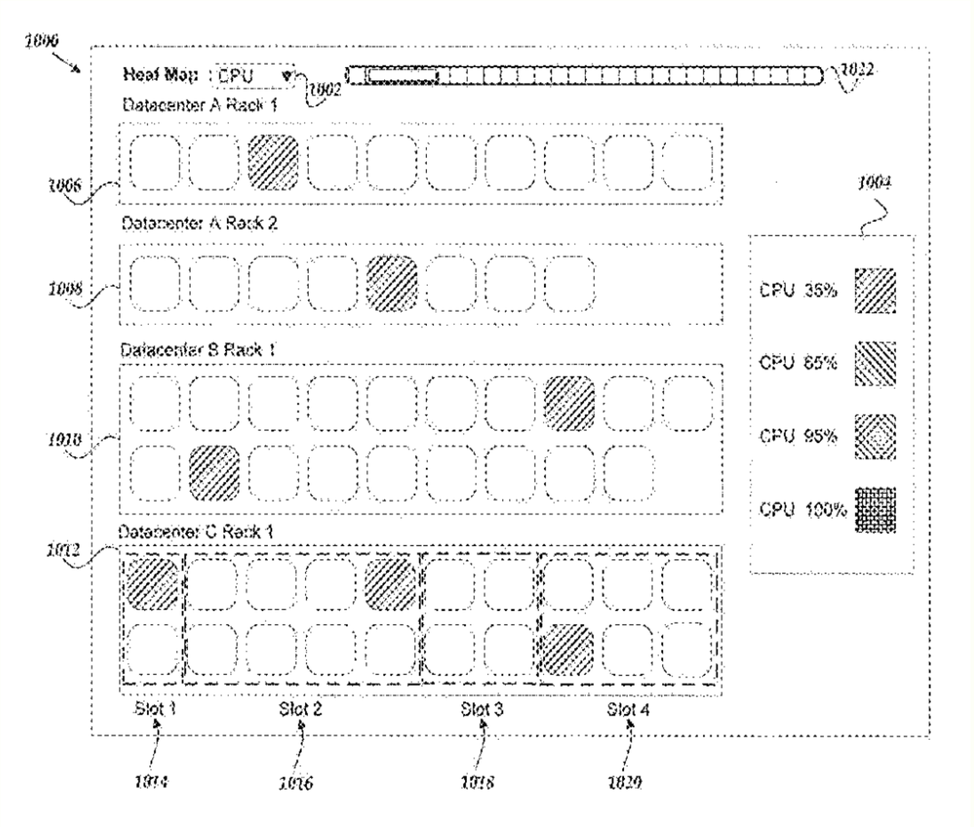
_ PATENT RESEARCH

CLUSTER PERFORMANCE MONITORING

US 9043332 B2

Splunk

Assignee



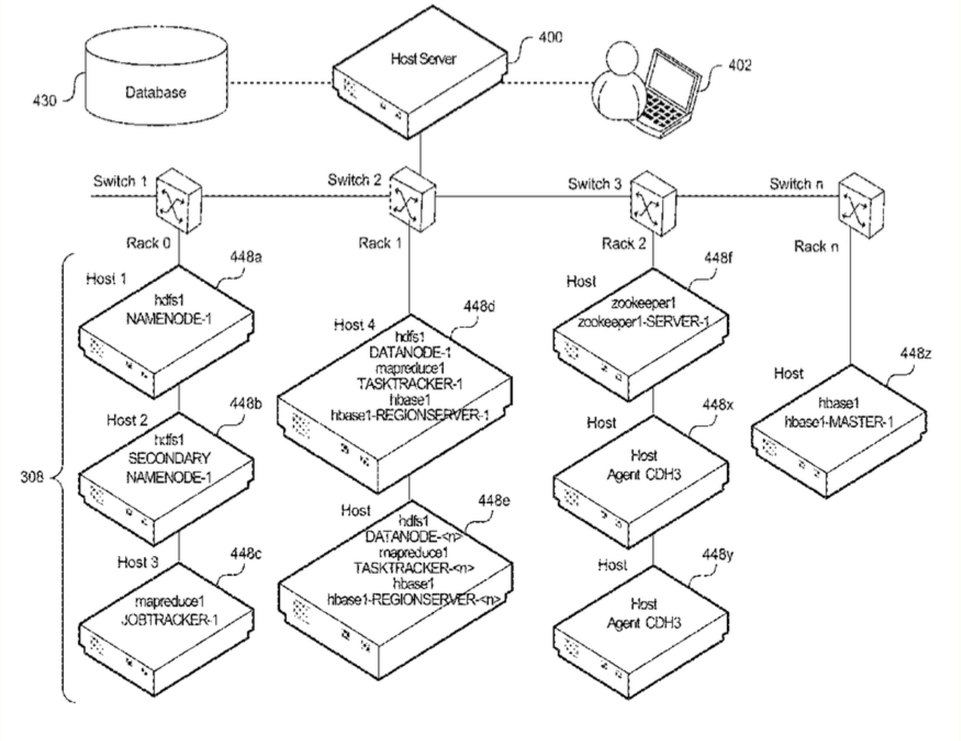
_ PATENT RESEARCH

CENTRALIZED CONFIGURATION AND MONITORING OF A DISTRIBUTED COMPUTING CLUSTER

US 9172608 B2

Cloudera

Assignee



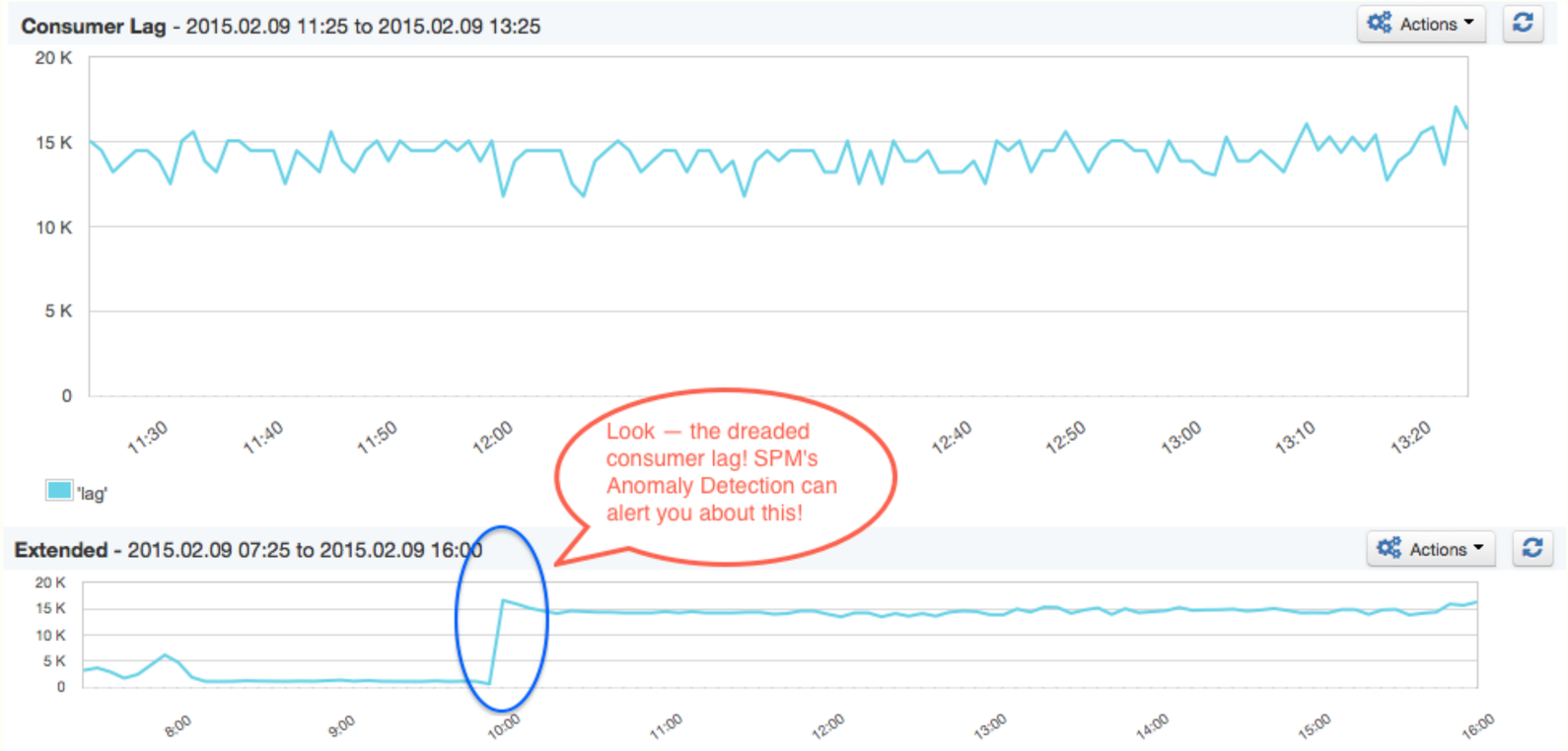
_ COMMERCIAL PRODUCT

METRIC ANALYSIS

	A	B	C	D	E	F	G	H
1	Category	Metrics		Questions	MBean name	Suggested Alert	Chart	
2	Running	1	Kafka Process	Is the right binary daemon process running?		Suggested Alert	맞추어야 하는 조건	
3	System	2	Memory Usage	Kafka should run entirely on RAM. JVM heap size shouldn't be bigger than		None	맞추어야 하는 조건	
4		3	Swap Usage	Watch for swap usage, as it will degrade performance on Kafka and lead to		When used swap is >	맞추어야 하는 조건	
5		4	Network Bandwidth	Kafka servers can incur a high network usage. Keep an eye on this, espec		None	그래프	1
6		5	Disk Usage	Make sure you always have free space for new data, temporary files, snap		When disk is > 85% u	? 그래프	10
7		6	Disk IO	Kafka partitions are stored asynchronously as a sequential write ahead log		None	? 그래프	10
8	Kafka	7	UnderReplicatedPartitions	아직 복제가 완료되지 못한 파티션의 개수 Number of under-r	kafka.server:type=	When UnderReplicate	존재하면 알림?	
9		8	OfflinePartitionsCount	리더가 없는 파티션의 개수 Number of partitions without an	kafka.controller:typ	When OfflinePartitions	존재하면 알림?	
10		9	ActiveControllerCount	잘 작동하는 controller 브로커(?)의 개수 Number of active c	kafka.controller:typ	When ActiveControlle	존재하면 알림?	
11		10	MessagesInPerSec	초당 들어오는 메세지 수 Incoming messages per second.	kafka.server:type=	None	그래프	2
12		11	BytesInPerSec / BytesOutPerSec	들어오고 나가는 바이트 수 Incoming/outgoing bytes per se	kafka.server:type=	None	그래프	2
13		12	RequestsPerSec	초당 요청 수 Number of requests per second.	kafka.network:type	None	그래프	2
14		13	TotalTimeMs	메세지 하나를 처리하는 데 걸리는 시간 Total time it takes to	kafka.network:type	None	그래프	3
15		14	UncleanLeaderElectionsPerSec	리더가 빠르게 선출되지 않는 선거의 개수 Number of dispute	kafka.controller:typ	When UncleanLeader	존재하면 알림?	
16		15	LogFlushRateAndTimeMs	로그 플러쉬가 일어난 속도/시간 Asynchronous disk log flus	kafka.log:type=Log	None	그래프	4
17		16	PartitionCount	전체 파티션의 개수 Number of partitions on your system.	kafka.server:type=	When PartitionCount !	이상하면 알림?	
18		17	ISR shrink/expansion rate	브로커가 죽어서 복제본의 숫자가 줄거나 늘었을 때 When a b	kafka.server:type=	IsrShrinksPerSec | Isr	이상하면 알림?	
19		18	NetworkProcessorAvgIdlePercent	네트워크 활동이 없는 시간의 비율 The average fraction of t	kafka.server:type=	When NetworkProces	이상하면 알림?	
20		19	RequestHandlerAvgIdlePercent	리퀘스트가 들어오지 않는 시간의 비율 The average fraction	kafka.server:type=	When RequestHandle	이상하면 알림?	
21		20	Heap Memory Usage	자바에 동적 할당된 메모리 (주키퍼) Memory allocated dynamically by the Java		None	그래프 위쓰 쓰레쉬홀드	5
22	Consumer	21	MaxLag	큐에 쌓인 메세지 개수 Number of messages by which the	kafka.consumer:ty	When MaxLag > 50.	그래프 위쓰 쓰레쉬홀드	6
23		22	MinFetchRate	컨슈머가 브로커에게 보내는 요청의 속도의 최소 Minimum rat	kafka.consumer:ty	When MinFetchRate <	그래프 위쓰 쓰레쉬홀드	7
24		23	MessagesPerSec	초당 소비되는 메세지 Messages consumed per second.	kafka.consumer:ty	None	그래프	8
25		24	BytesPerSec	초당 소비되는 바이트 Bytes consumed per second.	kafka.consumer:ty	None	그래프	8
26		25	KafkaCommitsPerSec	컨슈머가 카프카에게 오프셋을 보내는 속도 Rate at which co	kafka.consumer:ty	None	그래프	9
27		26	OwnedPartitionsCount	이 컨슈머가 갖고 있는 파티션 수 Number of partitions owne	kafka.consumer:ty	When OwnedPartition	이상하면 알림?	

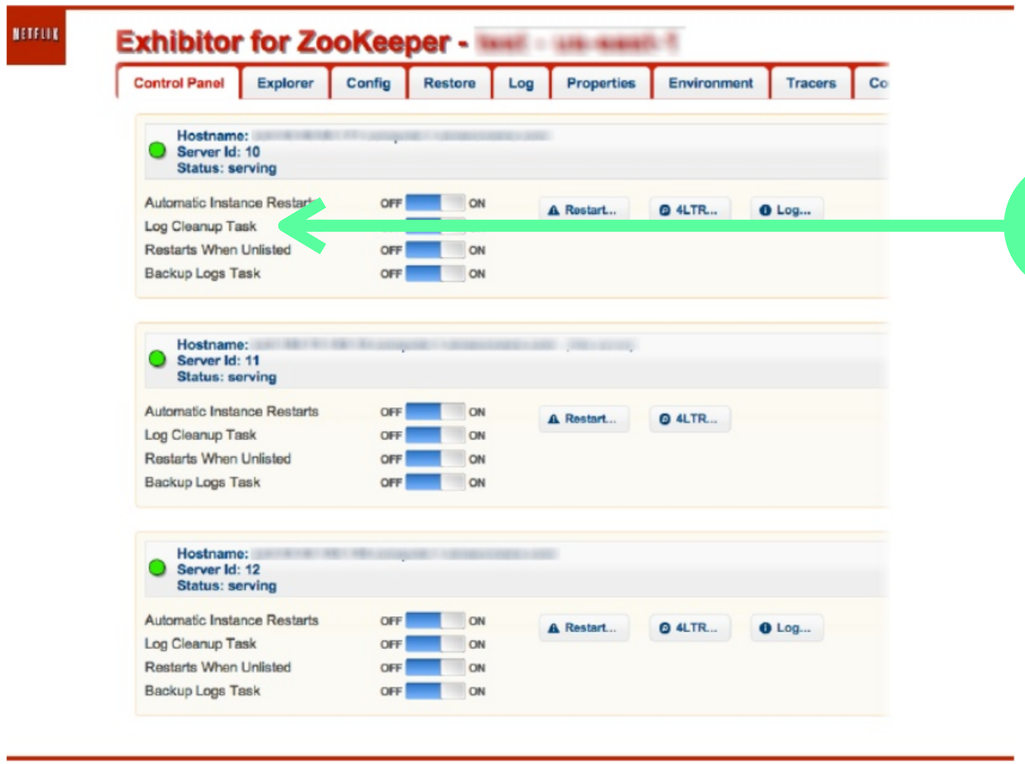
_ COMMERCIAL PRODUCT

SPM KAFKA - CONSUMER LAG



_ COMMERCIAL PRODUCT

NETFLIX EXHIBITOR FOR ZOOKEEPER



log cleanup task

_ OUR DRAWINGS OF FUTURE

Established Programs

Finished product for sale
Difficult to modify or fool freely
Take a long time to supplement new functionality
Hard to come up with creative one

Flamingo

Open source
Easy to be fooled by developers
Developer-driven modules;
can freely build creative tools

_ QUESTION

PHASE #1 **What is a monitoring?**

PHASE #2 **Why do we monitor?**

_ TWO NEEDS

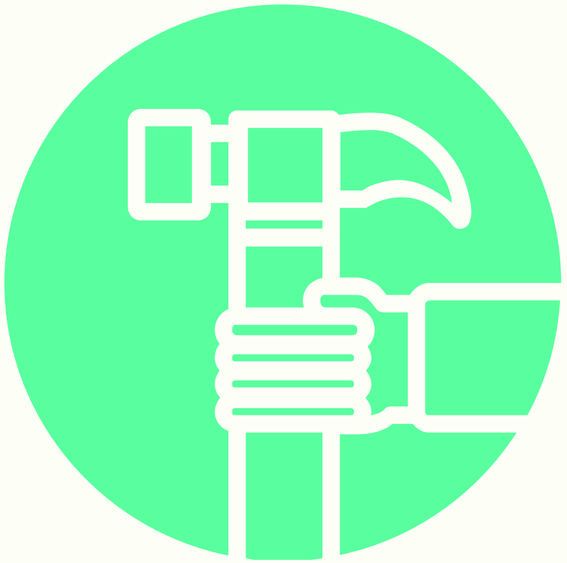
*To ensure the **normal operation** of the system*

*To find out the cause of **abnormal behavior***

_ TWO USERS

USER #1

Administrator



USER #2

Engineer

_ USERS

USER #1

Administrator

A. Hope everything stays normal
B. Determine whether to put more resources or not
C. Usually maintains a volume of system
D. Focus on real-time data

USER #2

Engineer

A. Fix the problem
B. Find out the cause of the problem by traveling the past data
C. Deeper understanding on whole system
D. Focus on specific events

_ DIFFERENT REQUIREMENTS

USER #1

Administrator

A. Visualize constantly changing statistics of sys.

B. At a glance view of metrics

C. Real-time update without user intervention

USER #2

Engineer

A. Visualize abrupt events

B. Can travel back to the past to find the cause of event

C. Detailed analysis on changing variables during specific timeframe

_ EXTERNAL INTERFACE

FUNC #1

Overview

A. Dashboard

B. ~~Configuration~~

FUNC #2

Timeline

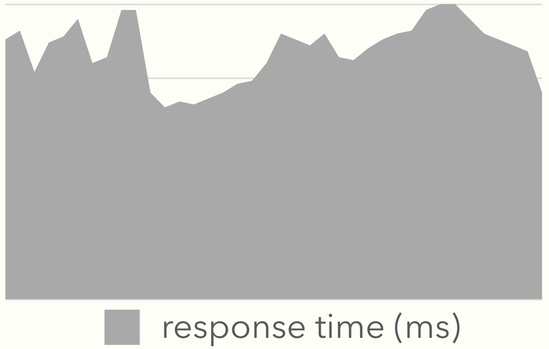
A. Event Timeline

B. Timemachine

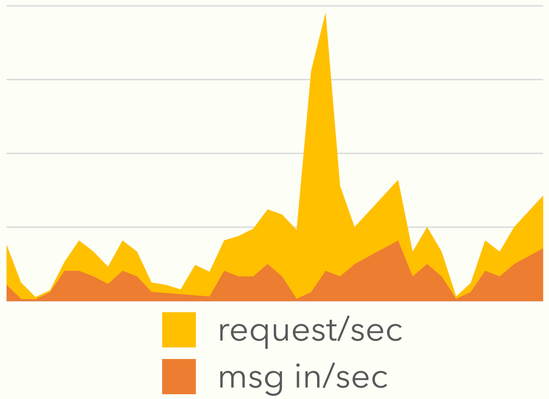
FUNC #1

Overview

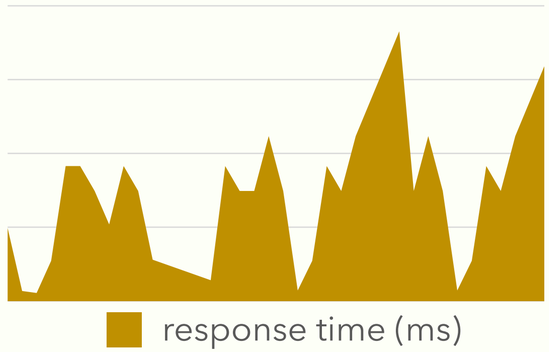
Heap memory usage



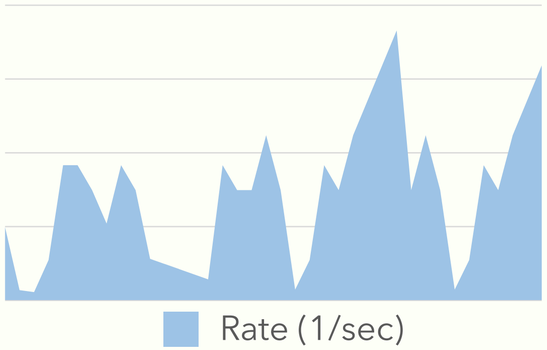
Message Condition



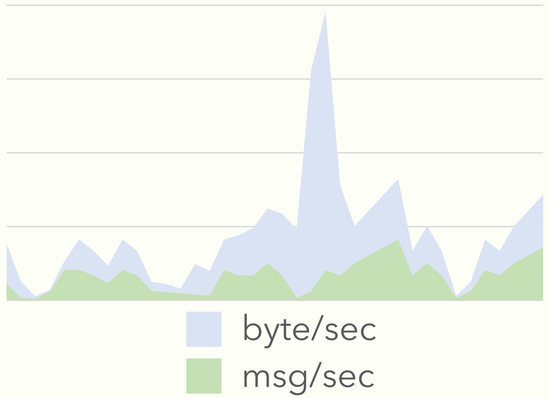
Response time



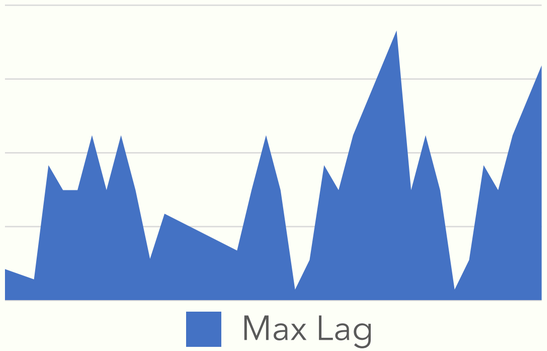
Minimum Fetch rate



Message Consumed

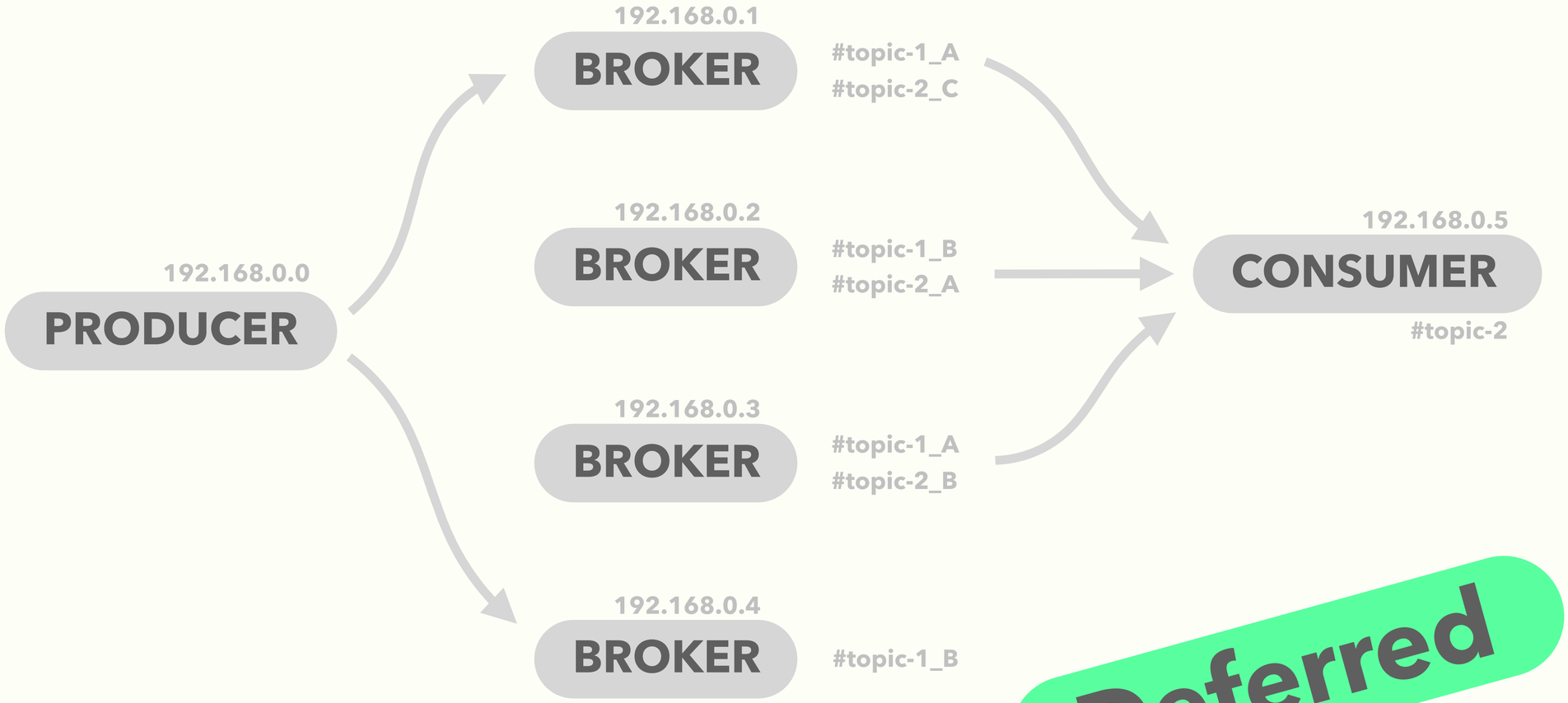


Max Lag



FUNC #1

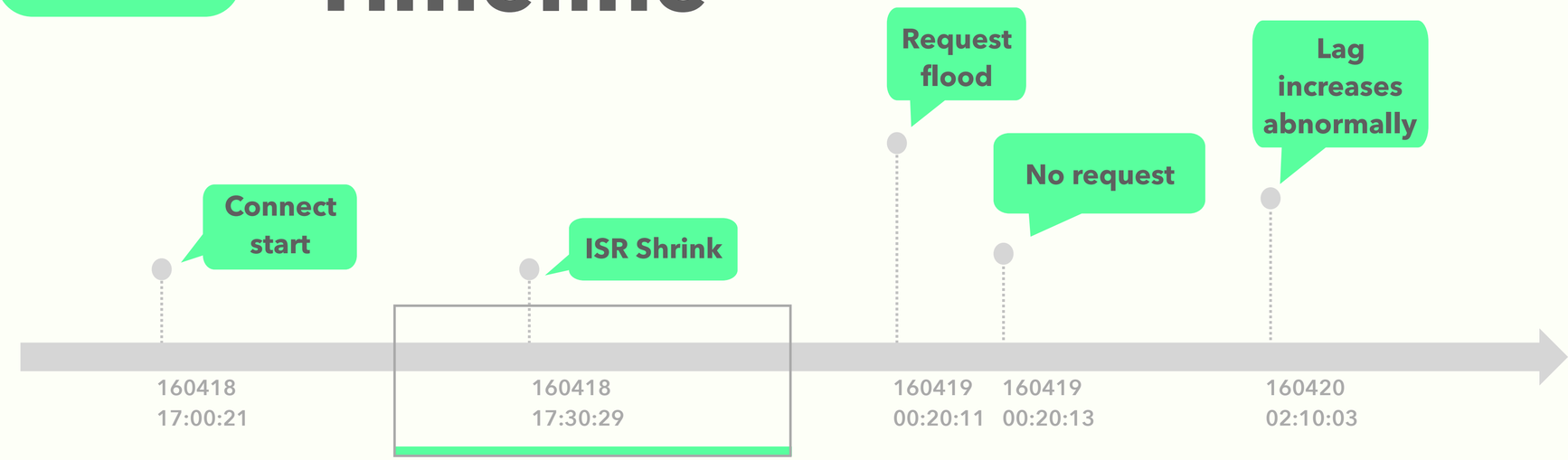
Overview



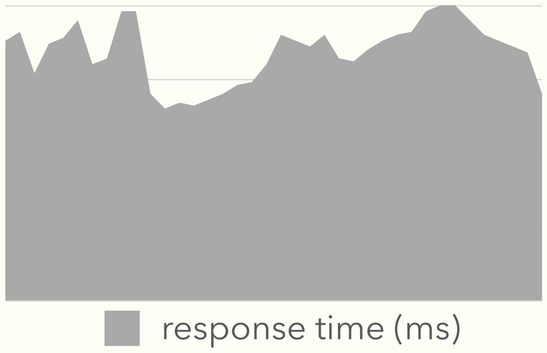
Deferred

FUNC #2

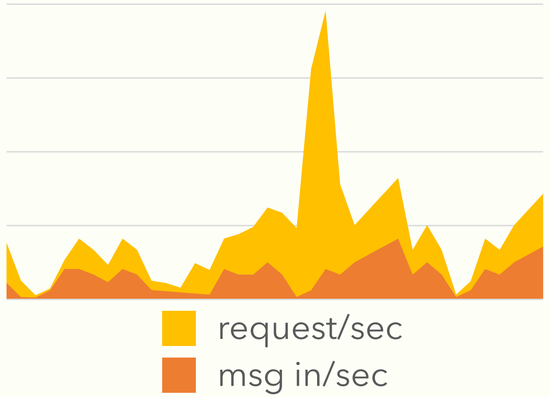
Timeline



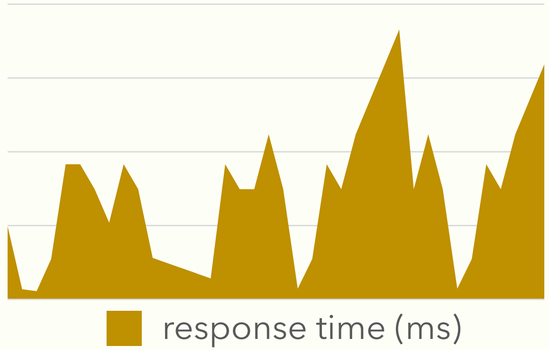
Heap memory usage



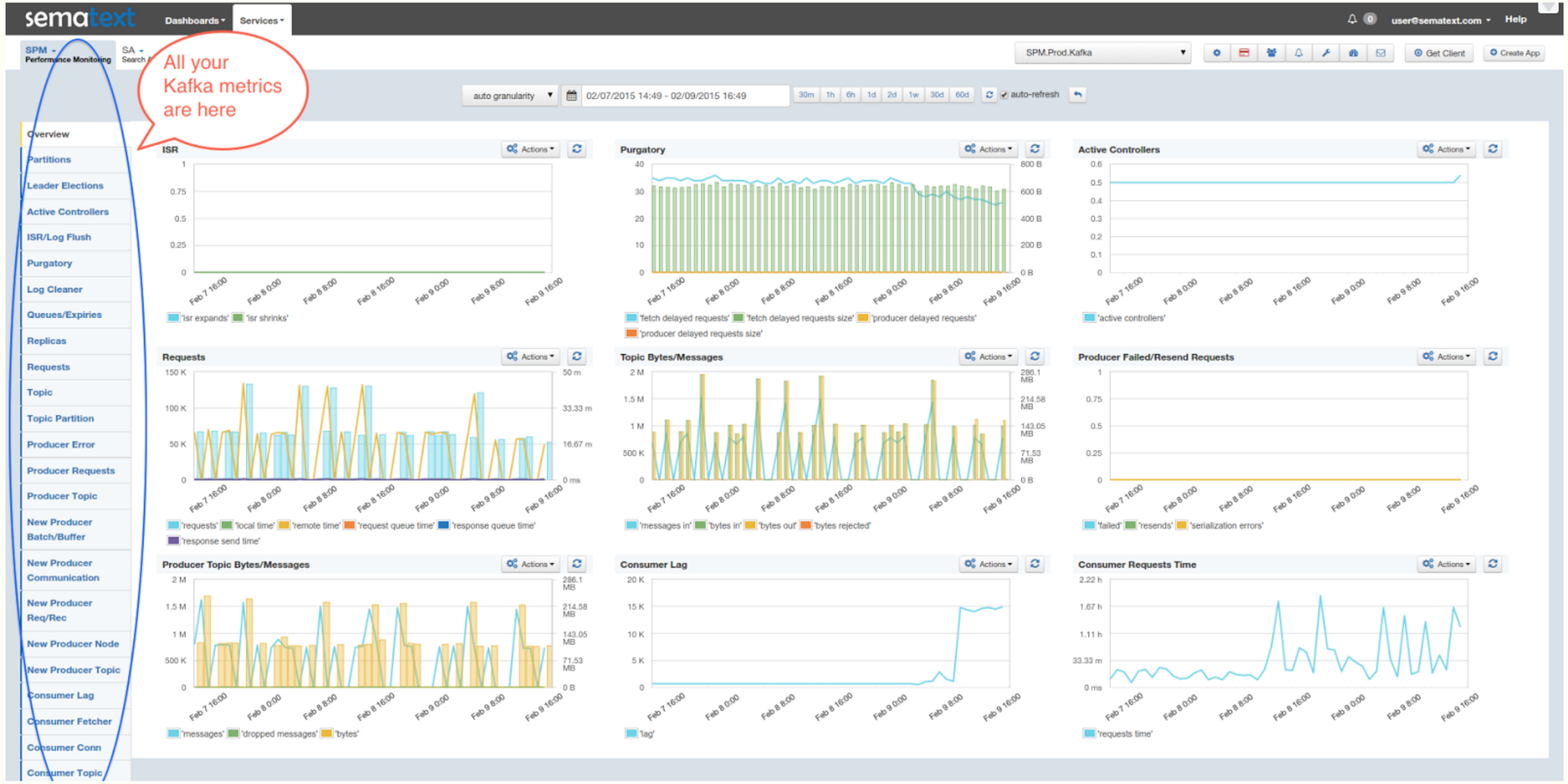
Message Condition



Response time



_ WHAT'S NEW?

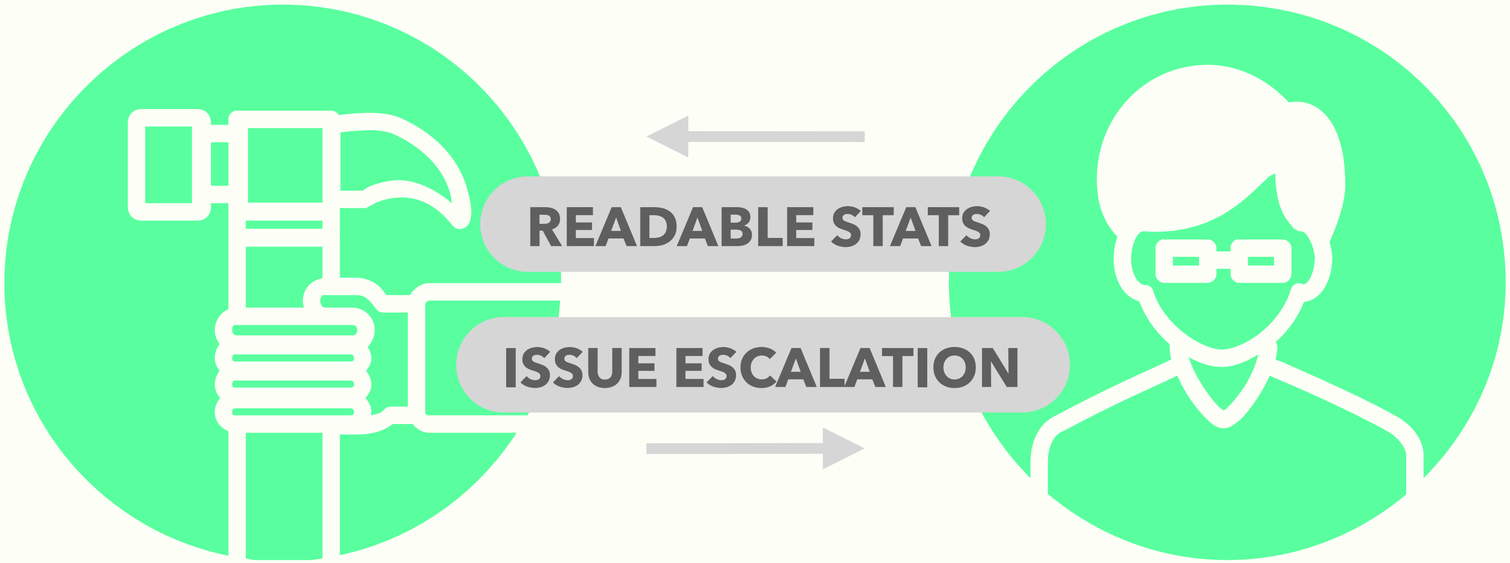


_ WHAT'S NEW?

Clear division of monitoring task

Further implication to BM

_ USER SCENARIO



client company

technical support

_ SPRINT 1 to 4

Sprint #1

Kafka Environment Setup

Sprint #2

Define Metrics

Flamingo Environment Setup

_ SPRINT 1 to 4

Sprint #3

Tried sample app, but dropped it

Sencha study

_ SPRINT #4

US#1 : As a developer, I can easily plug-in MBean for visualization

- ➔ Build General MBean Client Factory (Youngjae Chang)
- ➔ Find appropriate D3 chart design for charts (Jaryong Lee)
- ➔ Study websocket structure (Seunghyo Kang)
- ➔ Design Database schema for saving metric history (Youngjae Chang)
- ➔ Define API interface for data communication & update (Jaryong Lee)
- ➔ Design websocket communication structure (Seunghyo Kang)

US#2 : As a user, I can monitor Kafka Ecosystem

- ➔ Plug-In Kafka MBeans into Interfaces (Youngjae Chang)
- ➔ Place charts to fit designated Kafka monitoring module (Jaryong Lee)

_ New Architecture

Function	Flamingo	Our Stack
Collect	QuartzJob	JMXTrans
Store	MYSQL	Graphite (RRD Database)
Update	Ajax Query	Websocket
Draw	Sencha	D3.js

END