



Operational Data Processing Pipeline

BoF: Operational Data Analytics

Keiji Yamamoto

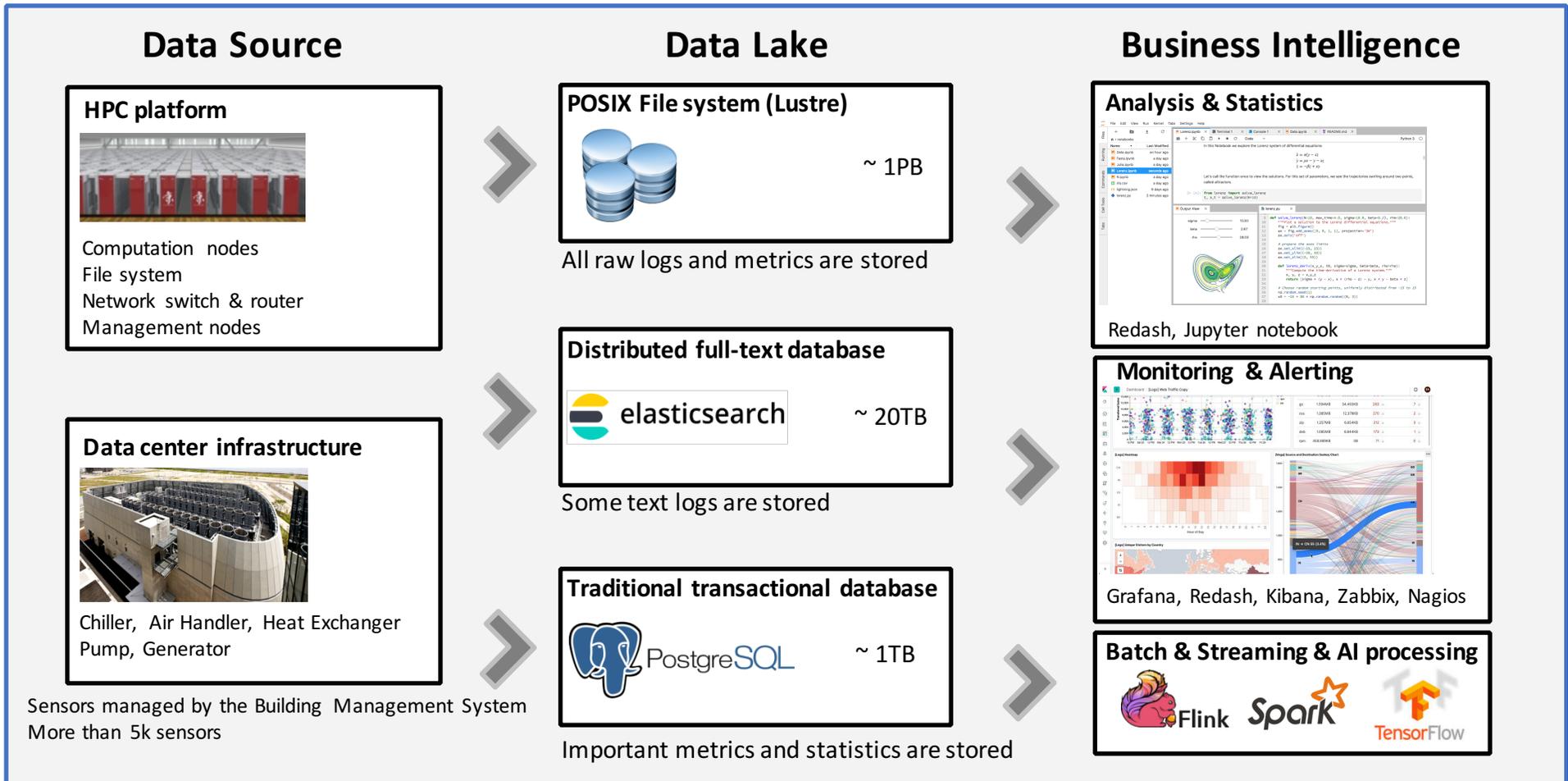
Operations and Computer Technologies Division

RIKEN R-CCS, Japan



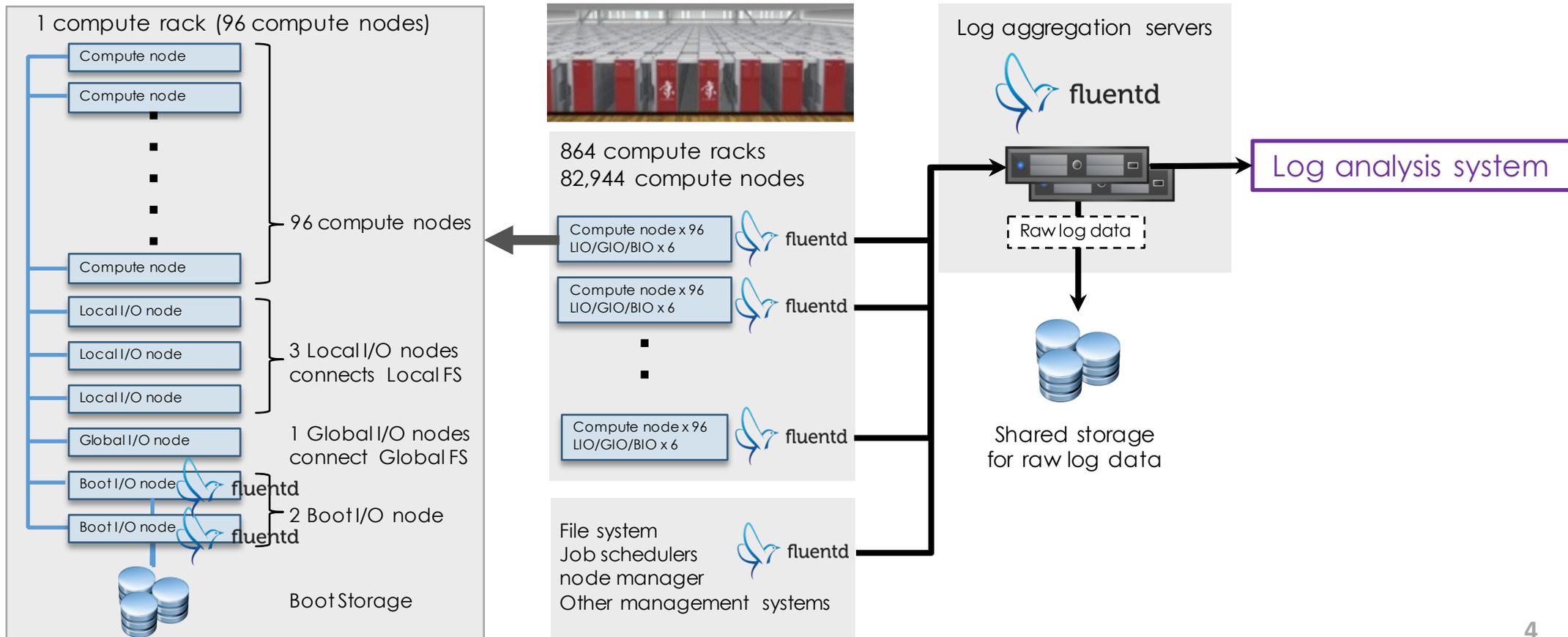
Collection and Analysis of Operational Data

- Data collection from systems and analysis is important
 - Failure detection, failure analysis
 - Anomaly detection
 - Give statistical data to improve operations
 - Many others
- We would like to efficiently collect **log** and **metric** in large-scale systems in a standardized way and analyze on these data
 - e.g.) **K** has over 80K nodes and **Fugaku** will have over 150K nodes



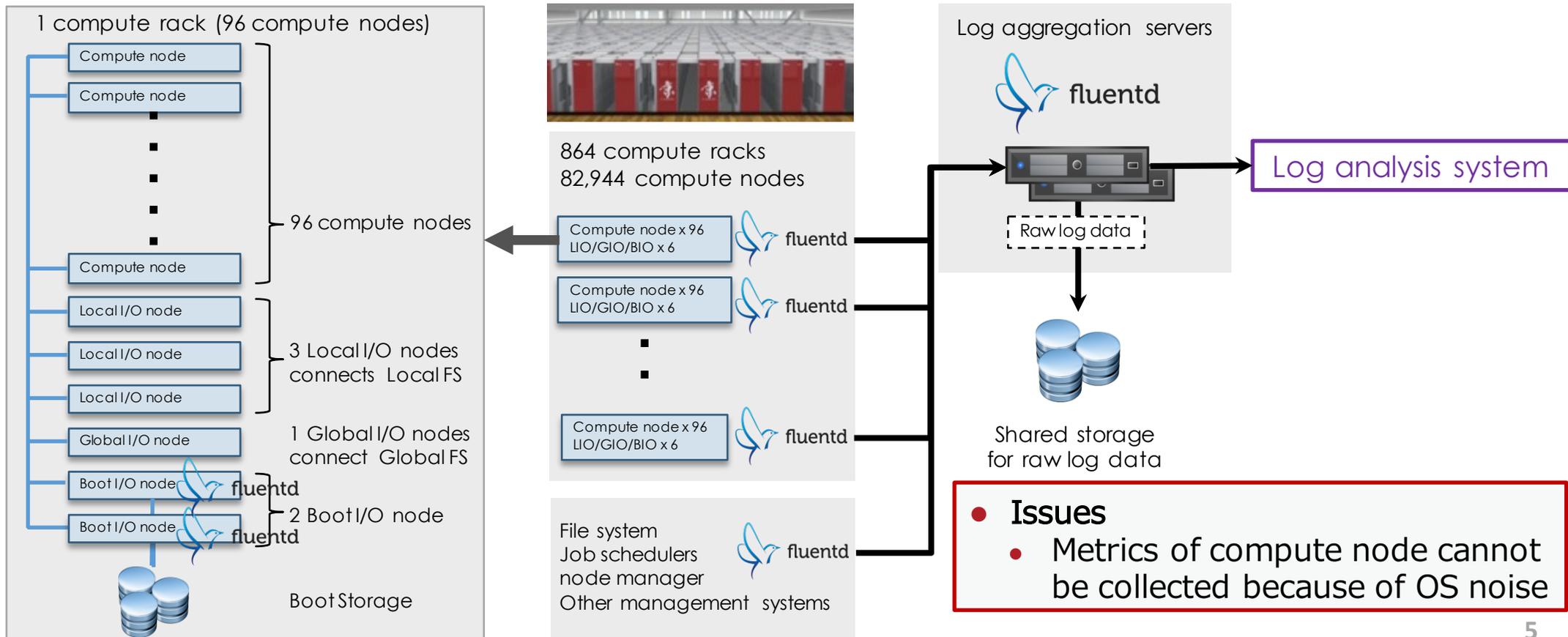
From data source to data lake: HPC Platform

- Boot I/O node is a NFS server and a fluentd agent
- Compute node, LIO and GIO nodes mount the boot storage
 - All logs are written in the boot storage

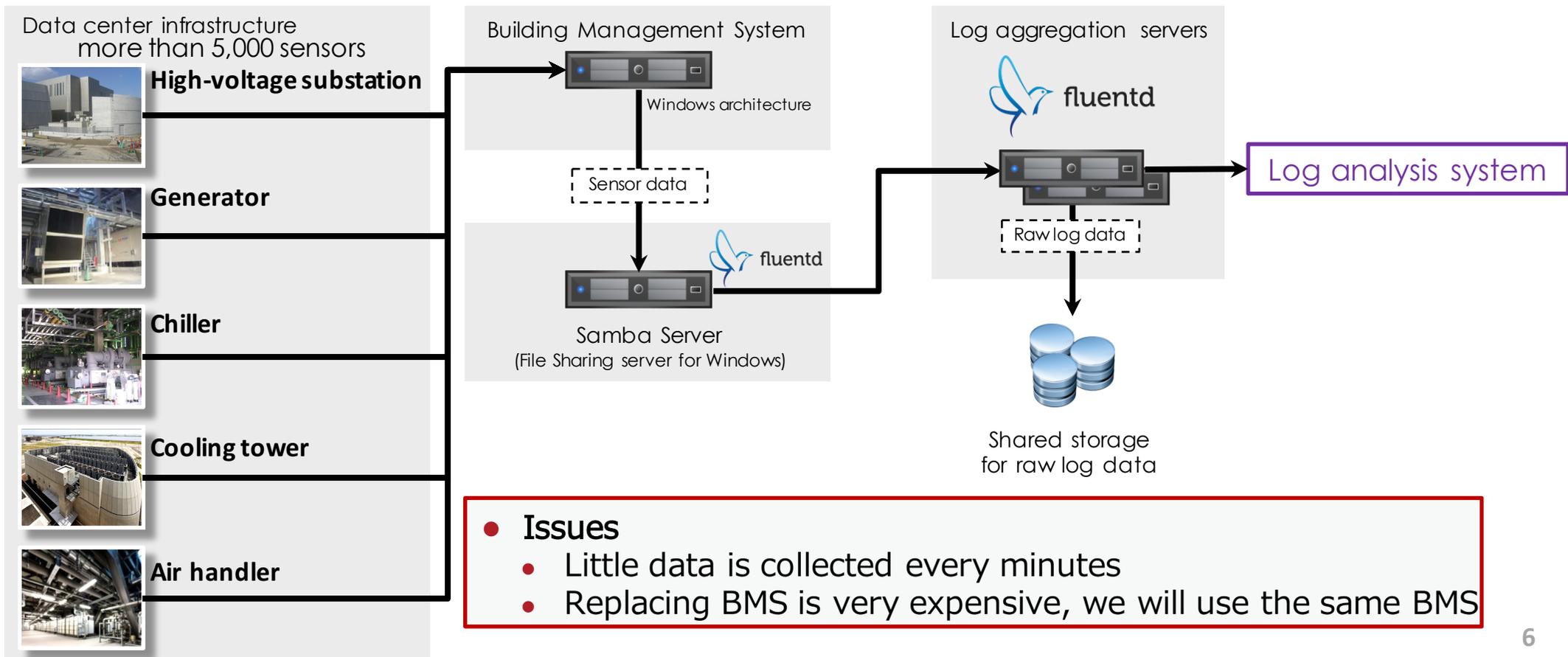


From data source to data lake: HPC Platform

- fluentd is an open source data collector, which helps to unify the data collection and consumption for data analytics
- fluentd retrieves input data from specified location, formats the data and output data to specified location
 - c.f.) Logstash, rsyslogd



- Data center infrastructures are managed by Building Management System (BMS)
 - BMS outputs some sensors (~30) data to CSV file every minutes
 - BMS outputs all sensors data daily



- **Issues**
 - Little data is collected every minutes
 - Replacing BMS is very expensive, we will use the same BMS

Log analysis system

- Data flow control (Kafka)

- Use Kafka for buffering streaming data coming from log collection system
- Control data flow for data processing and databases

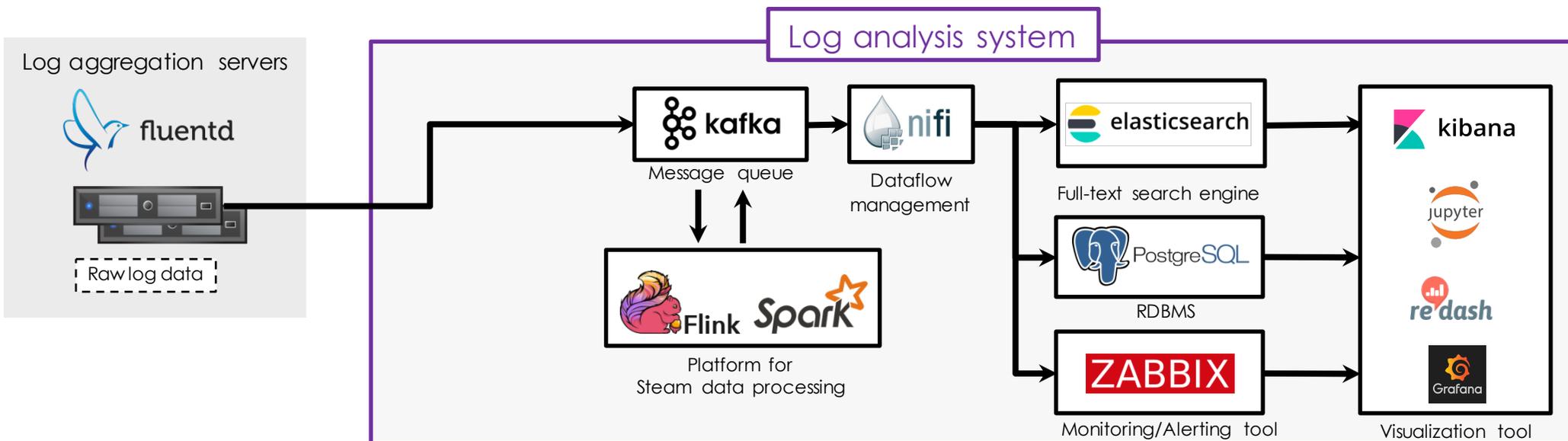
- Platform for stream data processing (Flink, Spark)

- Analyze streaming data in real time and then compute job workloads to detect overloaded nodes

- Data flow management (Nifi)

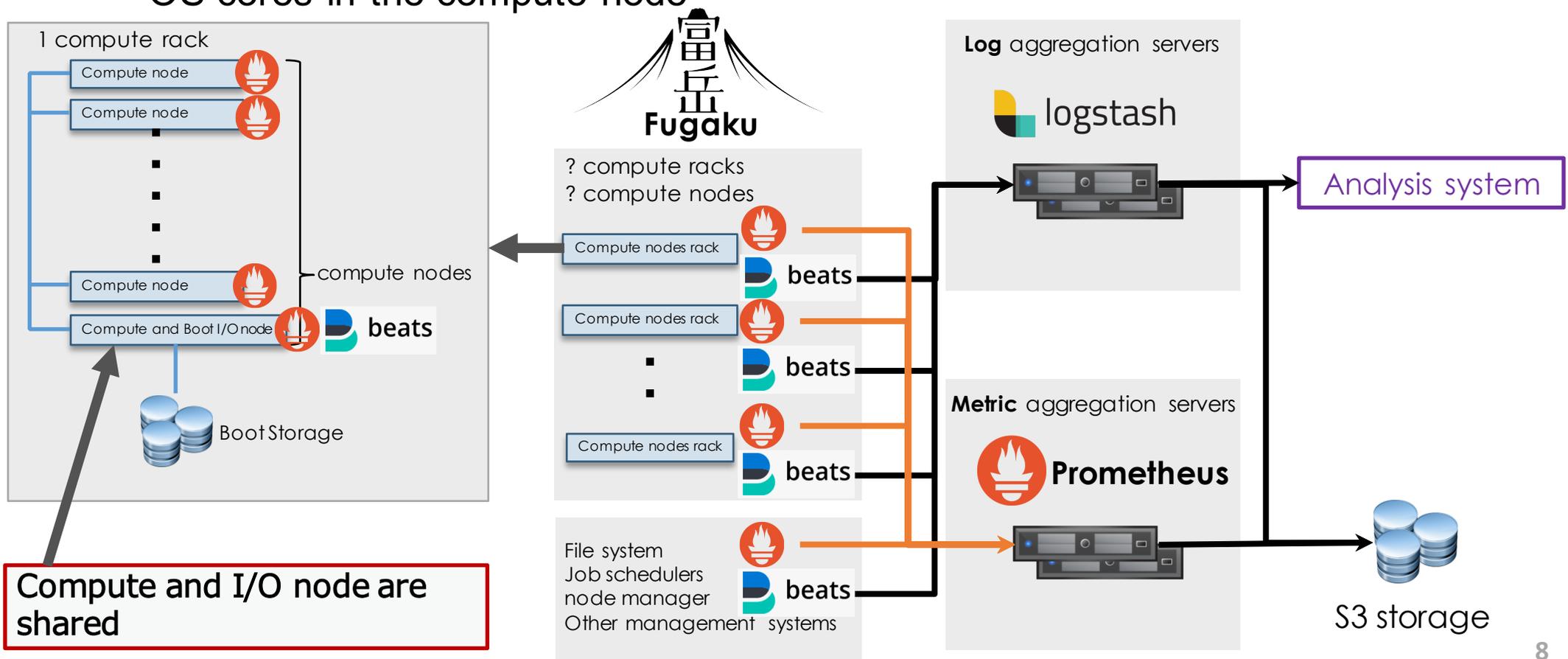
- Route data to databases

- Database (Elasticsearch, PostgreSQL)
 - Store analysis data to persistent DB
- Monitoring tool (ZABBIX)
 - Monitor system status
- Visualization (kibana, redash)



Log and metric collection system For Fugaku

- **A64FX: 48 compute cores + 2 or 4 assistant (OS) cores**
 - A lightweight metrics and logs collection process can be executed on the OS cores in the compute node



Log and metric collection system For Fugaku



- **Fluentd is replaced to Filebeat and Logstash**
 - Fluentd is written by ruby lang.
 - Memory usage is large
 - On compute node, daemon are required to consume less memory
 - Filebeat is a lightweight log shipper written by go lang.



Prometheus

- Collect compute node's metrics
 - Memory usage, CPU usage, I/O transfer bytes, Power consumption (maybe)
 - Prometheus is high throughput event monitoring and alerting software
 - In our preliminary evaluation, prometheus was able to collect 1M metrics per sec.



S3 storage

- **Store data to S3 instead of POSIX file system**
 - On the K computer, we stored data to Global File System (Lustre)
 - K account required for analysis
 - Visualization tools had to mount the Global File System
 - File system is affected by K maintenance
 - Cannot access data after the shut down of the K computer
 - Now we are moving logs from K to S3 storage

- Data flow control (Kafka)

- Use Kafka for buffering streaming data coming from log collection system
- Control data flow for data processing and databases

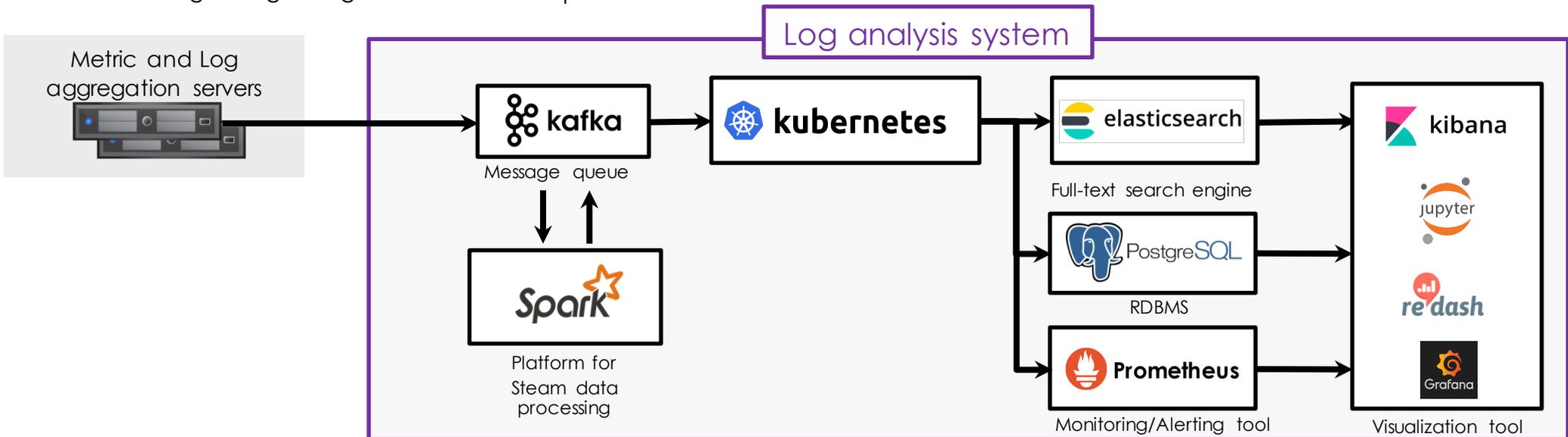
- Platform for stream data processing (Spark)

- Analyze streaming data in real time and then compute job workloads to detect overloaded nodes

- Container orchestration tool (Kubernetes)

- Manages a lightweight daemon that outputs data to a database

- Database (Elasticsearch, PostgreSQL)
 - Store analysis data to persistent DB
- Monitoring tool (Prometheus)
 - Monitor system status
- Visualization (kibana, redash)





- **NiFi is replaced to kubernetes**
 - We used NiFi only to insert records into the DB.
 - In general, streaming processing is executed by a daemon process.
 - Various daemons are required to parse various log formats and store them in DB
 - Management costs of daemon are high
 - Since NiFi is configured with a GUI, the management cost of handling many logs and streams has increased.
 - We will use Kubernetes for manage various daemon