The concept of user services on Fugaku

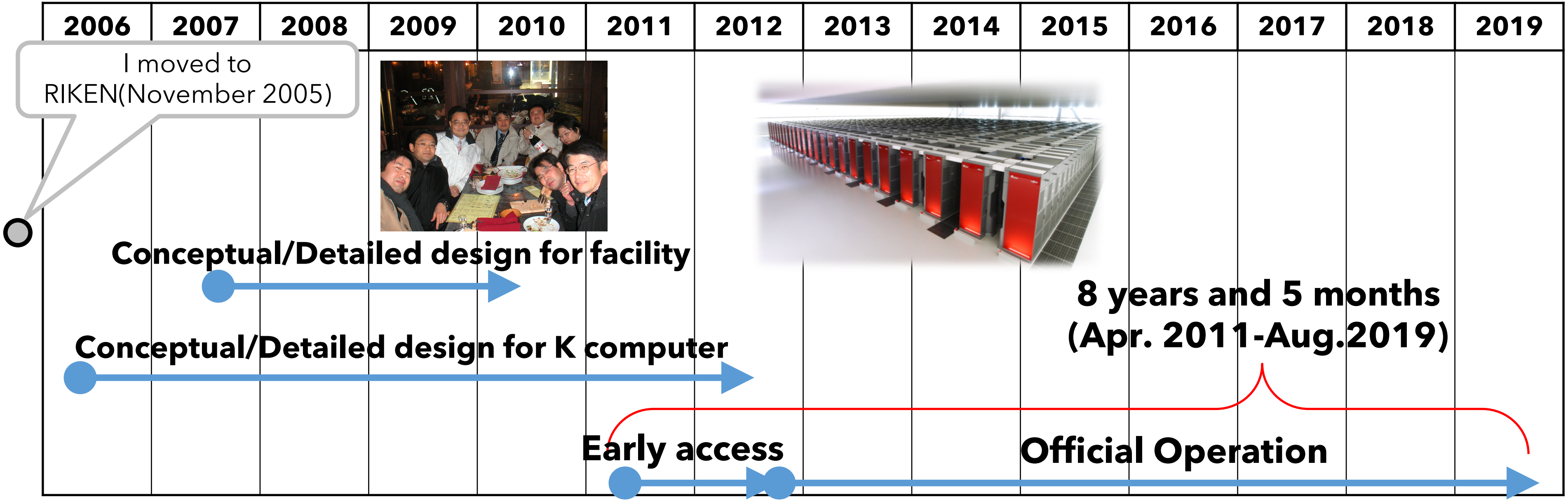
Fumiyoshi Shoji

Operations and Computer Technologies Div., R-CCS, RIKEN

@ 2nd R-CCS international symposium

February 17, 2019

K computer retired Aug.2019

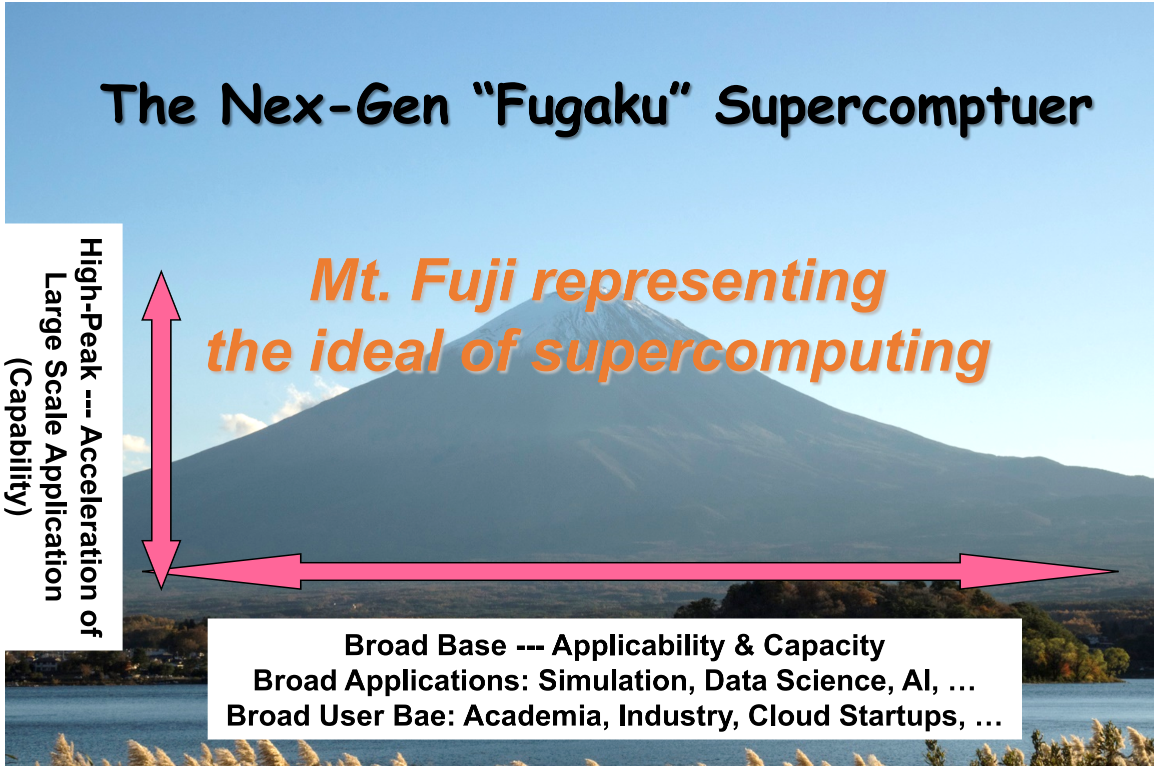

2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
I moved to RIKEN(November 2005)
Conceptual/Detailed design for facility
Conceptual/Detailed design for K computer
8 years and 5 months (Apr. 2011-Aug.2019)
Early access
Official Operation

- **Achievements:**
  - **TOP500 #1** **x 2**
  - **Graph500 #1** **x 10**
  - **HPCG #1** **x 3**
  - **Gordon Bell prize winner** **x 2**

Operation/service stats of K computer

Service duration	2,513 days 9 hours (Sep. 28th, 2012 – Aug. 16th, 2019)
# of job	4,178,431
Node x time delivered	3,637,258,658
Average job filling rate	75.6%
System availability (for the service duration/for planned service node time)	93.6/97.3%
# of user (cumulative/no double counting)	11,095/2,631
# of project (cumulative)	1,015

Fugaku – successor of K computer -



The Nex-Gen "Fugaku" Supercomptuer

Mt. Fuji representing the ideal of supercomputing

High-Peak --- Acceleration of Large Scale Application (Capability)

Broad Base --- Applicability & Capacity
Broad Applications: Simulation, Data Science, AI, …
Broad User Bae: Academia, Industry, Cloud Startups, …

ふがく
富岳

Presentation by Satoshi Matsuoka @EEHPC SOP Workshop 2019

https://sites.google.com/view/eehpcsop2019/

Action

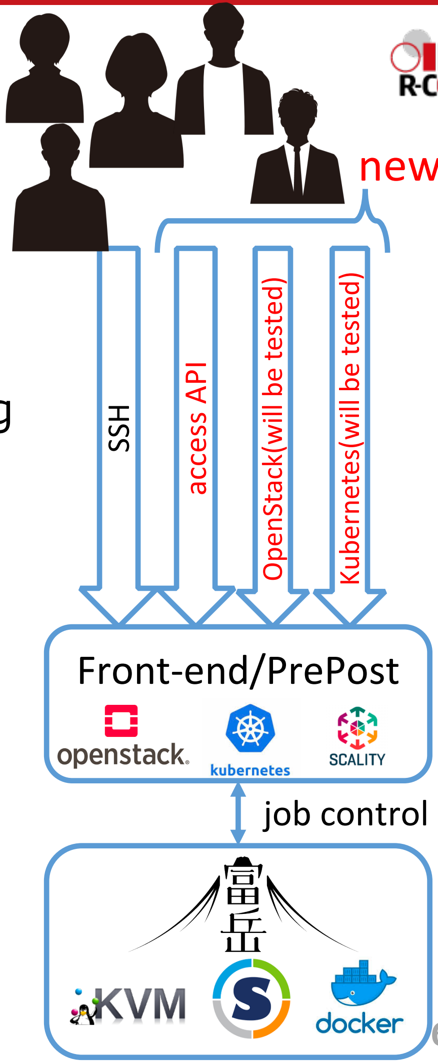
- **Improving usability**
  - accessibility
  - open source software deployment
  - data science platform

- **Improving efficiency**
  - Pre/Post I/O
  - node allocation
  - checkpoint/restart
  - power knob by user and admin

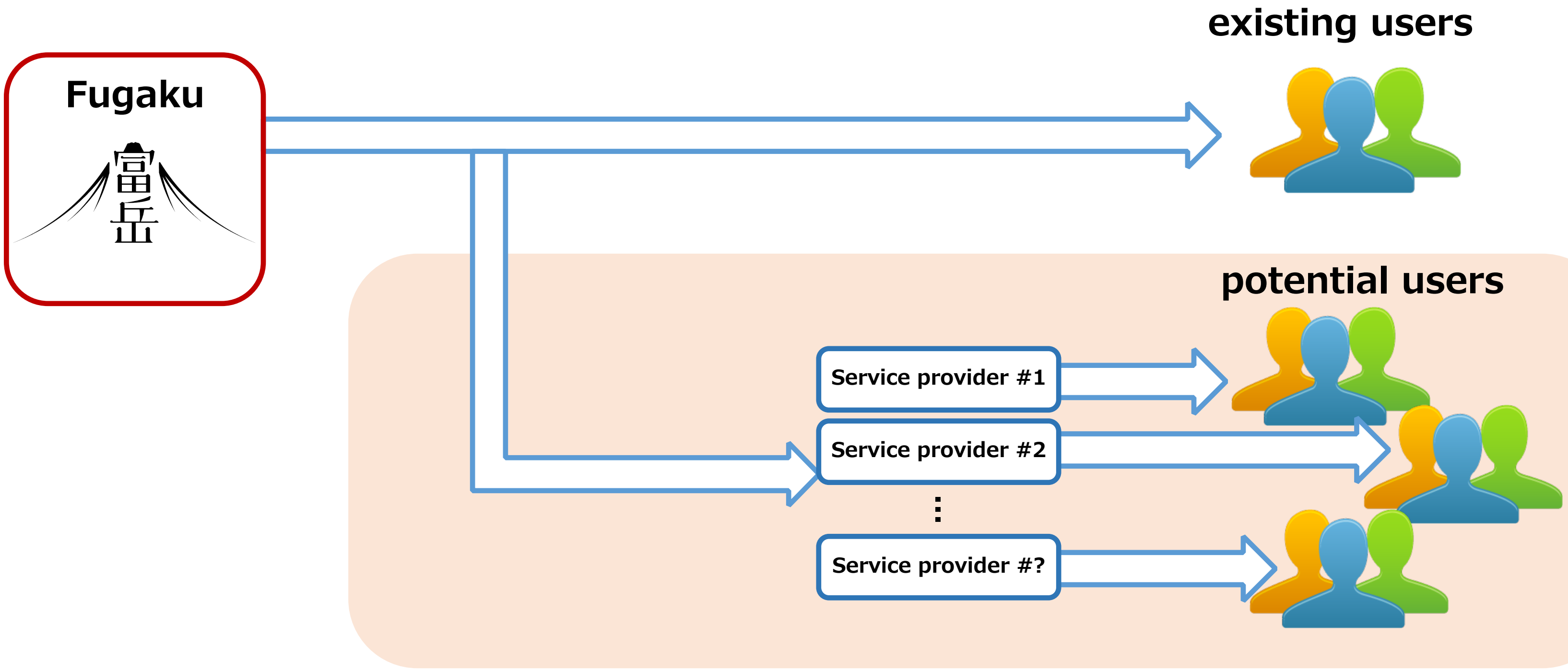
Accessibility

- **Compute nodes**
  - Jobs can be executed via Fujitsu batch job scheduler
    - CUI and access API(NEWT2.0 based) are available
    - interactive use is also available under batch job scheduling
  - KVM and Singularity will be tested

- **Front-end/PrePost environment**
  - Multi architecture based
    - x86(w/ GPU), arm TX2(w/ GPU), A64FX(48 nodes)
    - interactive/batch/OpenStack/Kubernetes (will be tested)
  - Amazon S3 compatible object storage (under procurement)



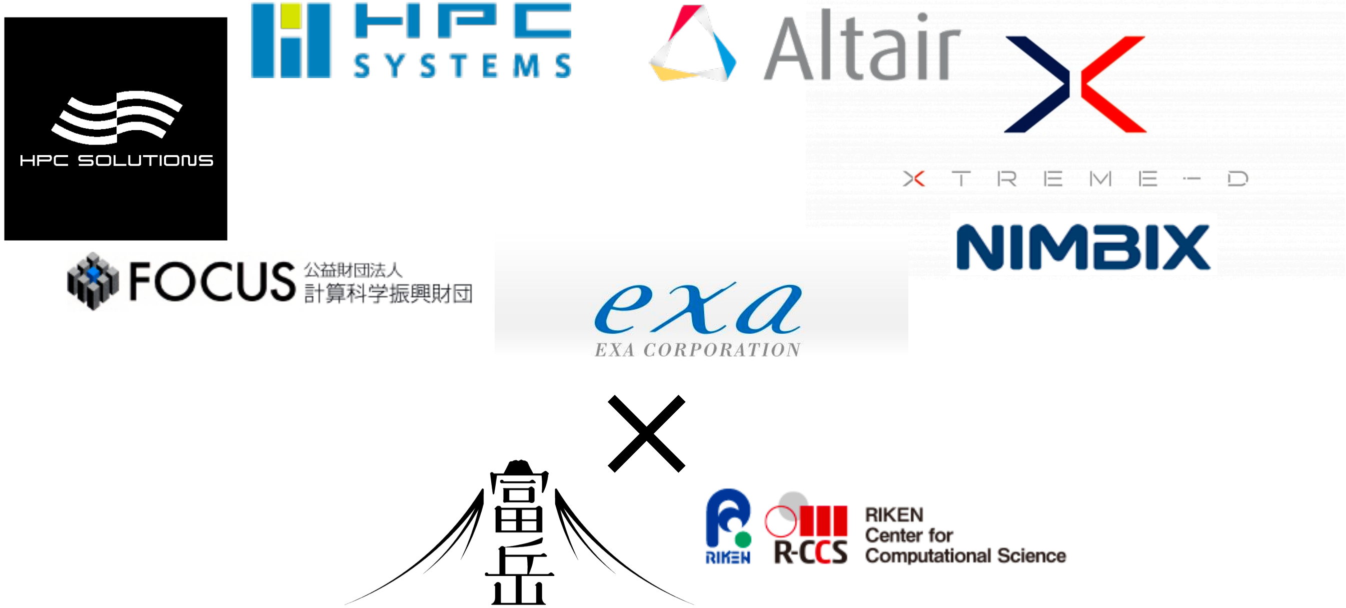
Collaboration with commercial service providers



Collaborating with service providers, we can provide more flexible service for wider field of science and engineering users

Collaboration partners selected

https://www.r-ccs.riken.jp/library/topics/200213.html (in Japanese)

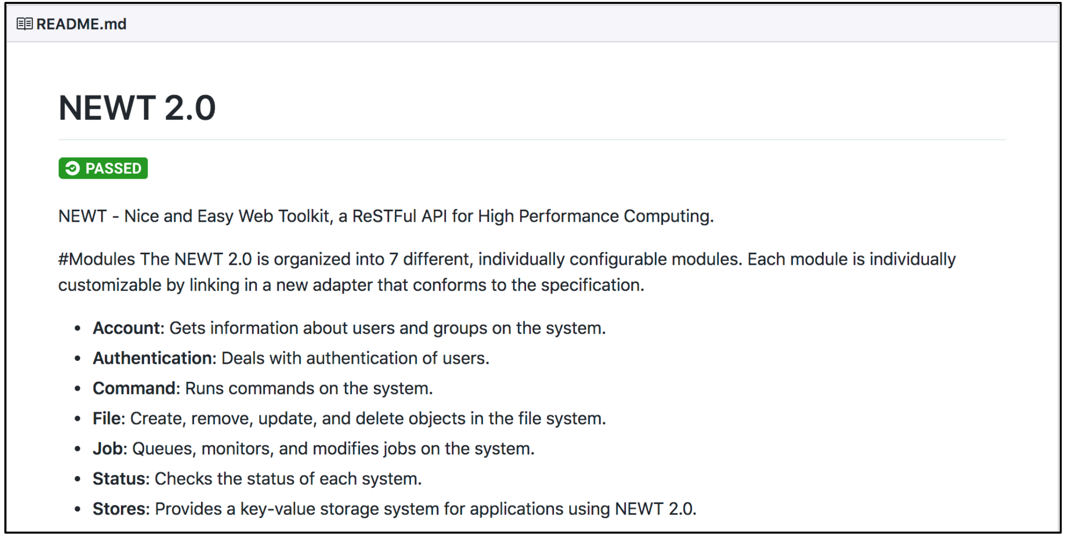


Action

- Cool Project name and logo!
- Trial methods to provide computing resources of Fugaku to end-users via service providers
- Evaluate the effectiveness of the methods quantitatively as possible and organize the issues
- The knowledges gained will be feedbacked to scheme design of Fugaku by the government

Access API

- We employed NEWT2.0 as a prototype of access API of Fugaku



README.md

NEWT 2.0

PASSED

NEWT - Nice and Easy Web Toolkit, a ReSTFul API for High Performance Computing.

#Modules The NEWT 2.0 is organized into 7 different, individually configurable modules. Each module is individually customizable by linking in a new adapter that conforms to the specification.

- **Account**: Gets information about users and groups on the system.
- **Authentication**: Deals with authentication of users.
- **Command**: Runs commands on the system.
- **File**: Create, remove, update, and delete objects in the file system.
- **Job**: Queues, monitors, and modifies jobs on the system.
- **Status**: Checks the status of each system.
- **Stores**: Provides a key-value storage system for applications using NEWT 2.0.

- We will discuss standardization of API with HPC centers/providers
- An implementation of the API on Fugaku will be available August 2020

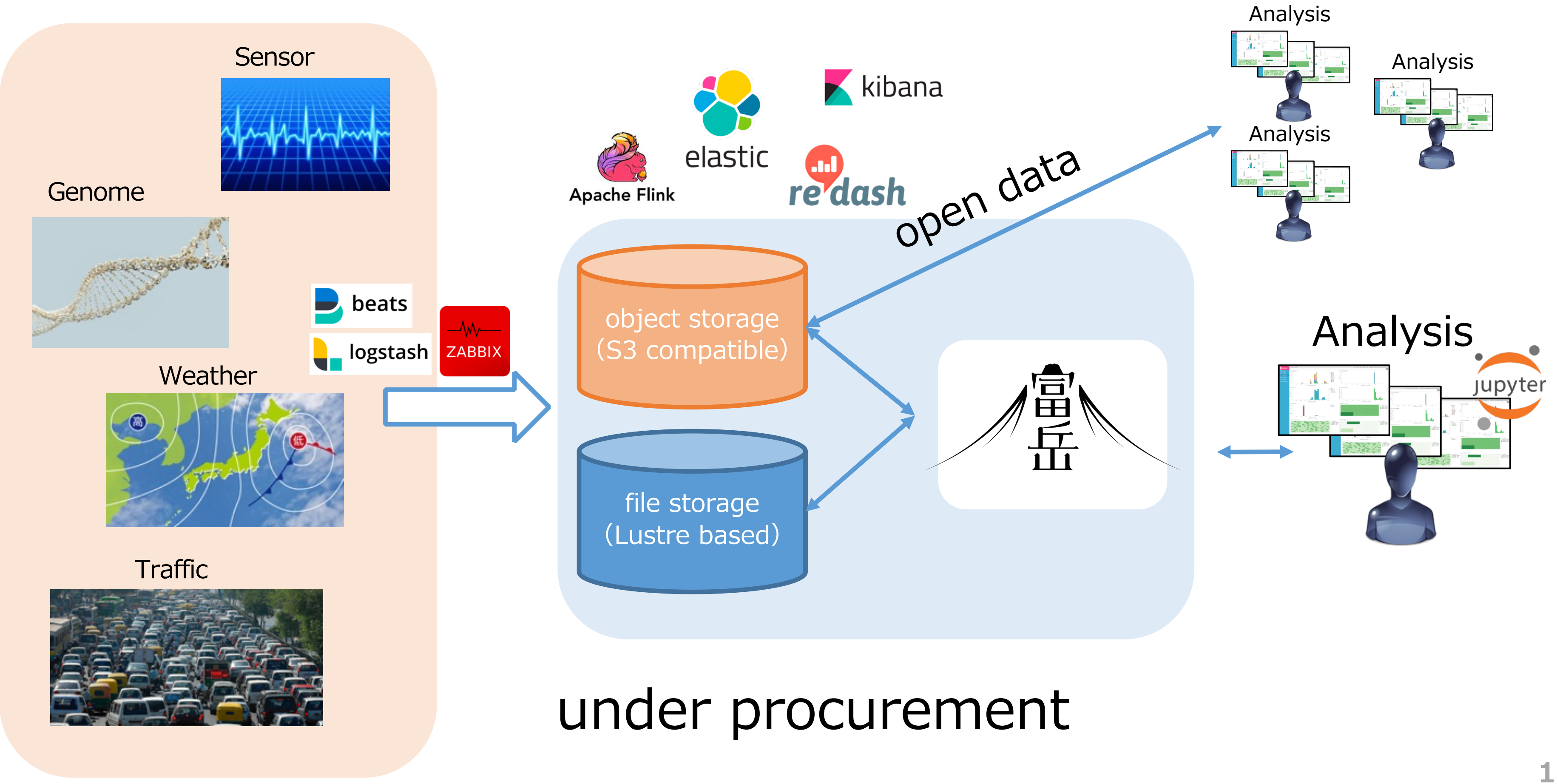
https://github.com/NERSC/newt-2.0

Open source software for Fugaku

- For K computer
  - Due to special ISA (Sparc based), there was no software eco-system…

- For Fugaku
  - Activities for open source software on arm ISA
    - Arm HPC Users Group https://arm-hpc.gitlab.io/
    - Linaro https://www.linaro.org/
    - Spack: https://spack.io/
      - Official software package manager of the Exascale Computing Project
  - R-CCS Software Center https://www.r-ccs.riken.jp/software_center/
    - Activity in R-CCS to develop, deploy and promote high quality applications, libraries, programming tools, etc. make in R-CCS for many HPC platforms including Fugaku.
  - DL4Fugaku https://github.com/dl4fugaku/dl4fugaku/wiki
    - R-CCS & Fujitsu collaboration for Deep learning framework on Fugaku
      - Target: PyTorch, TensorFlow, Chainer, etc.

Data science platform



Sensor

Genome

Weather

Traffic

beats

logstash

ZABBIX

Apache Flink

elastic

kibana

redash

open data

object storage
(S3 compatible)

file storage
(Lustre based)

富岳

Analysis

Analysis

Analysis

Analysis

jupyter

under procurement

Action

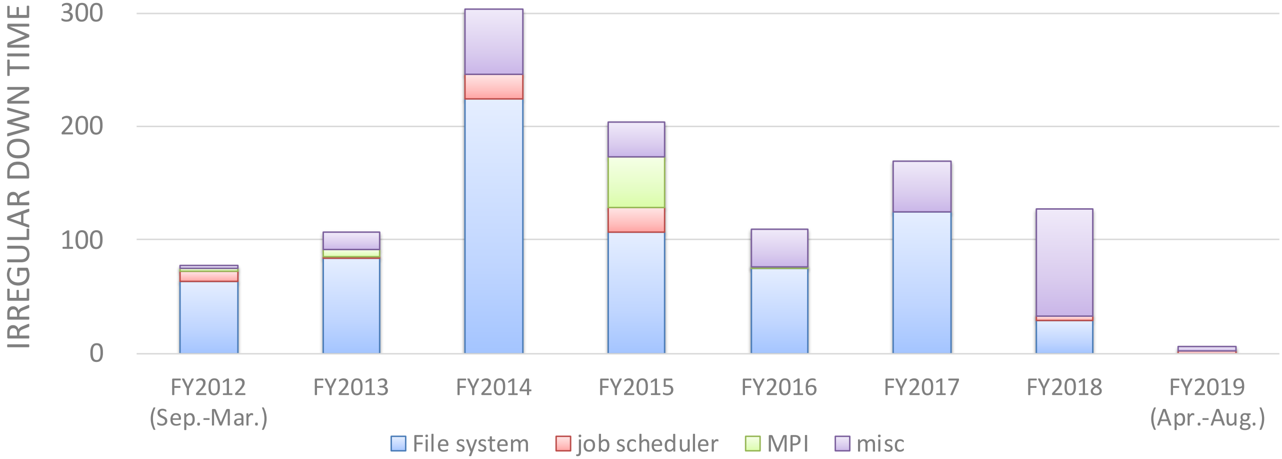
- **Improving usability**
  - accessibility
  - open source software deployment
  - data science platform

- **Improving efficiency**
  - Pre/Post I/O
  - node allocation
  - checkpoint/restart
  - power knob by user and admin

Sharing pain for efficiency

- **Average job filling rate : 75.6% (= node allocation loss : 24.4%)**

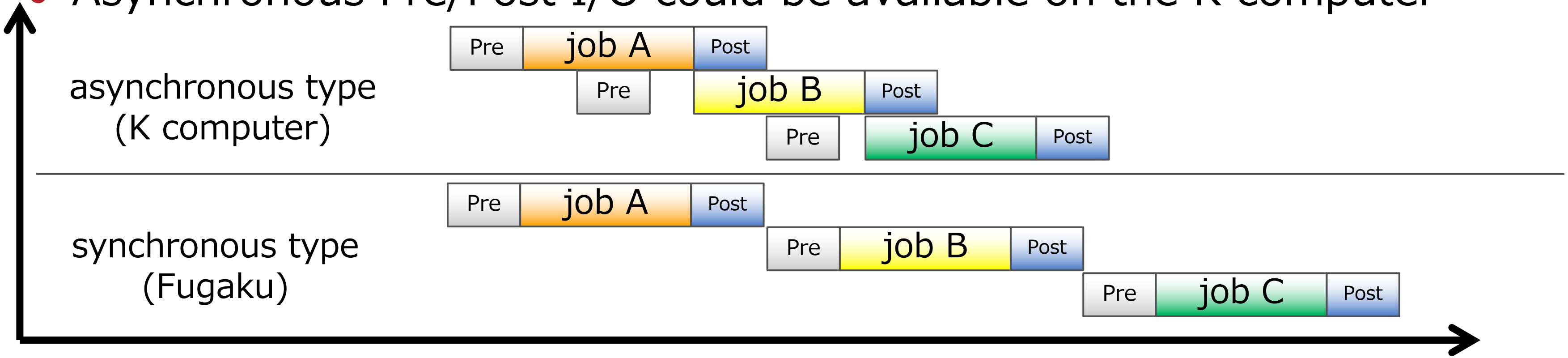
1. A complicated Pre/Post I/O implementation and operation rule



2. An inefficient node allocation rule (2-3%)
3. Resource compensation rule for system failure (1-2%)

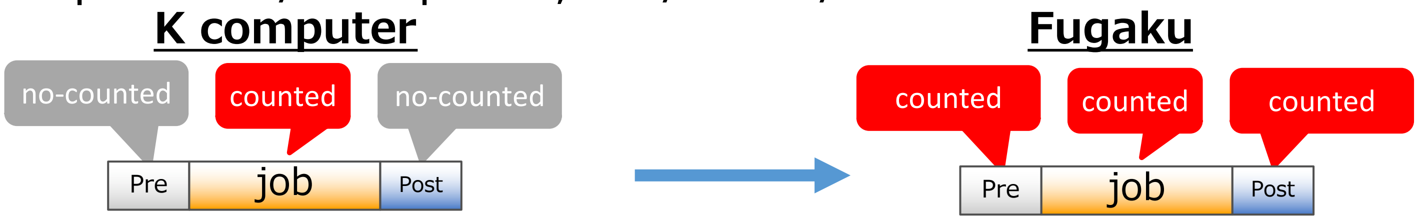
Sharing pain for efficiency (Pre/Post I/O)

- Pre/Post I/O
  - Asynchronous Pre/Post I/O could be available on the K computer



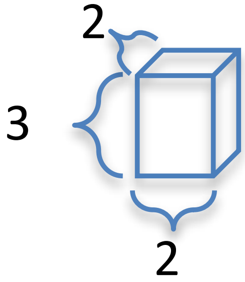
An asynchronous Pre/Post I/O was much more difficult to implement and its complexity might induce many serious bugs in system software. → We adopt a synchronous type for Fugaku

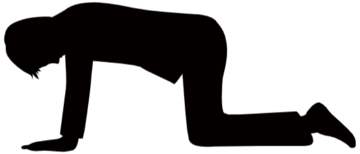
  - To optimize I/O requests, Pre/Post I/O will be counted as user time



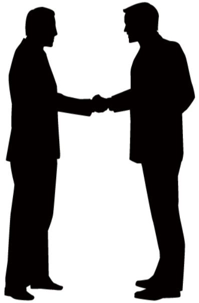
Sharing pain for efficiency (node allocation)

- **K computer**
  - A block-wise (≠distributed) node allocation policy due to a direct connection network topology
  - node allocation unit is 2x3x2 = 12 nodes
  - User can run a job with any node size (even not a multiple of 12 nodes)
  - → node allocation loss
    - → by the gap between user request and system assigned
    - → by scheduling difficulty



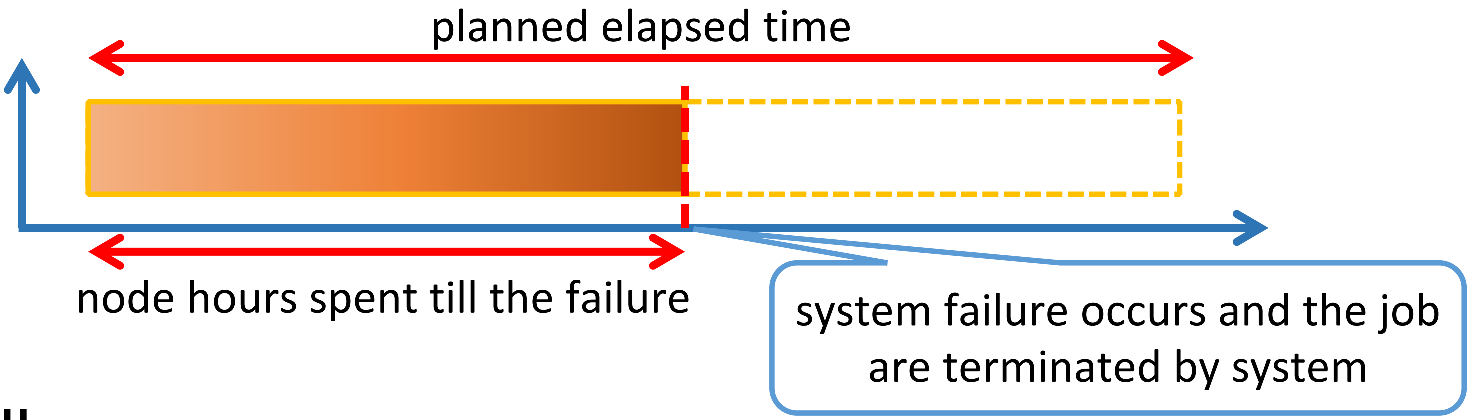


- **Fugaku**
  - A block-wise policy and node allocation unit is 2x3x8 = 48 nodes
  - User can choose node size in a multiple of 2x3x8 (job with more than 48 node case)



Sharing pain for efficiency (checkpoint/restart)

- **K computer**
  - Node hours lost by system failure was compensated.



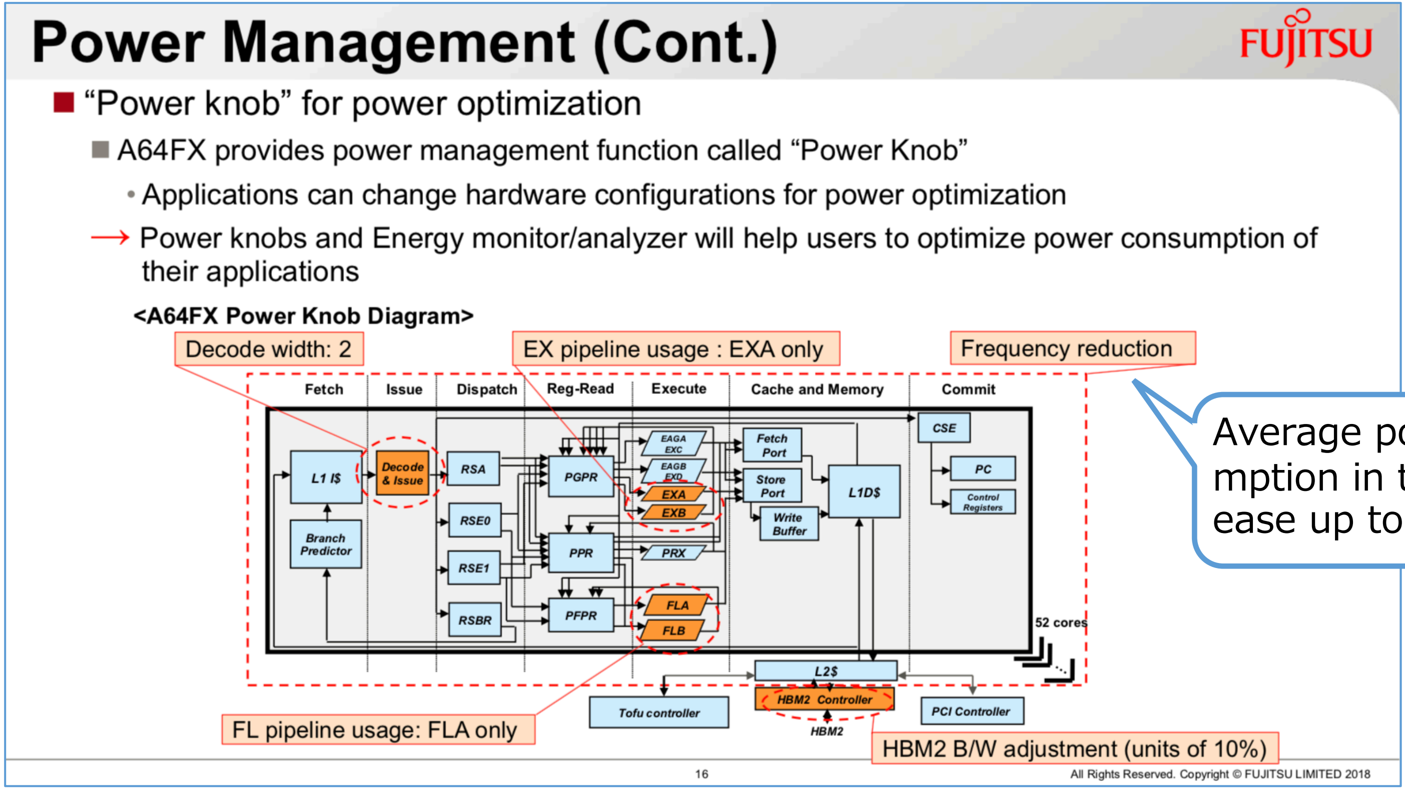
system failure occurs and the job are terminated by system

- **Fugaku**
  - An user level checkpoint/restart tools (e.g. ECP-VeloC/VELOC) will be available on Fugaku
    - https://github.com/ECP-VeloC/VELOC

→ It’s time to finish resource compensation for system failure…

New functions of Fugaku for energy saving

Fujitsu’s presentation @ Hot Chips30 https://www.fujitsu.com/jp/Images/20180821hotchips30.pdf



Power Management (Cont.)

FUJITSU

- “Power knob” for power optimization
  - A64FX provides power management function called “Power Knob”
    - Applications can change hardware configurations for power optimization

→ Power knobs and Energy monitor/analyzer will help users to optimize power consumption of their applications

<A64FX Power Knob Diagram>

Decode width: 2

EX pipeline usage : EXA only

Frequency reduction

FL pipeline usage: FLA only

HBM2 B/W adjustment (units of 10%)

All Rights Reserved. Copyright © FUJITSU LIMITED 2018

Average power consumption in the idle decrease up to 5-10%

User can use the power knob via Power API

How can we motivate users for energy saving?

- **Which policy is better?**
  1. all power knobs are turn off at default (start from minimum saving)
     - admin finds out jobs that are wasting power from profiling data
     - admin requests user to turn on the knob
     - Pros : Less user complaints
     - Cons : Less energy saving
  2. all power knobs are turn on at default (start from maximum saving)
     - user shows to admin that using the knob reduces (keeps) energy-to-solution for his/her job by trial
     - admin allow the user to turn off the knob
     - Pros : More energy saving
     - Cons : More user complaints

How can we motivate users for energy saving? (cont'd)

- **Grant incentives depending on the contribution to the power saving**
  - additional node hours, higher priority, etc.
  - Concern: How can we fairly evaluate "contributions" for energy saving ("as-is" --> tuned)?

- **Change resource allocation unit**
  - node x hours -> energy (watt hour)
  - Concern: How can we keep fairness between applications which have different power profile?

Summary

- **Easy to use**
  - accessibility by collaboration with commercial service providers
  - open source software deployment by Spack
  - data science platform by object/file storages with analysis env.

- **Sharing pain for efficiency**
  - Pre/Post I/O
  - node allocation
  - aggressive use of power knob for power saving

Thank you for your attention