FUJITSU
shaping tomorrow with you

Post-K Status and Lustre Testing Reports

Shinji Sumimoto, Ph.D.
Next Generation Technical Computing Unit
FUJITSU LIMITED May. 14th, 2019

Outline

FUJITSU

- Lustre 20 years Anniversary
- Post-K Development in Fujitsu
- Lustre Testing Reports

FUJITSU

2019: Lustre 20 year Anniversary

20 years Lustre History

FUJITSU

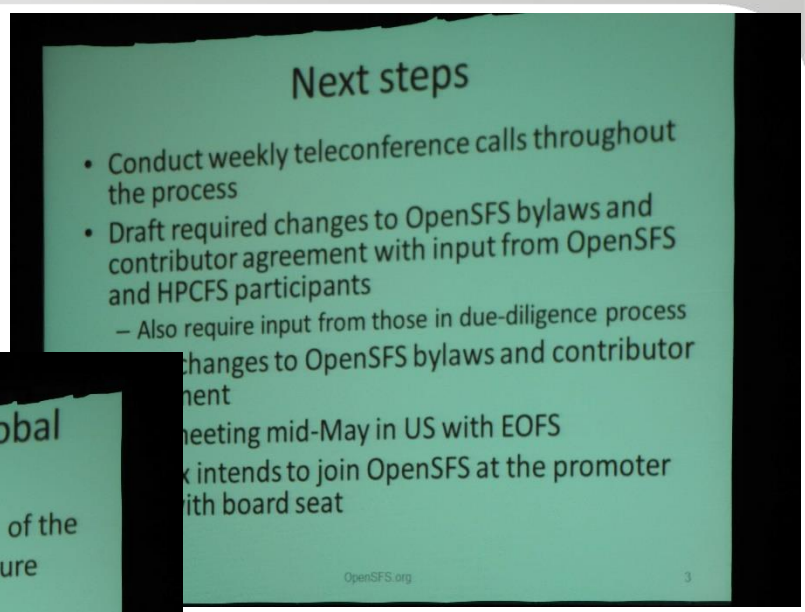
- 1999: Lustre file system Project Started by Peter J. Braam
- 2001: Cluster File Systems
- 2007/9: Sun's Acquisition of Cluster File Systems
- 2009/4: Oracle's Acquisition of Sun
- 2010/4: Oracle limited Lustre support to Oracle's Hardware
  - Three Lustre community: EOFS@EU, US DOE OpenSFS, World Wide HPCFS
- 2010/9: Establishment of Whamcloud
  - Lustre Engineer moved from Oracle to Whamcloud
- 2010/12: Oracle freezed Lustre Development
- 2011/4: OpenSFS and EOFS re-started@LUG2011
- 2012/7: Intel's Acquisition of Whamcloud
- 2018/6: DDN's Acquisition of Intel's Lustre Business, Establishment of Brand-new Whamcloud2
- 2019: Luster 20 years anniversary

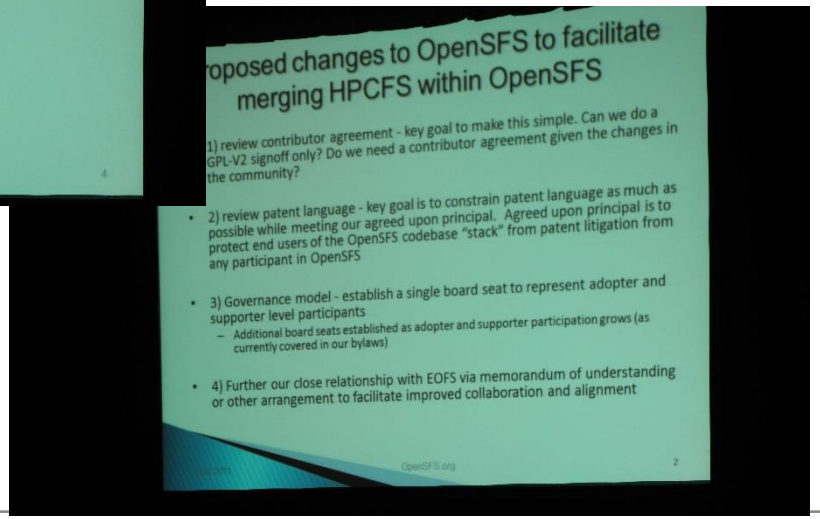
LUG2009: Lustre 10th Anniversary

FUJITSU

LUG 2011: Single Community

FUJITSU





2018/6: DDN's Acquisition of Intel's Lustre Business

FUJITSU

ISC18: Lustre BoF

FUJITSU

FUJITSU

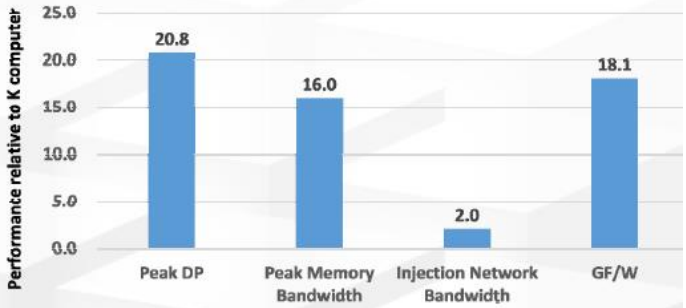
Post-K Development in Fujitsu

Post-K

- A Post-K prototype machine was built in Summer 2018. Since then, Fujitsu has been testing and evaluating the machine.
- Ten racks of Post-K achieve almost the same performance of K computer (864 racks)

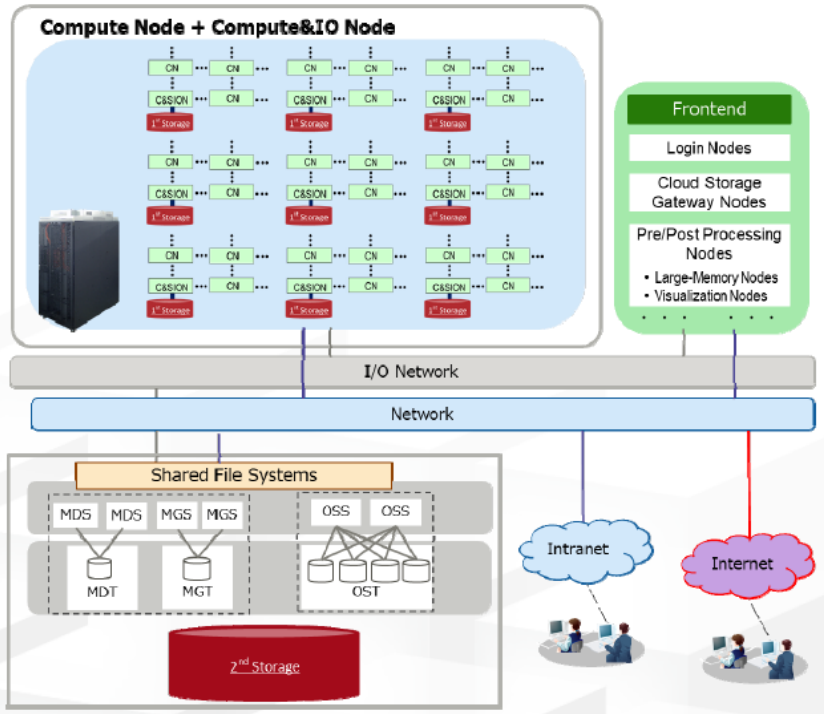
X 10 =

		Post-K	K
CPU Architecture		A64FX (Armv8.2-A SVE +Fujitsu Extension)	SPARC64 VIIIfx
Node	Cores	48	8
	Peak DP performance	2.7+ TF	0.128 TF
	Main Memory	32 GiB	16 GiB
	Peak Memory Bandwidth	1024 GB/s	64 GB/s
	Peak Network Performance	40.8 GB/s	20 GB/s
Rack	Nodes	384	102
	Peak DP performance	1+ PF	< 0.013PF
Process Technology		7 nm FinFET	45 nm



An Overview of Post-K Hardware

- **150k+ node**
- **Two types of nodes**
  - Compute Node and Compute & I/O Node connected by Fujitsu TofuD, 6D mesh/torus Interconnect
- **3-level hierarchical storage system**
  - 1st Layer
    - One of 16 compute nodes, called Compute & Storage I/O Node, has SSD about 1.6 TB
    - Services
      - Cache for global file system
      - Temporary file systems
        - Local file system for compute node
        - Shared file system for a job
  - 2nd Layer
    - Fujitsu FEFS: Lustre-based global file system
  - 3rd Layer
    - Cloud storage services



FUJITSU

Post-K Programming Environment

R-CCS

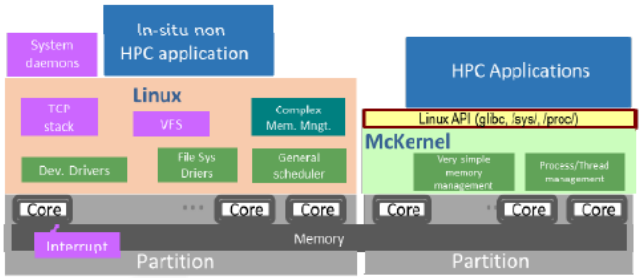
- **Programing Languages and Compilers provided by Fujitsu**
  - Fortran2008 & Fortran2018 subset
  - C11 & GNU and Clang extensions
  - C++14 & C++17 subset and GNU and Clang extensions
  - OpenMP 4.5 & OpenMP 5.0 subset
  - Java
  - GCC, LLVM, and Arm compiler will be also available
- **Parallel Programming Language & Domain Specific Library provided by RIKEN**
  - XcalableMP
  - FDPS (Framework for Developing Particle Simulator)
- **Process/Thread Library provided by RIKEN**
  - PiP (Process in Process)
- **Script Languages provided by Fujitsu**
  - E.g., Python+NumPy, SciPy
- **Communication Libraries**
  - MPI 3.1 & MPI4.0 subset
    - Fujitsu MPI (Based on Open MPI), Riken MPI (Based on MPICH)
  - Low-level Communication Libraries
    - uTofu (Fujitsu), LLC(RIKEN)
- **File I/O Libraries provided by RIKEN**
  - pnetCDF, DTF, FTAR
- **Math Libraries**
  - BLAS, LAPACK, ScaLAPACK, SSL II (Fujitsu)
  - EigenEXA, Batched BLAS (RIKEN)
- **Programming Tools provided by Fujitsu**
  - Profiler, Debugger, GUI

FUJITSU

Other Software

R-CCS

- **Batch Job System (Fujitsu)**
  - Technical Computing Suite
    - Successor of K's batch job system
- **Operating System on Compute Nodes**
  - Linux (Fujitsu)
  - McKernel, Light-weight Kernel (RIKEN)
    - Executes the same binary of Linux without any recompilation
    - One of advantages is that McKernel provides much larger page sizes
      - Applications, accessing a huge memory area randomly, may benefit
    - User may select one of McKernel configurations without rebooting
- **Other User-Land**
  - A Linux distribution
- **Open Source Management Tools**
  - Spack/EasyBuild



		McKernel Default 4K	McKernel Default 64K	Linux
.text		4K	64K	64K
.data		64K,2M,32M, 1G	2M, 512M	2M
.bss		64K,2M,32M, 1G	2M, 512M	2M
Stack		64K,2M,32M, 1G	2M, 512M	2M
malloc		64K,2M,32M, 1G	2M, 512M	2M
thread stack		64K,2M,32M, 1G	2M, 512M	2M
Shared memory	System V IPC	64K,2M,32M, 1G	2M, 512M	64K
	POSIX	4K	64K	64K
	XPMEM	64K,2M,32M, 1G	2M, 512M	64K

A64FX Chip Overview

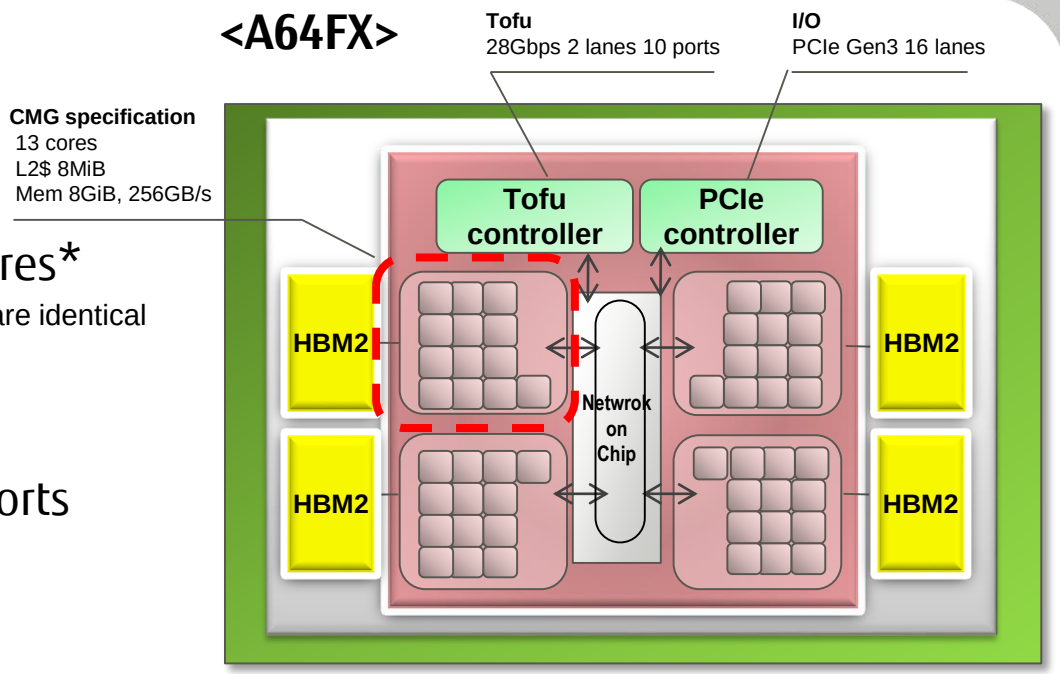
FUJITSU

Architecture Features

- Armv8.2-A (AArch64 only)
- SVE 512-bit wide SIMD
- 48 computing cores + 4 assistant cores*

*All the cores are identical

- HBM2 32GiB
- TofuD 6D Mesh/Torus
  28Gbps x 2 lanes x 10 ports
- PCIe Gen3 16 lanes



7nm FinFET

- 8,786M transistors
- 594 package signal pins

Peak Performance (Efficiency)

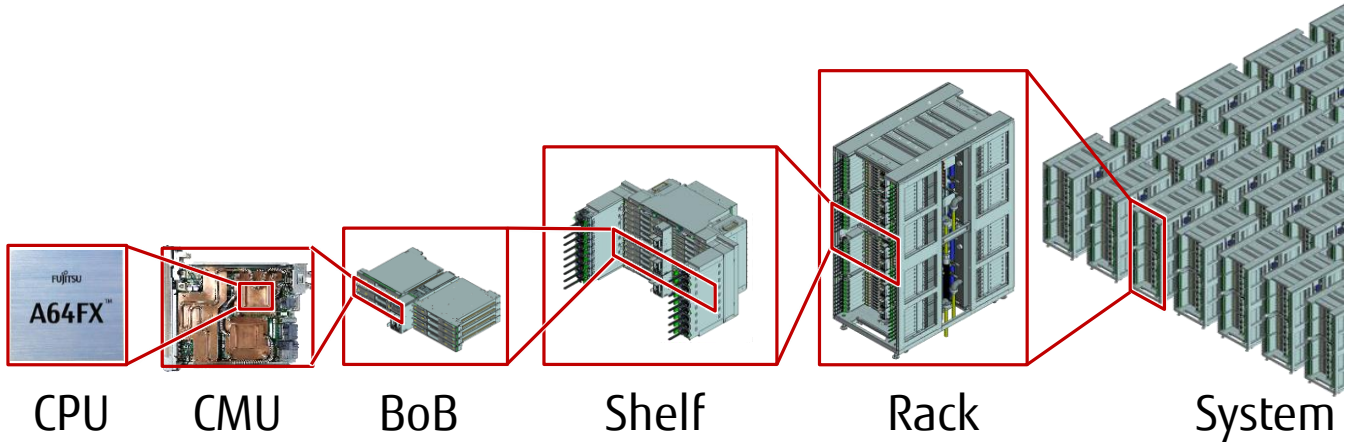
- >2.7TFLOPS (>90%@DGEMM)
- Memory B/W 1024GB/s (>80%@Stream Triad)

	A64FX (Post-K)	SPARC64 XIfx (PRIMEHPC FX100)
ISA (Base)	Armv8.2-A	SPARC-V9
ISA (Extension)	SVE	HPC-ACE2
Process Node	7nm	20nm
Peak Performance	>2.7TFLOPS	1.1TFLOPS
SIMD	512-bit	256-bit
# of Cores	48+4	32+2
Memory	HBM2	HMC
Memory Peak B/W	1024GB/s	240GB/s x2 (in/out)

Post-K system configuration

FUJITSU

- Scalable design



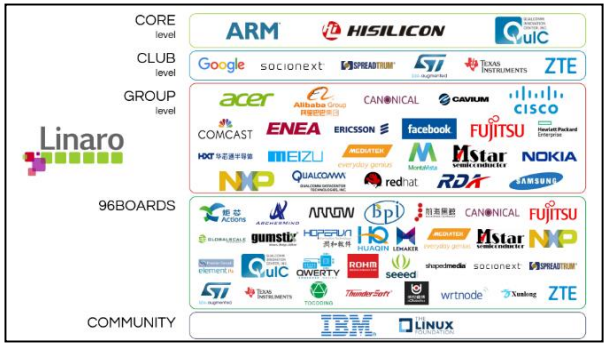
Unit	# of nodes	Description
CPU	1	Single socket node with HBM2 & Tofu interconnect D
CMU	2	CPU Memory Unit: 2x CPU
BoB	16	Bunch of Blades: 8x CMU
Shelf	48	3x BoB
Rack	384	8x Shelf
System	150k+	As a Post-K system

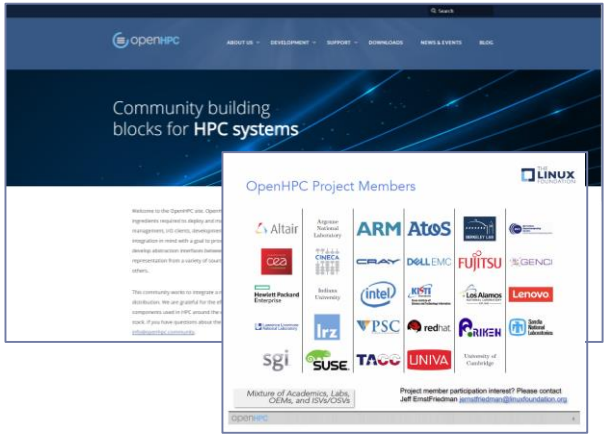
Strong Relationship with Arm HPC community

FUJITSU

- Arm
  - Great Establishment and Contribution to Arm HPC base such as SVE Support of Linux GCC and OpenHPC
    https://developer.arm.com/hpc
- Linaro
  - Building binary portability on Arm HPC
    - Standardization of Arm Basic System Software (Linux Kernel, glibc, GCC etc.) and Upstreaming to OSS community
    - Developing and upstreaming SVE software to OSS community
      https://www.linaro.org/sig/hpc/
- OpenHPC
  - Developing Standard IA and Arm HPC software portability
  - Distribution Schedule
    - 2017/11: v1.3.3 for Arm Normal version distributed
    - 2018/6: v1.3.5 for Arm distributed

    http://www.openhpc.community/





FUJITSU

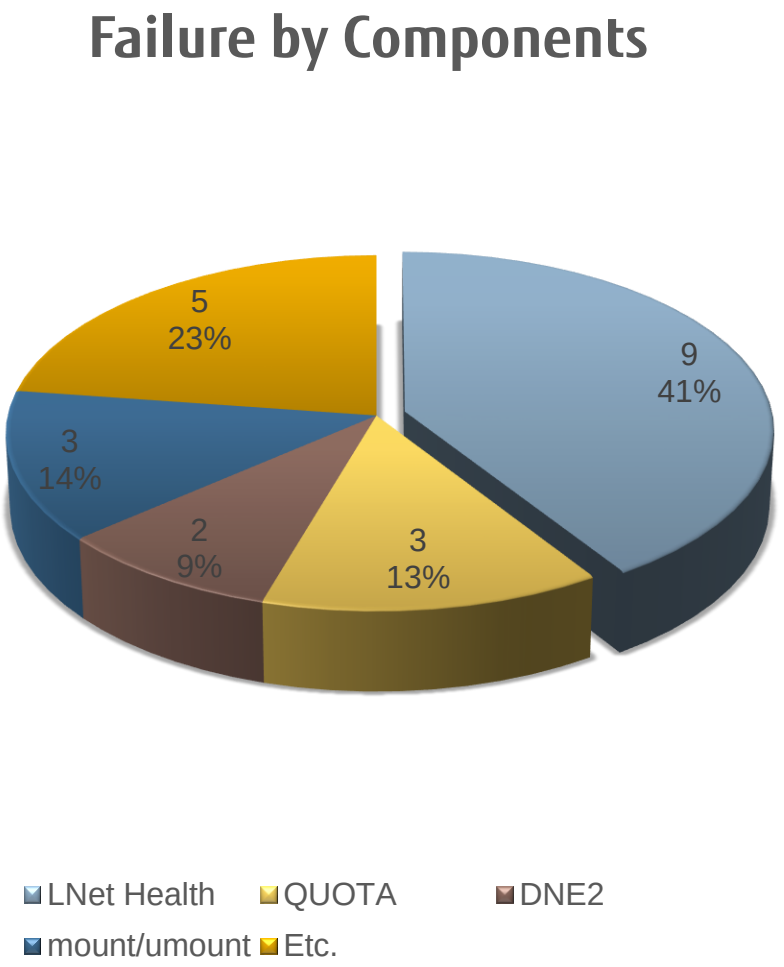
Lustre Testing Reports

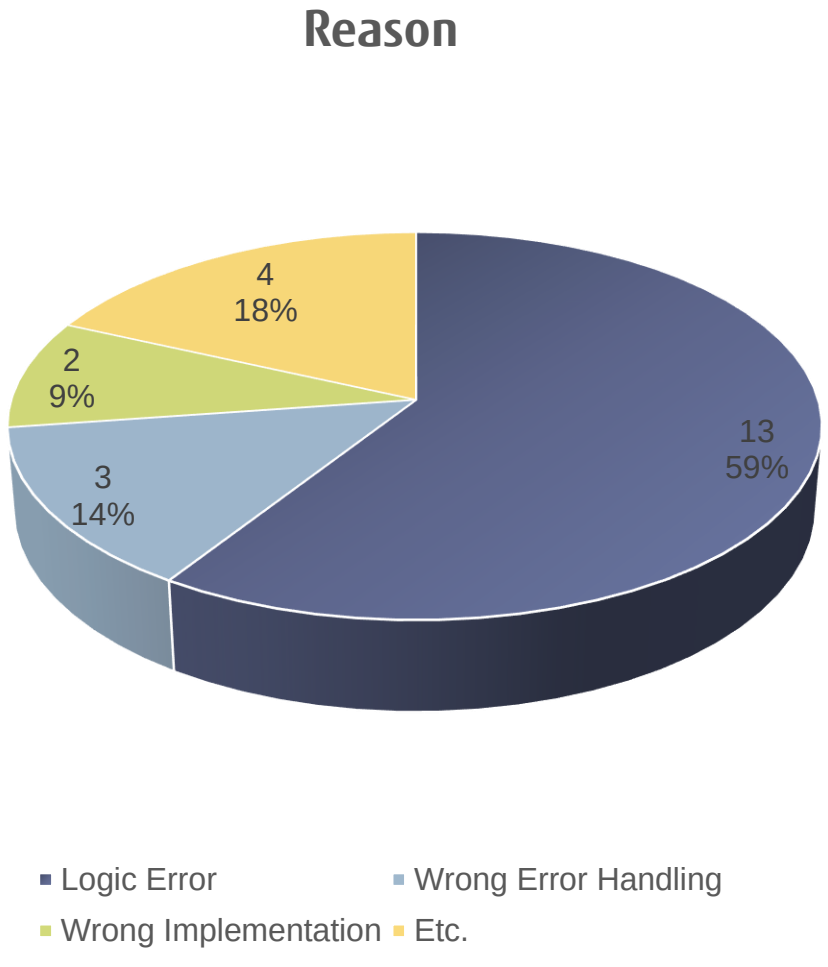
- Lustre 2.10.5

Lustre 2.10.x Testing Results @Fujitsu

FUJITSU

- Failure Analysis on FEFS (Lustre.2.10.5+LU-9120+LU-9480)





Testing Results Error: LNet Network Health

FUJITSU

- Testing Error Summery on LNet Network Health
  1. Unable to detect device faults from IB event queue (LU-12287)
  2. Preferred flag of route selection policy does not work (LU-12288)
  3. Route with fault remote device selected on separated IB subnet (LU-12289)
  4. Inconsistent Timeout value (one is 5sec, the others are 50s) (LU-12290)
  5. Wrong NI selection on asymmetric Multi-rail environment (LU-12291)
  6. Decrementing Health Value even if recovery processing fault (LU-12292)
  7. Wrong counting remote device fault as local device fault (LU-12287)
  8. Memory leak after router checker packet processing (LU-12293)
  9. Memory leak after recovery packet processing (LU-12294)

No.1 Unable to detect device faults from IB event queue

FUJITSU

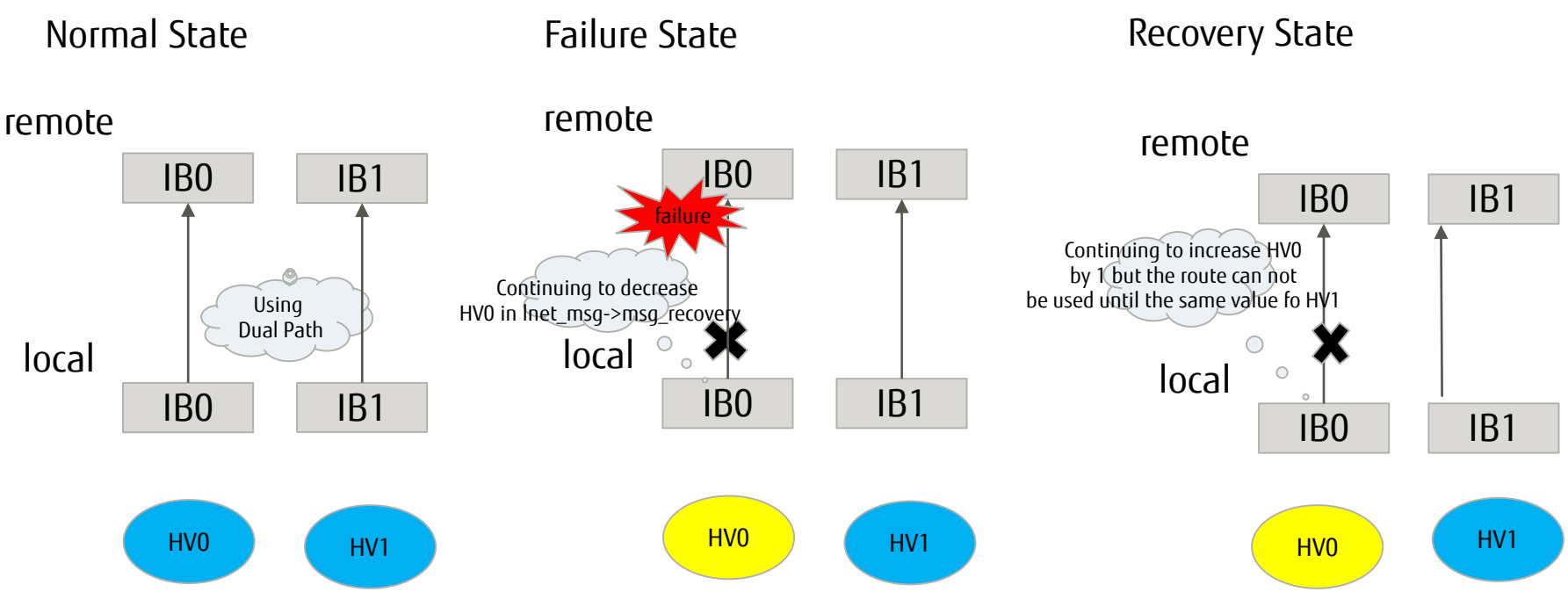
- Issue
  - Unable to detect device hardware errors such as hardware errors and link down errors
- Why came from?
  - Current implementation only processes CQ error, and does not handle async event data
- Proposed Solution
  - Add async event handler routine by using ib_register_event_handler()

No.6: Decrement Health Value even if recovery processing fault

FUJITSU

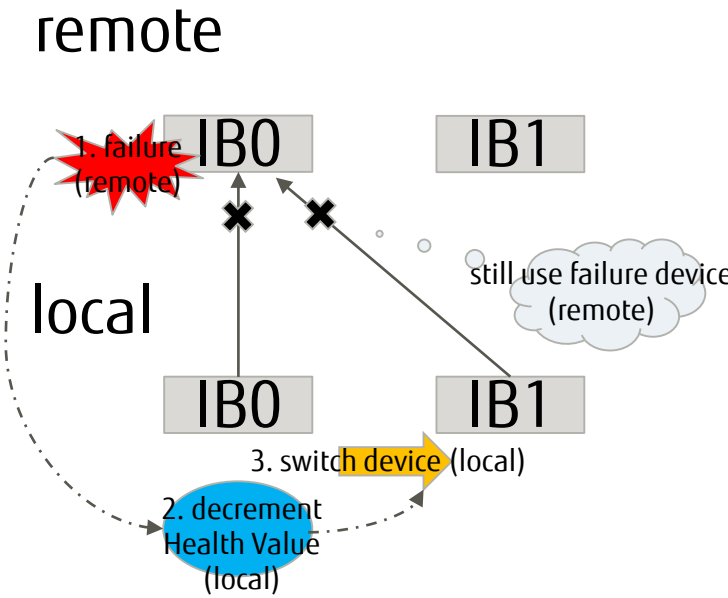
- Issue
  - Unable to use route for a while after device failure recovery
- Why came from?
  - Health value is:
    - Used to know error route
    - Decreased when a routing error is detected
  - Health value continues to be decreased in device failure phase to be zero
  - After a device recovery from the device failure, the health value is increased.
  - In this case, it takes number of loops until the device can be used even if the device can be used
- Proposed Solution
  - Stopping health value decrement after a device failure is detected and handles in lnet_msg->msg_recovery for the following status:
    - LNET_MSG_STATUS_LOCAL_DROPPED (Connection failure)
    - LNET_MSG_STATUS_LOCAL_TIMEOUT (Connection timeout)

Decrement Health Value even if recovery processing fault



No.7: Wrong counting remote device fault as local device fault

- Issue
  - After remote device failure, local node still sends to the failure remote device using different local device
- Why came from?
  - When remote device failure is detected, HV of the local device is decreased.
  - IB0 is not used and local device uses IB1 to send IB0 of remote device.
- Solution
  - Device failure of remote device can be detected by connection failed or connection time out.
  - In the cases, receiving the following events should be decrement as remote device HV
    - LNET_MSG_STATUS_LOCAL_DROPPED (Connection Failure)
    - LNET_MSG_STATUS_LOCAL_TIMEOUT (Connection timeout)



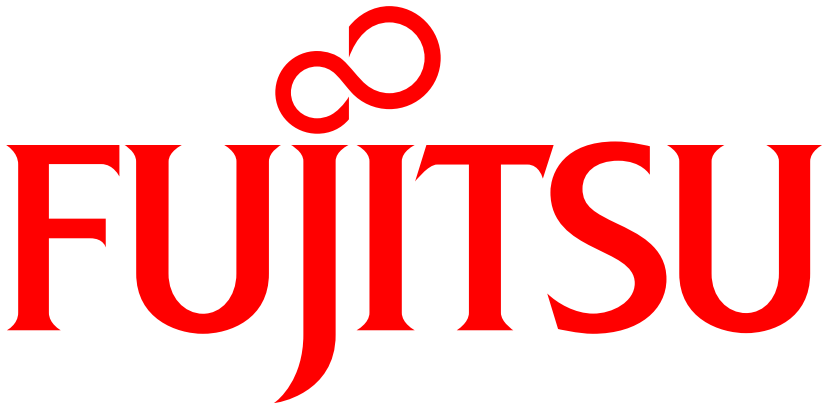
The other under investigating Failures

FUJITSU

- MDS Panic on DNE2 directory removing (LU-12295)
  - In case of full of MDT disk and returning –ENOEPC
  - In case of failure of memory allocation and returning –ENOMEM

  As a result of out_tx_write_exec() function returns error

Etc…



shaping tomorrow with you