



スーパーコンピュータ「富岳」の ファイルシステム

宇野 篤也

System Operation and Development Unit,
Operations and Computer Technologies Div.
RIKEN Center for Computational Science

- 富岳概要
- **Storage system**
 - 1st layer file system
 - 2nd layer file system
 - Monitoring and Log collection
- 運用上の問題点
- まとめ

富岳概要

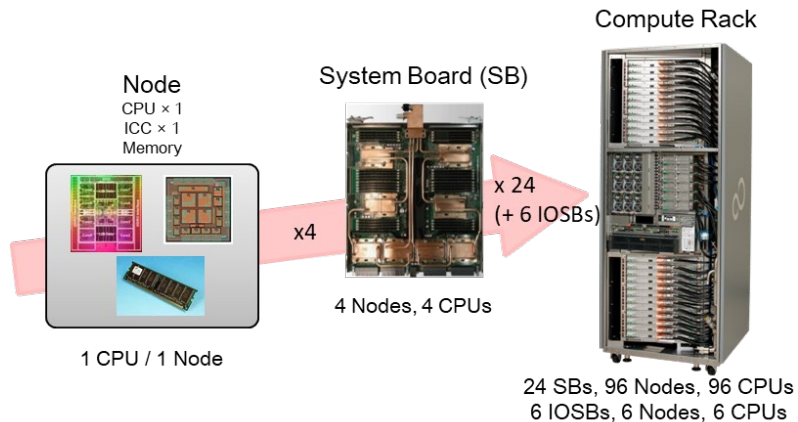
スーパーコンピュータ「富岳」

- 2014年からフラッグシップ2020プロジェクトが開始
- 2019年に詳細設計が完了し、2020年5月に全ての計算ノードの搬入が完了
- 2021年の共用開始に向けて、システム調整を実施
 - 2020年4月から一部の計算ノードを共用前評価環境として提供
 - 成果創出加速プログラム, covid19対策課題等で利用
- 2021年3月より共用運用開始
- 3期連続四冠達成 (ISC20, SC20, ISC21)
 - TOP500, HPCG, HPL-AI, Graph500

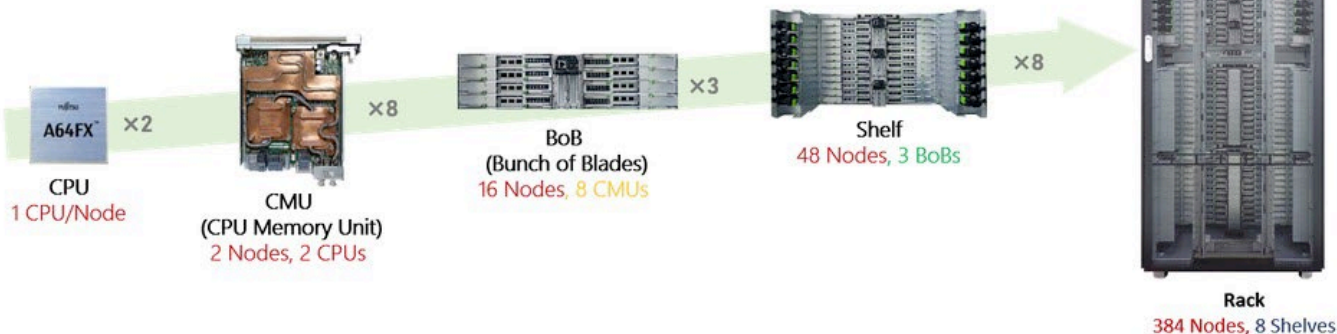


Ranking	Performance	Racks	Nodes
TOP500	442.01 PFLOPS	432	158,976
HPCG	16.0 PFLOPS	432	158,976
HPL-AI	2,004 PFLOPS	432	158,976
Graph500	102,955 GTEPS	432	158,976

ref. <https://www.r-ccs.riken.jp/fugaku/>



	京	富岳
Nodes	82,944+5,184	158,976
Racks	864 (24 compute nodes) (6 IO nodes)	432 (396 : 384 nodes) (36 : 192 nodes)

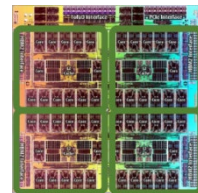
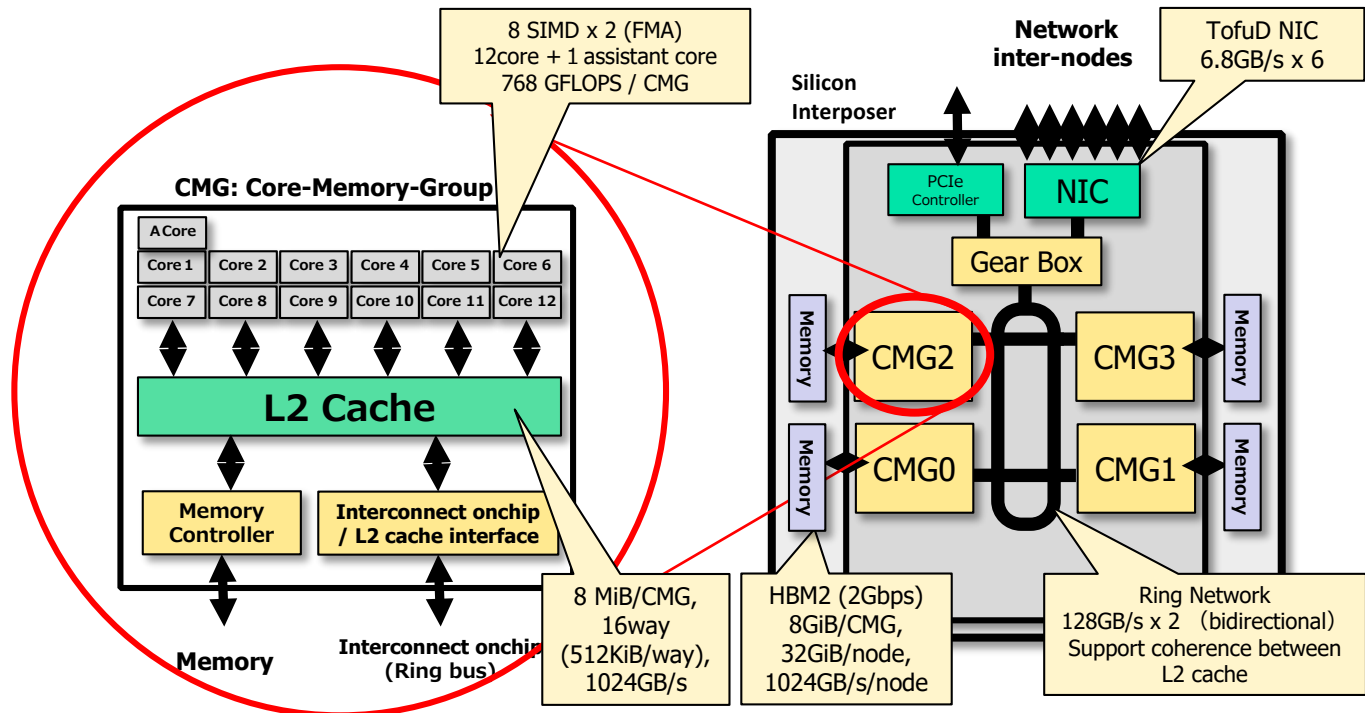


CPU - Fujitsu A64FX

- **A64FX**

- Arm 8.2-A SVE CPU made by Fujitsu (<https://github.com/fujitsu/A64FX>)
 - 48 cores for compute and 2/4 for OS activities
- TofuD interconnect interface is integrated into CPU chip

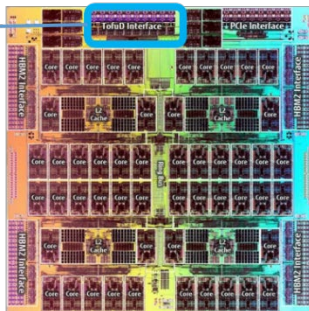
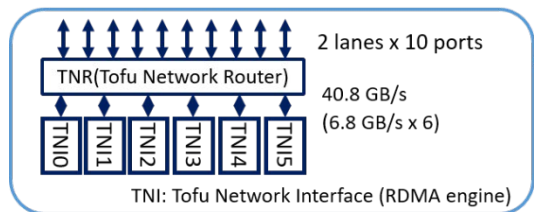
@normal mode (2.0GHz)



● Tofu Interconnect D

- Features
 - 6 RDMA Engines
 - Hardware barrier support
 - Network operation offloading capability

- Performance
 - 8B Put latency : 0.49-0.54 usec
 - 1MiB Put throughput : 6.35 GB/s

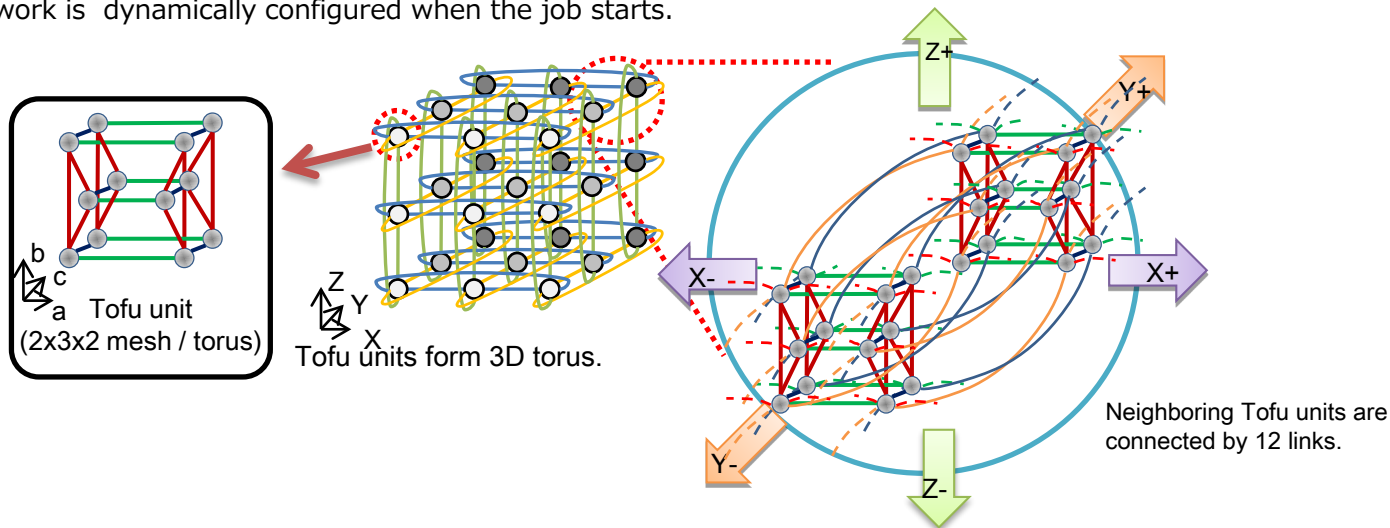


Comparison with Tofu

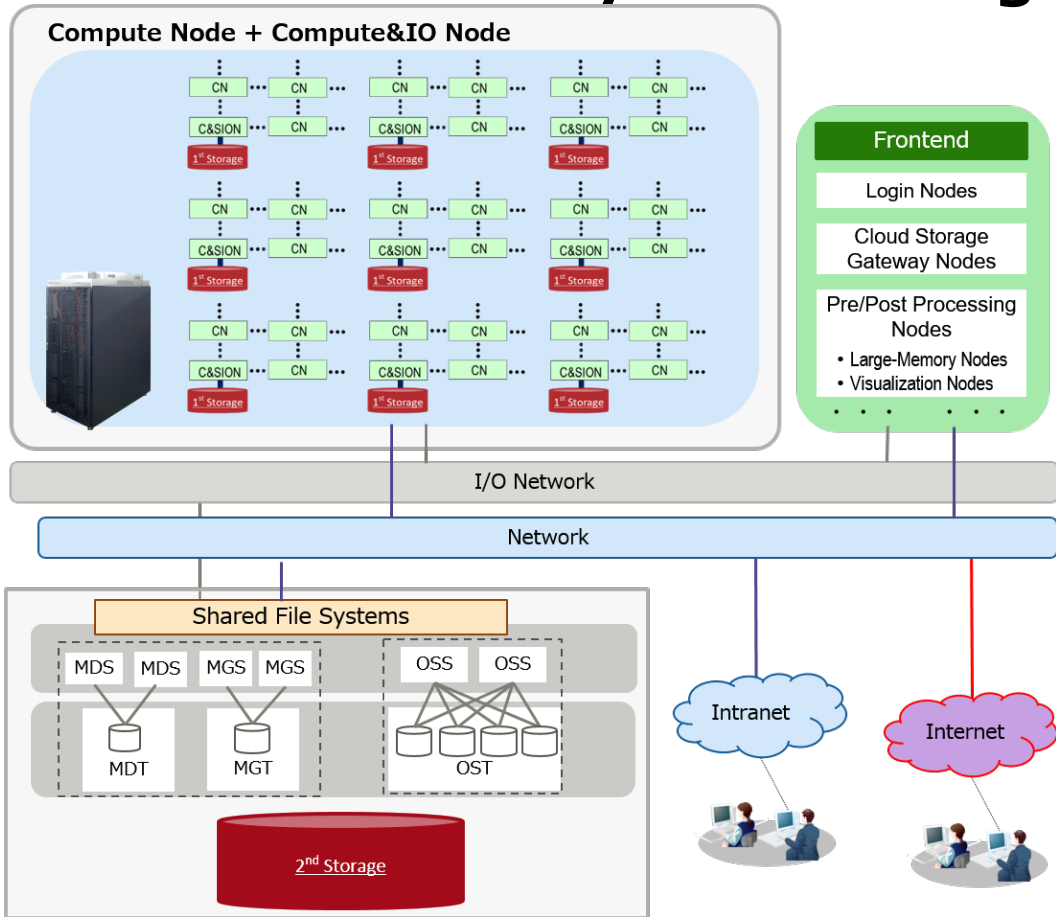
	Tofu	TofuD
Data rate (Gbps)	6.25	28.05
Link bandwidth (GB/s)	5.0	6.8
# of TNIs / node	4	6
Injection bandwidth / node (GB/s)	20	40.8

Tofu (Torus fusion) Interconnect D

- High communication performance and fault-tolerant network
- Network topology : 6D mesh / torus network
 - Each node has 10 links (28 GB/s x 2 bandwidth)
 - 4 links for making a Tofu unit (2x3x2 mesh / torus, a,c:mesh / b:torus) and 6 links for XYZ link (XZ:torus / Y:mesh)
 - Network size : (X, Y, Z, a, b, c) = (24, 23, 24, 2, 3, 2)
 - Multi-path routing by a combination of XYZ link and 2x3x2 mesh / torus network enables to make a detour of faulty nodes
- From user's point of view, network topology of the job is 1,2 or 3D torus network.
 - This torus network is dynamically configured when the job starts.



System Configuration



- **Compute node**

- 432 racks (158,976 nodes)
 - Each rack consists of 384 nodes, but 36 racks includes half nodes)

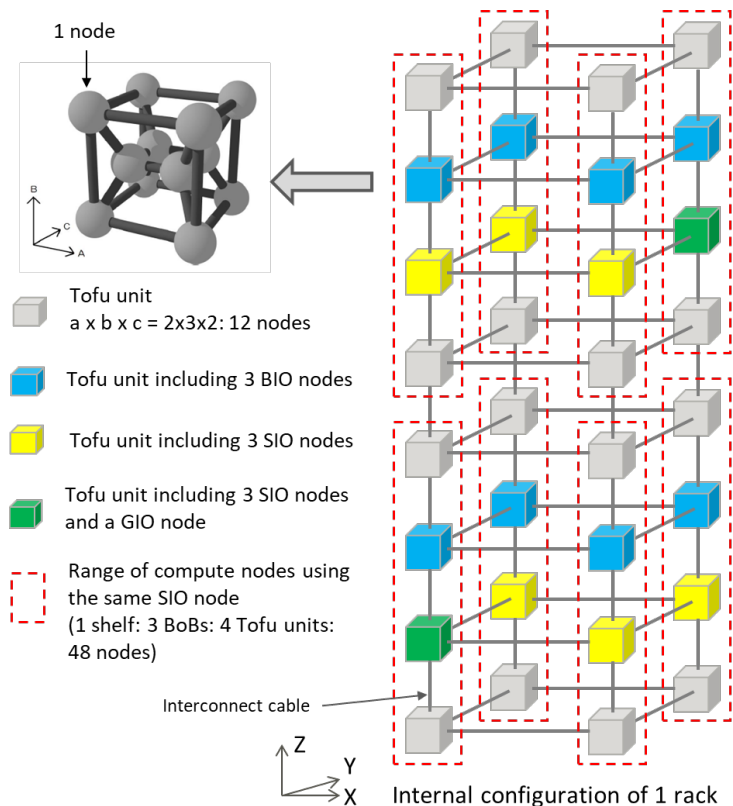
- **Interconnect**

- TofuD (28 Gbps x 2 lanes x 10 ports)
 - 6D mesh/torus (Physical configuration)
 - 3D torus (Logical configuration)

- **Storage system**

- 1st Layer
 - Cache for 2nd Layer file system
 - Temporary file systems
 - Local file system for compute node
 - Shared file system for a job
- 2nd Layer
 - Fujitsu FEFS: Lustre-based file system, about 150 PB
- 3rd Layer
 - Cloud storage services are considered

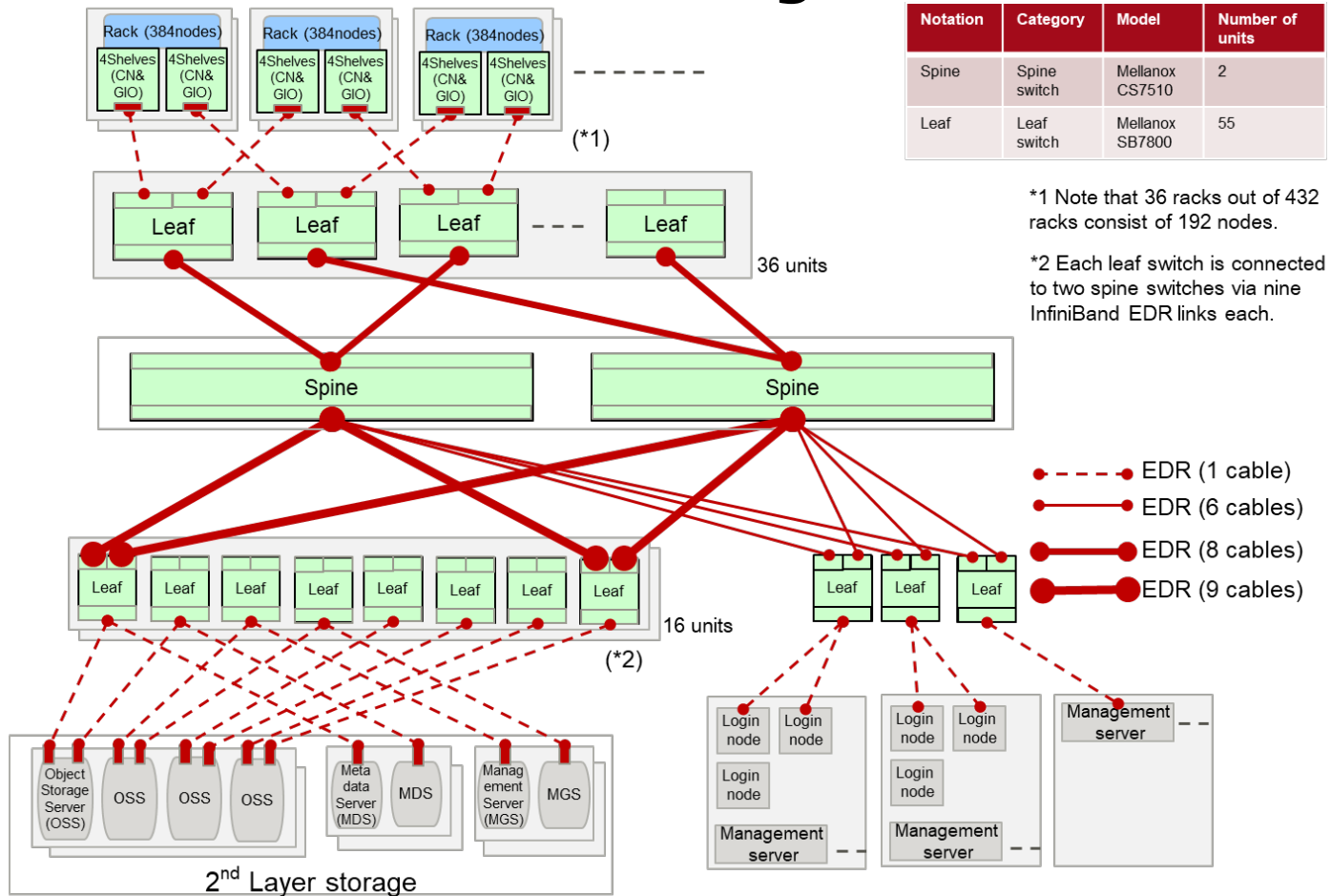
Types of nodes



- **Compute node**
 - Two assistant cores
 - they improve yield and reduce power
- **Compute node with I/O (I/O node)**
 - Four assistant cores
 - they serve I/O operations as well as OS operations
- Three types of I/O nodes
 - BIO (Boot I/O node)
 - a boot server including boot storage, one BIO in a BoB (16 nodes)
 - SIO (Storage I/O node)
 - a storage server including SSD of 1st layer storage, one SIO in a BoB
 - GIO (Global I/O node)
 - a relay node over InfiniBand for FEFS, which is a cluster file system of 2nd layer storage, two GIOs in a rack (384 nodes)

fig. Internal configuration of compute rack.

IB Network configuration



Comparison with K computer

		Fugaku	K computer
Architecture		A64FX (Armv8.2-A SVE + Fujitsu Extension)	Sparc64 VIIIfx
Node	Cores	48 cores for compute and 2/4 for OS activities	8 cores
	Peak DP performance	3.072 TF (3.3792 TF)	0.128 TF
	Main memory	HBM2 32 GiB	DDR3 16 GiB
	Peak Memory Bandwidth	1024 GB/s	64 GB/s
	Peak Network performance	TofuD (28 Gbps x 2 lane x 10 port), 40.8GB/s	Tofu (5 Gbps x 2 lane x 10 port), 20GB/s
Technology		7 nm FinFET	45 nm CMOS
Rack	Nodes	384 or 192	102
	Peak DP performance	1.2 / 1.3 PF	< 0.013 PF
System	Racks	432 (36 racks is a92 nodes)	864
	Peak DP performance	488 PFLOPS / 537 PFLOPS	11.28 PFLOPS
	Power (HPL)	29.9MW	12.7MW
	Rpeak	442 PFLOPS	10.51 PFLOPS

HPL Performance is about 42x , but power consumption is only 2.4x

Storage system

• Three-layer storage system

• 1st Layer

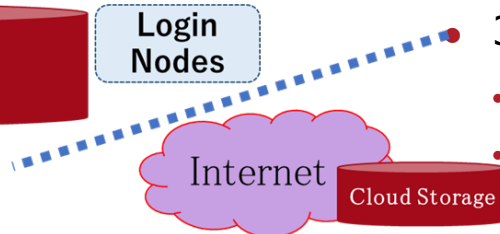
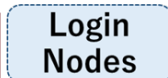
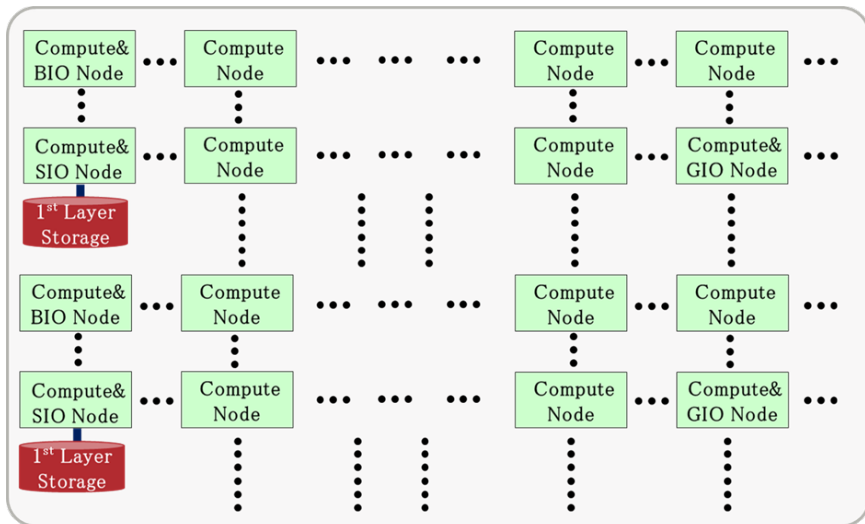
- One of 16 compute nodes, SIO node, has SSD about 1.6 TB
- Services on SIO node
 - Cache for 2nd layer
 - Temporary file systems
 - Local file system for compute node
 - Shared file system for a job

• 2nd Layer

- FEFS (Fujitsu Exabyte File System):
Lustre-based file system, about 150 PB

• 3rd Layer

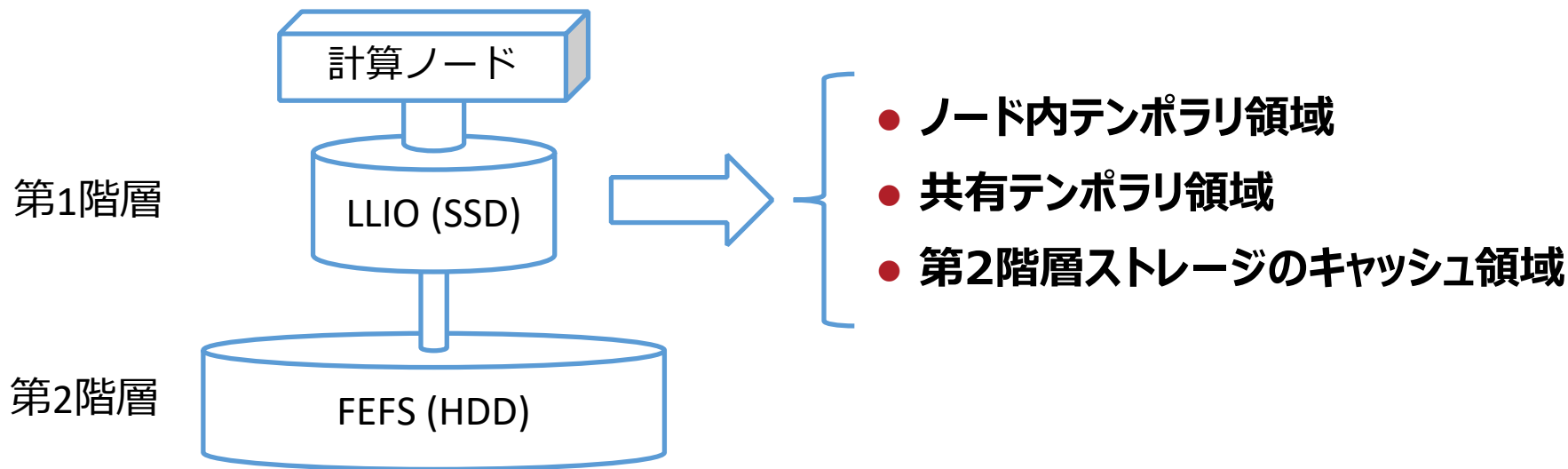
- Cloud storage services are considered
- HPCI storage

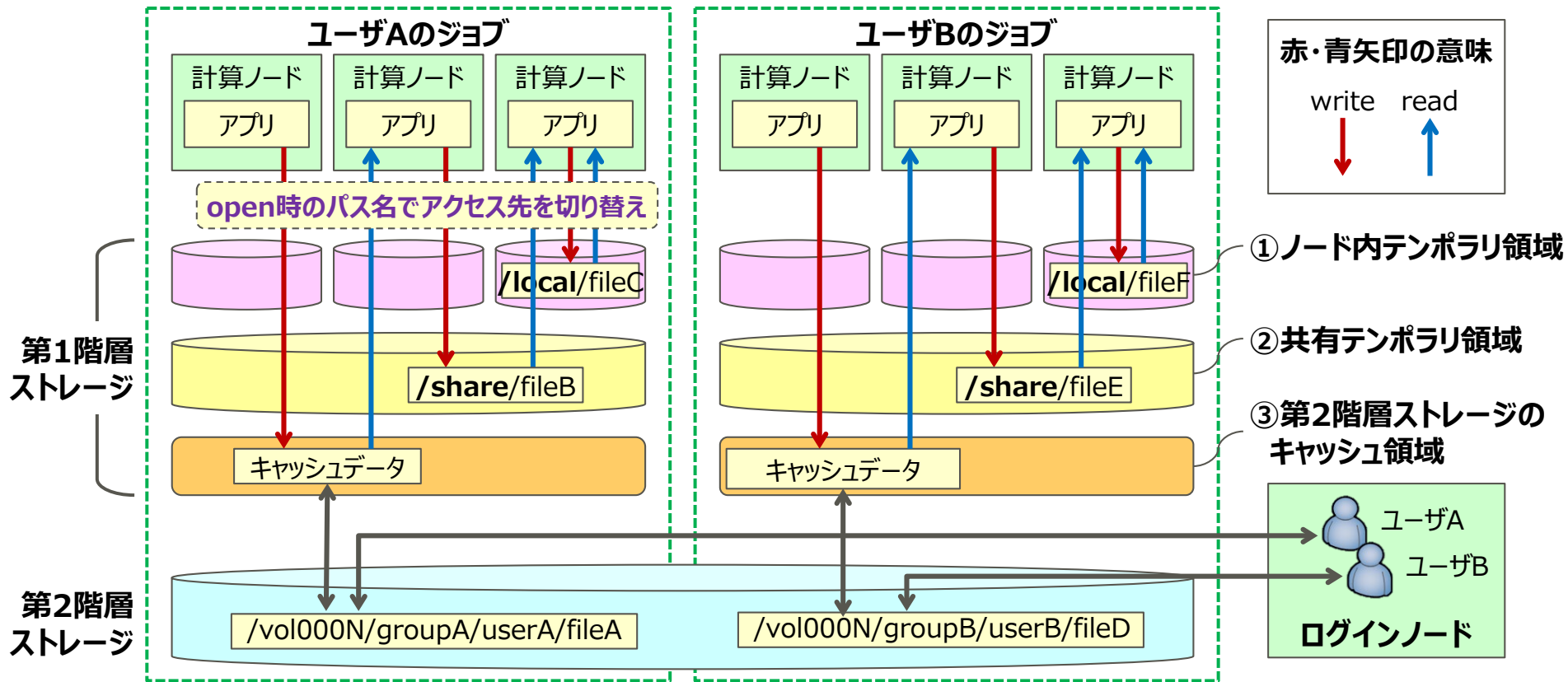


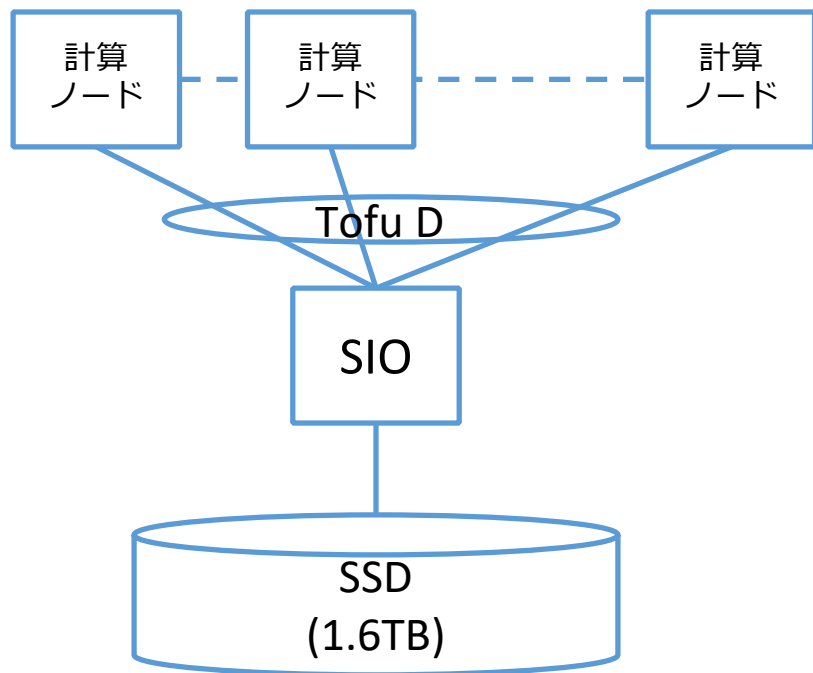
1st layer file system

LLIO: Lightweight Layered I/O Accelerator

- LLIOとは、並列分散ファイルシステムであるFEFSと計算ノードの間に位置する、SSDを使用した高性能なファイルシステム、またはそれを実現する技術
- ジョブ用の一時ファイルをFEFSに書き出さない、またはLLIOからFEFSへの書出しを計算処理中に非同期に行うことで高速化を実現

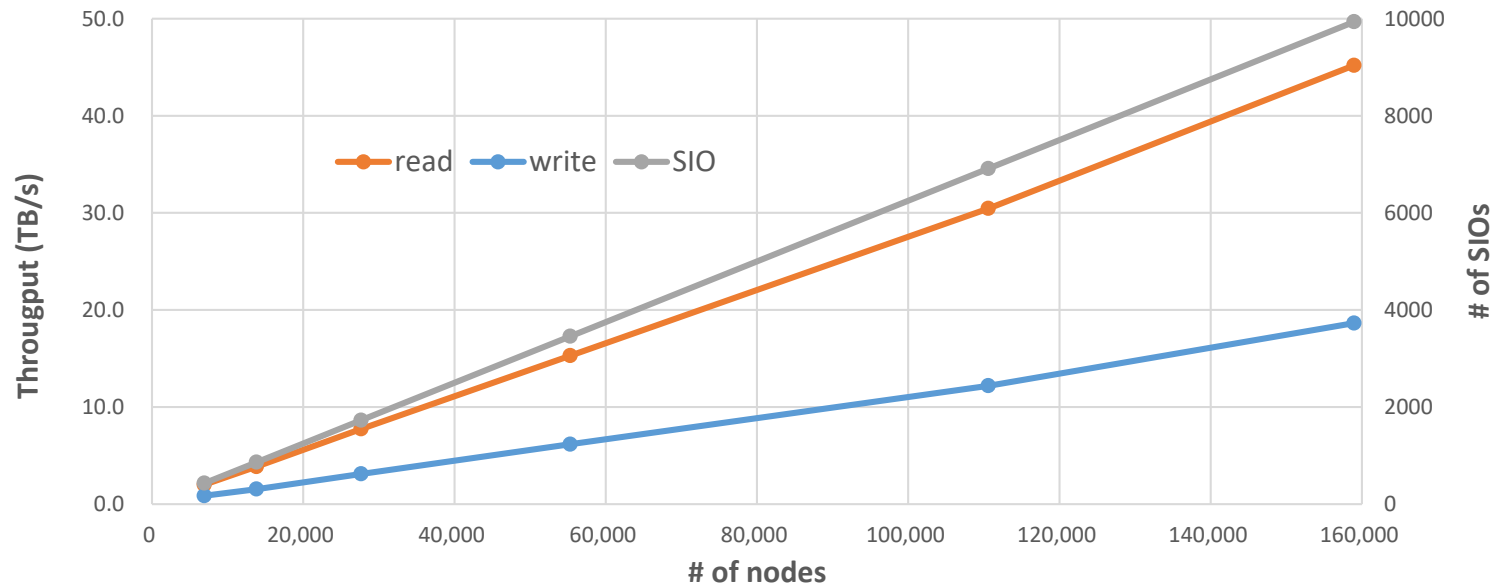




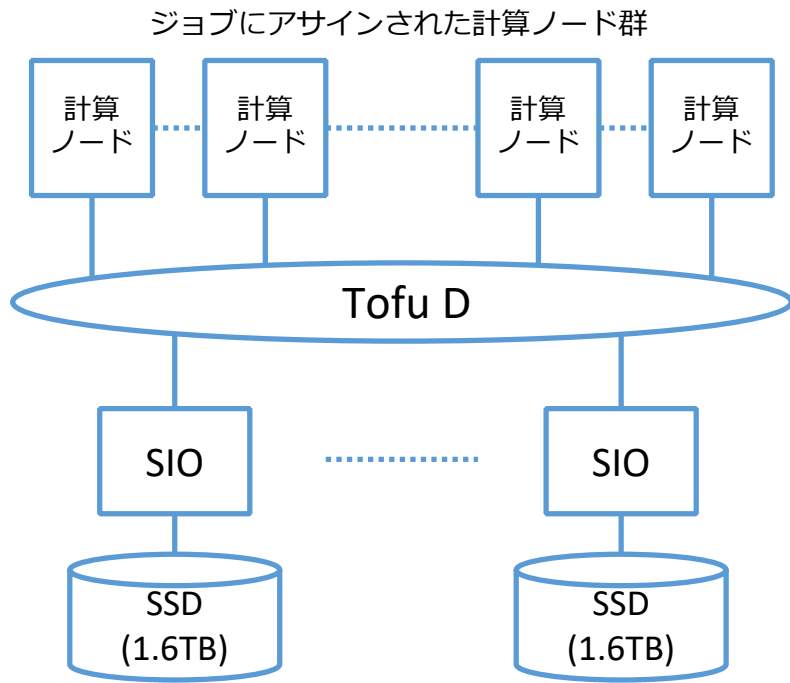


- ジョブに割り当てられたそれぞれの計算ノードで利用できる一時領域
 - ジョブ開始時に作成, 終了時に削除
- SIOに接続されたSSDを16台の計算ノードで共有
 - 最大 87GiB / ノード
- SIOがメタデータサーバ

IORの結果

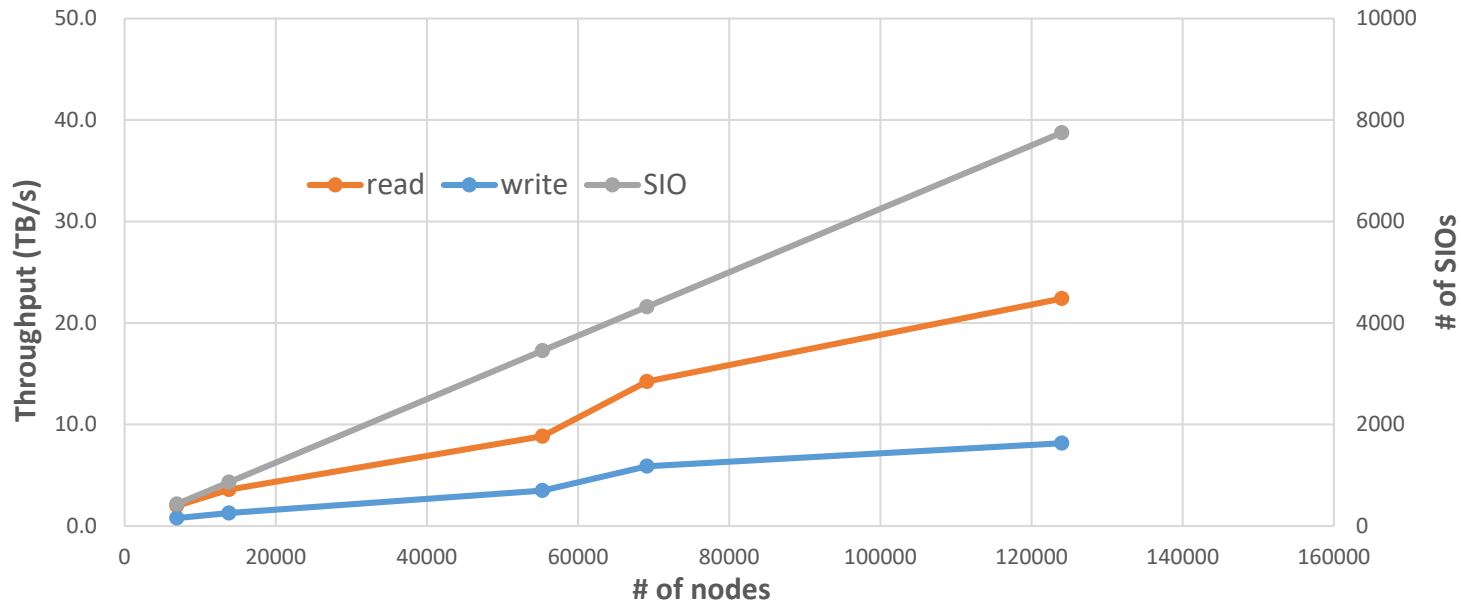


- 1proc/node, 1GiB, 1 stripe count
- 性能はSIO数に比例

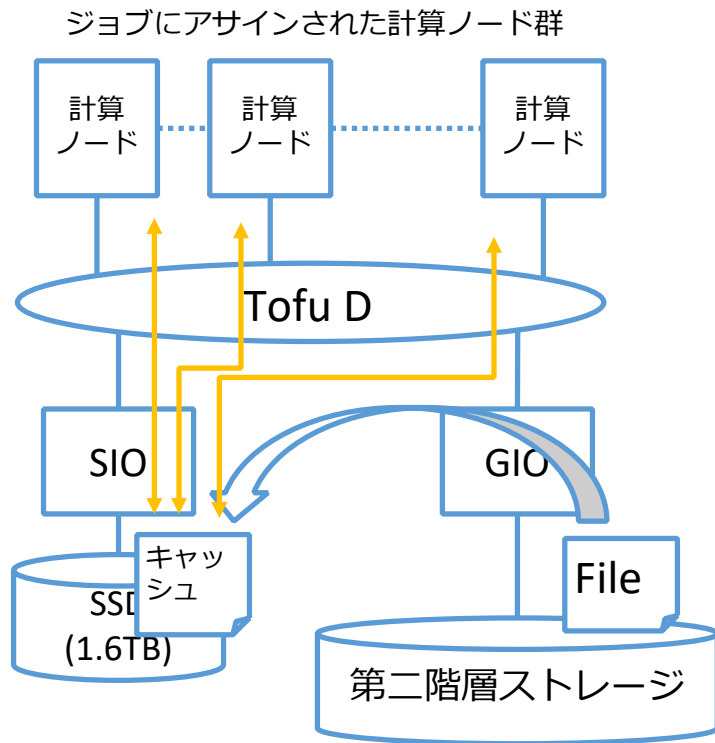


- **ジョブにアサインされた計算ノード間で共有できるテンポラリ領域**
 - 最大容量 87GiB x ノード数
 - SSD上にのみ存在
 - 第2階層へは書き出されない
- **ジョブで利用できるSIOの数はジョブが使用するノード数でに比例**
 - 規模の大きいジョブは, SIOを専有するようにノードを割り当て
- **LLIO用のMDSを用意**

IORの結果

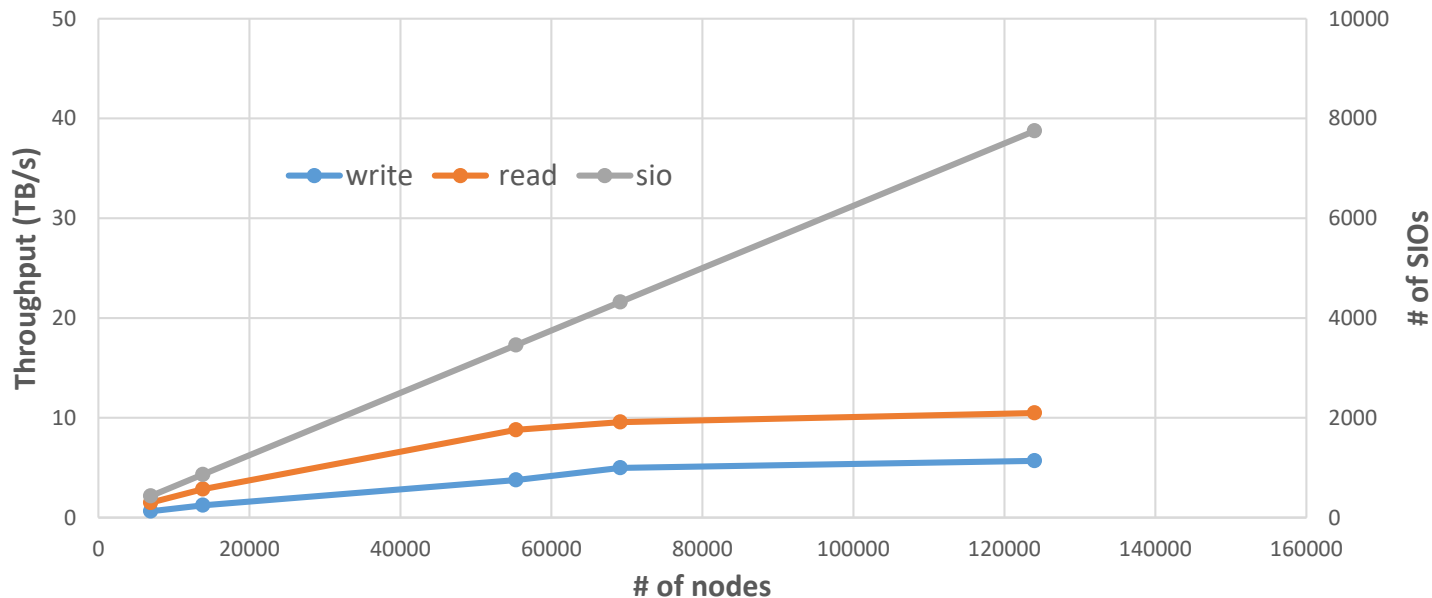


- 4proc/node, 1GB(-13,824 nodes), 4GB(55,296- nodes), 1 stripe count
- SIO数に比例して性能は向上するものあまり伸びない



- **第二階層ストレージのキャッシュ機能**
 - 同期/非同期クローズ
- **ジョブにアサインされた全ての計算ノードから参照可能**
 - ジョブ開始時に作成, 終了時に削除
- **共有読み込みファイルの一斉転送機能**
 - 通常アクセスでは, 一つのSIO上にキャッシュが作られるため, 多数ノードからのアクセス時にアクセス性能が低下
 - llio_transferコマンド
 - 全てのSIOにレプリカを作成し, 高速なファイルアクセスを実現

IORの結果



- 4proc/node, 4GiB, 1 stripe count
- MDSの性能で頭打ち

- **概要**

- 共通ファイルの配布コマンド (llio_transfer) を使用したメタアクセス・データアクセスの性能を測定
- FEFSアクセスと, llio_transferを使用した場合の性能を比較

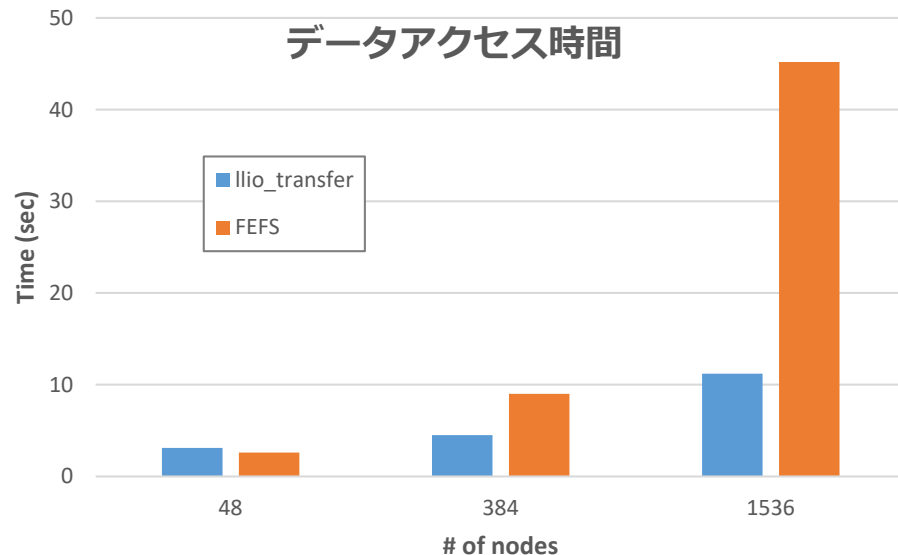
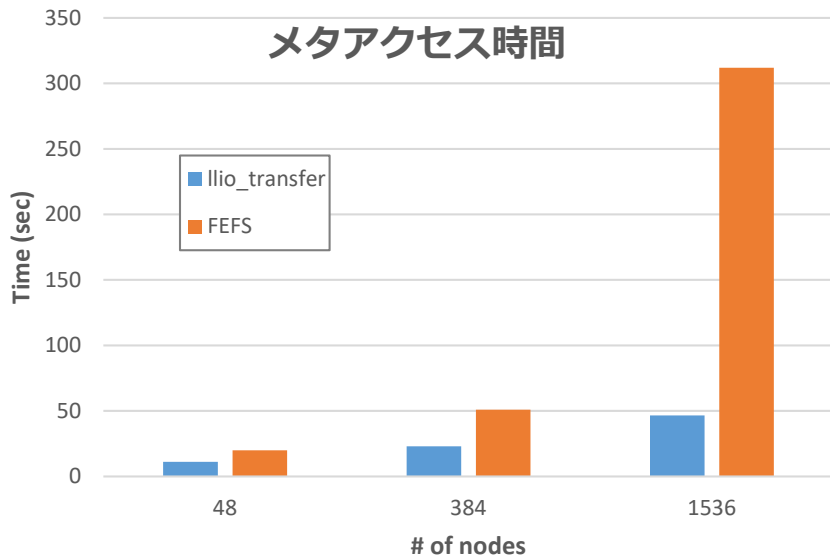
- **測定方法**

- mpiexec の実行時間を測定
 - メタアクセス: import のみのpythonスクリプトの起動 (約 600 files)
 - データアクセス: dd で 1プロセスあたり256MiBのデータを読み込み
 - llio_transfer 処理時間を含む

- **測定条件**

- nodes: {48, 384, 1536:8x6x32} (48, 384はノード形状指定無し)
- 4プロセス/ ノード
- rank-map-bychip
- llio-async-close off
- stripe-count 24
- 通常運用時の他ジョブが流れていない状態で測定

llio_transfer 測定結果

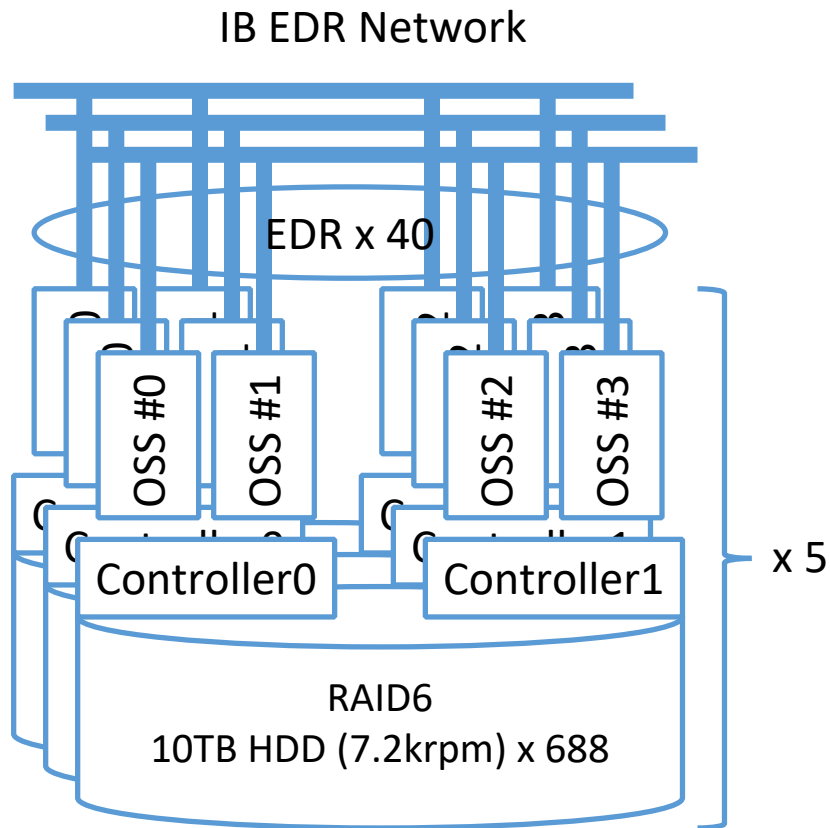


- ノード規模に比例して性能が向上

- メタアクセス・データアクセスともに、SIO台数による性能のボトルネックが解消され、高速化

2nd layer file system

第2階層ストレージ構成



- **FEFS (Fujitsu Exa-scale File System)**

- 富士通が開発した Lustreベース(Ver.2.1)のファイルシステム
- 「京」の8年の運用経験を反映

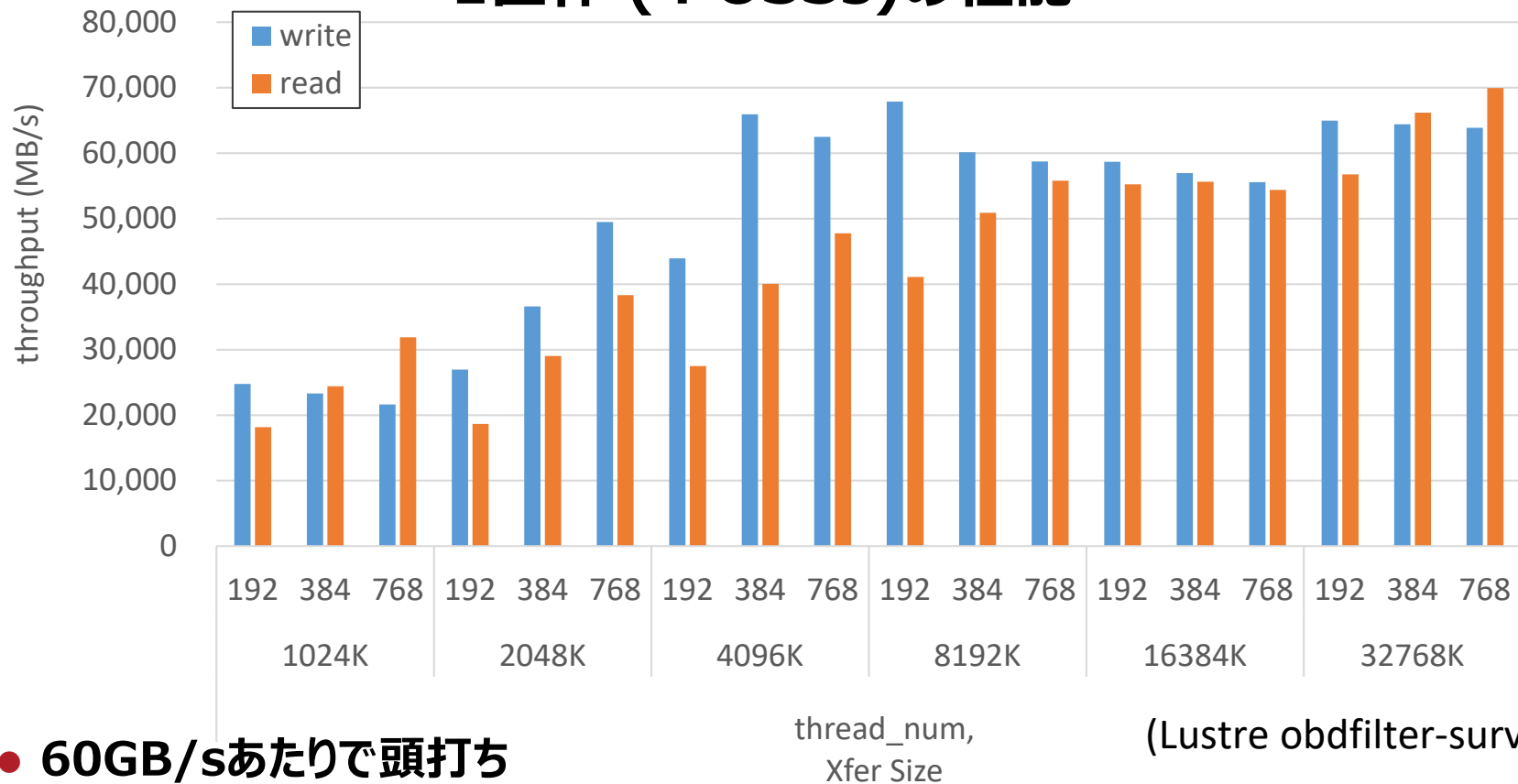
- **合計容量 150PB**

- 6ボリュームに分割
 - 1volume の構成 (DDN装置5台で構成)
 - 容量 : 25PB
 - 20 OSSs, 60 OSTs

- **QoSを設定**

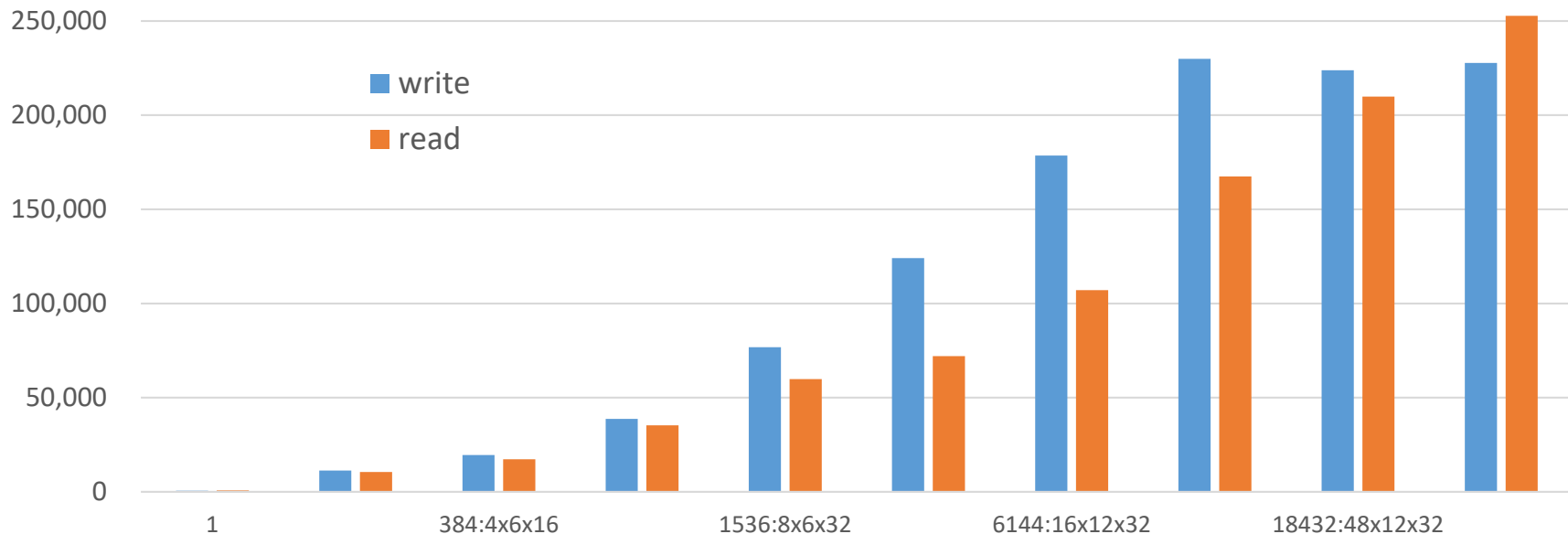
- 計算ノードとログインノード間
- ユーザ間

1筐体 (4 OSSs) の性能



IORの結果

スループット(MB/s)



- **ior -b 1g -t 16m, 1proc/node**
- **GIOの数に比例して性能が向上するが, readの性能が悪い**
 - ランダムアクセスが原因
- **第2階層ファイルシステムのOSSの性能で頭打ちに**

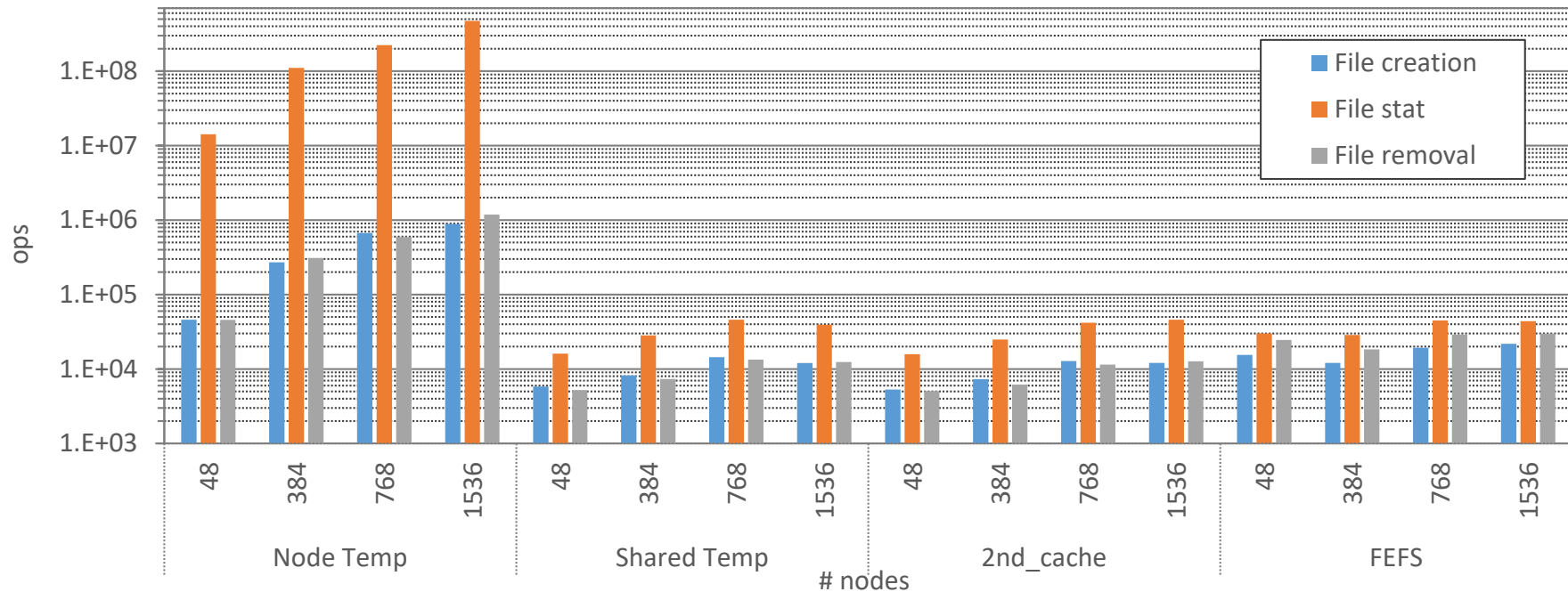
- **ベンチマーク**

- mdtest-3.4.0+dev
- オプション: -i 3 -n 25 -u -F -d OUT_DIR
 - 最大値の計測

- **測定条件**

- nodes: {48:2x3x8, 384:4x6x16, 768:4x6x32, 1536:8x6x32}:strict-io
- 4プロセス/ ノード
- rank-map-bychip
- llio-async-close on
- sio-read-cache off
- stripe-count 1
- 通常運用時の他ジョブが流れていない状態で測定

メタアクセス性能 測定結果



- ノード内テンポラリ領域はノード規模（SIO数）に比例して性能が向上
- 共有テンポラリ領域，第2階層ストレージのキャッシュ領域，FEFSはメタアクセスを処理する共有のMDSがボトルネックとなりノード規模の増加に対し，性能向上は鈍化

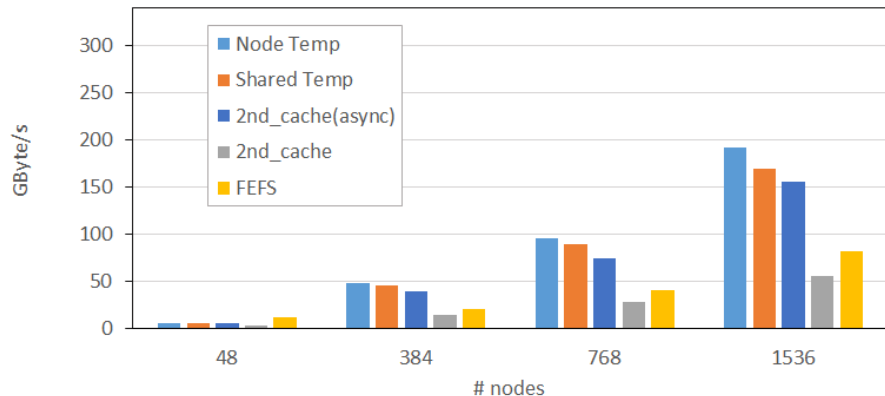
- **ベンチマーク**

- IOR-3.4.0+dev
- オプション: -a POSIX -b 1g -t 16m -i 3 -e -F -v -o dir/out.dat
 - 最大値の計測

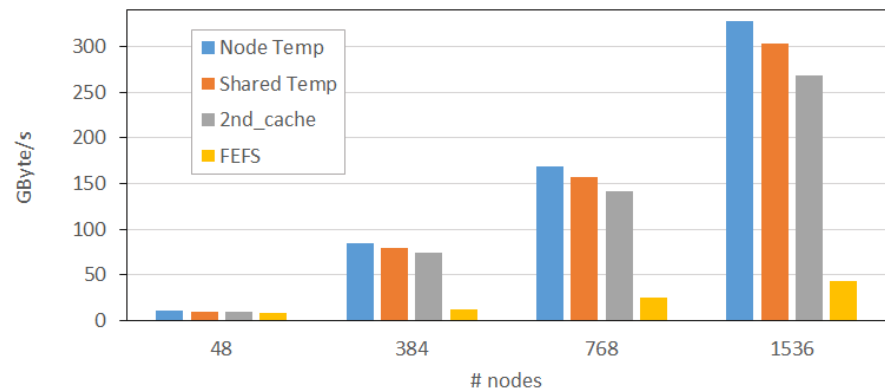
- **測定条件**

- nodes: {48:2x3x8, 384:4x6x16, 768:4x6x32, 1536:8x6x32}:strict-io
- 4プロセス/ ノード
- rank-map-bychip
- llio-async-close on
- sio-read-cache off
- stripe-count 1
- 通常運用時の他ジョブが流れていない状態で測定

Write



Read



- ノード内テンポラリ領域はノード規模（SIO数）に比例してスループット性能が向上
- 共有テンポラリ領域，第2階層のストレージのキャッシュ領域，FEFSもノード規模に比例してスループット性能が向上
- FEFSはストレージサーバがボトルネックとなり，ノード数が増えても 250GB/s 付近で性能限界
- 第2階層ストレージのキャッシュ領域は，close時にfsyncしない場合，SIOへのデータアクセスが完了した時点で復帰するためwrite処理が高速

- 計算ノードからの大量のリクエストの影響で、ログインノードからの処理に遅延が発生
- 第2階層ストレージにQoSを設定
 - 計算ノードとログインノード間
 - ユーザ毎の上限

● 設定内容

	スレッド数の配分	ユーザ毎のスレッド数上限割合	
	ログインノード : 計算ノード	ログインノード	計算ノード
MDS	5:95	30%	80%
OSS	20:80	--	80%

- ログとメトリックの収集

- ログ
 - Logstash/Filebeat
- メトリック
 - Prometheus

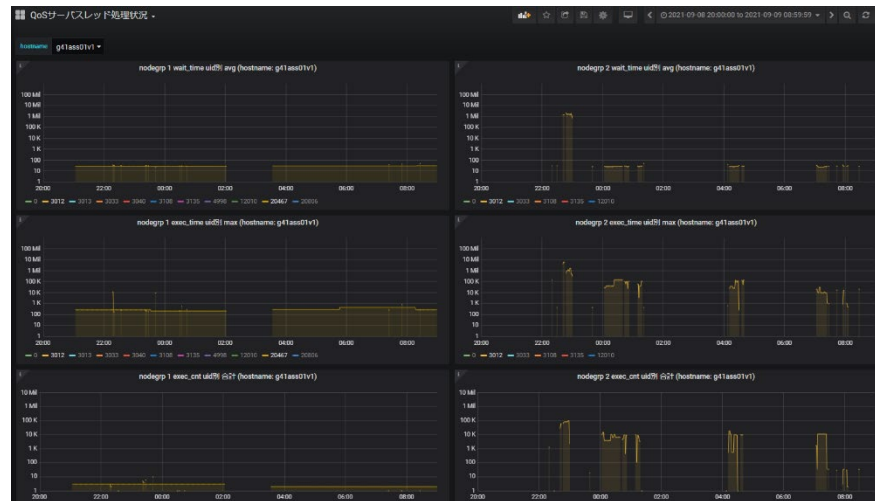
- モニタリングと分析

- DB
 - Elasticsearch, PostgreSQL
- モニタリング
 - Prometheus
- 可視化
 - kibana, redash, Grafana

FEFSサーバのrequest que, threadの監視



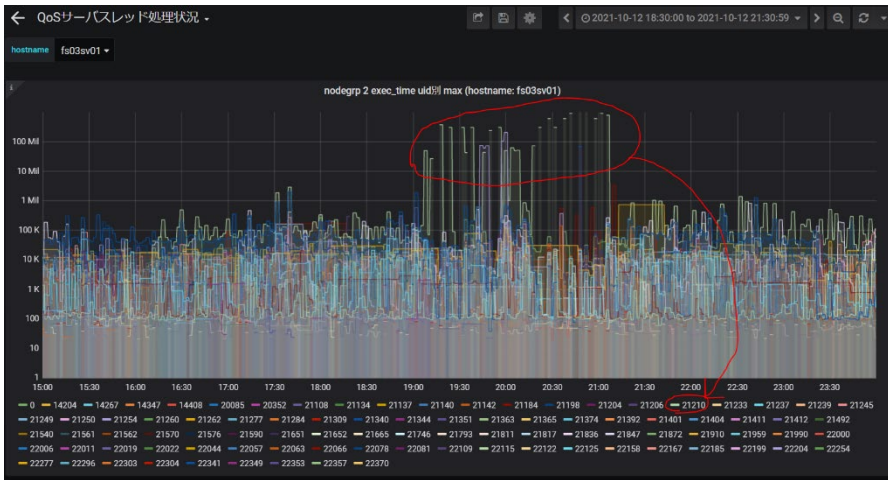
MDS 1台分の表示



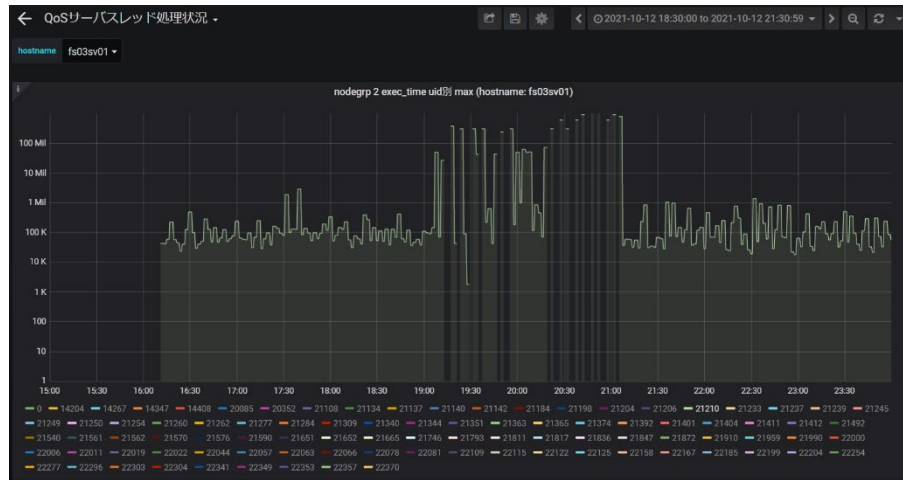
OSS 1台分の表示

- uid毎のリクエストキュー内での待ち時間, スレッド割り当て後の実行時間, リクエスト処理数を表示

● IO全体のスローダウンとなるldlmのロック回収待ちを起こしたジョブ



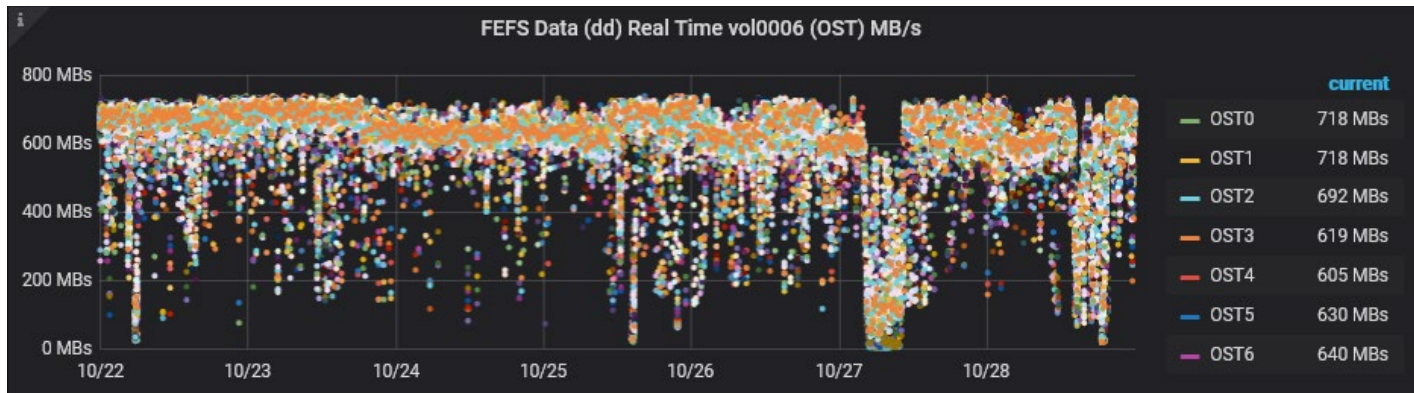
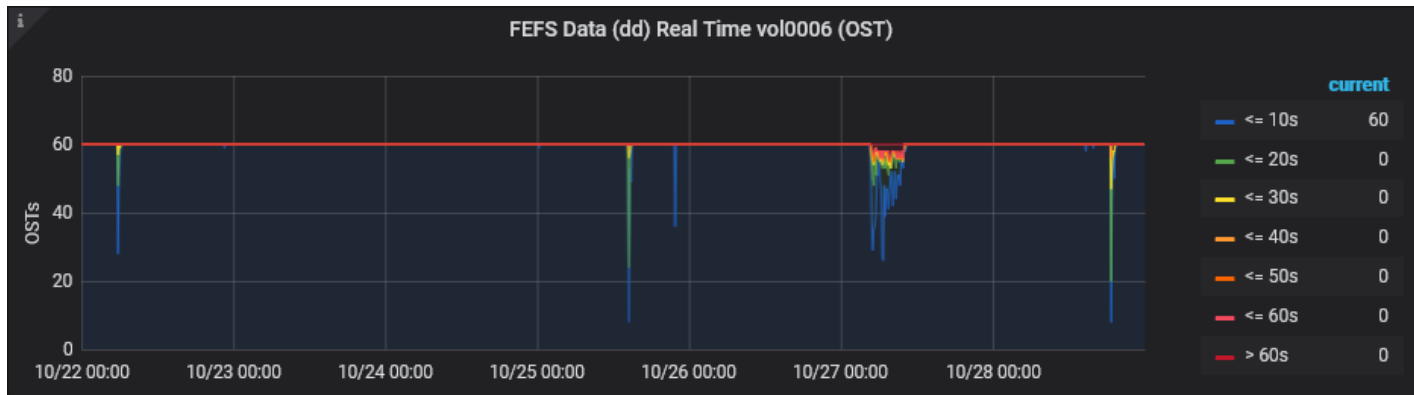
全uidのスレッドの待ち時間



問題のuidのスレッドの待ち時間

FEFSの書き込み・読み込み時間の監視

- 定期的に各OSTの書き込み・読み込み時間を監視



- LLIOは分散ファイルのアクセス性能はよいが、1ファイルにアクセスが集中する場合に問題あり
 - 高並列のMPI-IO等は同時アクセス制限から対応が難しい
- LLIOのMDSはジョブ単位で決まるため、MDSの性能がボトルネックになる場合がある
 - ノード内テンポラリ領域と併用するなどジョブ側でファイルアクセスの工夫が必要
- ファイルアクセスが複雑なため、何が問題になっているかユーザが分析するのが困難
 - LLIOの機能で取得できるファイルIOのプロファイル情報は高並列時は多すぎるため、効率よく取得・分析できる手段が必要

- **「富岳」は2020年5月に全ての計算ノードの設置が完了し、2021年3月から共用運用を開始**
 - 3期連続四冠達成 (ISC20, SC20, ISC21)
 - TOP500, HPCG, HPL-AI, Graph500
- **「富岳」のファイルシステムは3階層構成**
 - 第1階層にLLIOと呼ばれる計算ノードと第2階層の間に位置する高速ファイルシステムを実装
 - ノード内テンポラリ領域, 共有テンポラリ領域, 第2階層ストレージのキャッシュ領域
 - LLIOを活用することで、ファイルアクセス性能を向上させることができるが、利用上の制限も
 - 第2階層はFEFS (6 volume, 計250PB)
- **ジョブのIO特性に合わせた利用方法を模索中**
 - ジョブのIOプロファイルの分析が重要

Thank you for your attention !

