

大阪大学サイバーメディアセンター

OCTOPUS 利用説明会

2018年 6月14日
日本電気株式会社

Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ
類のないインテグレーターとしてリーダーシップを発揮し、
卓越した技術とさまざまな知見やアイデアを融合することで、
世界の国々や地域の人々と協奏しながら、
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。

アジェンダ

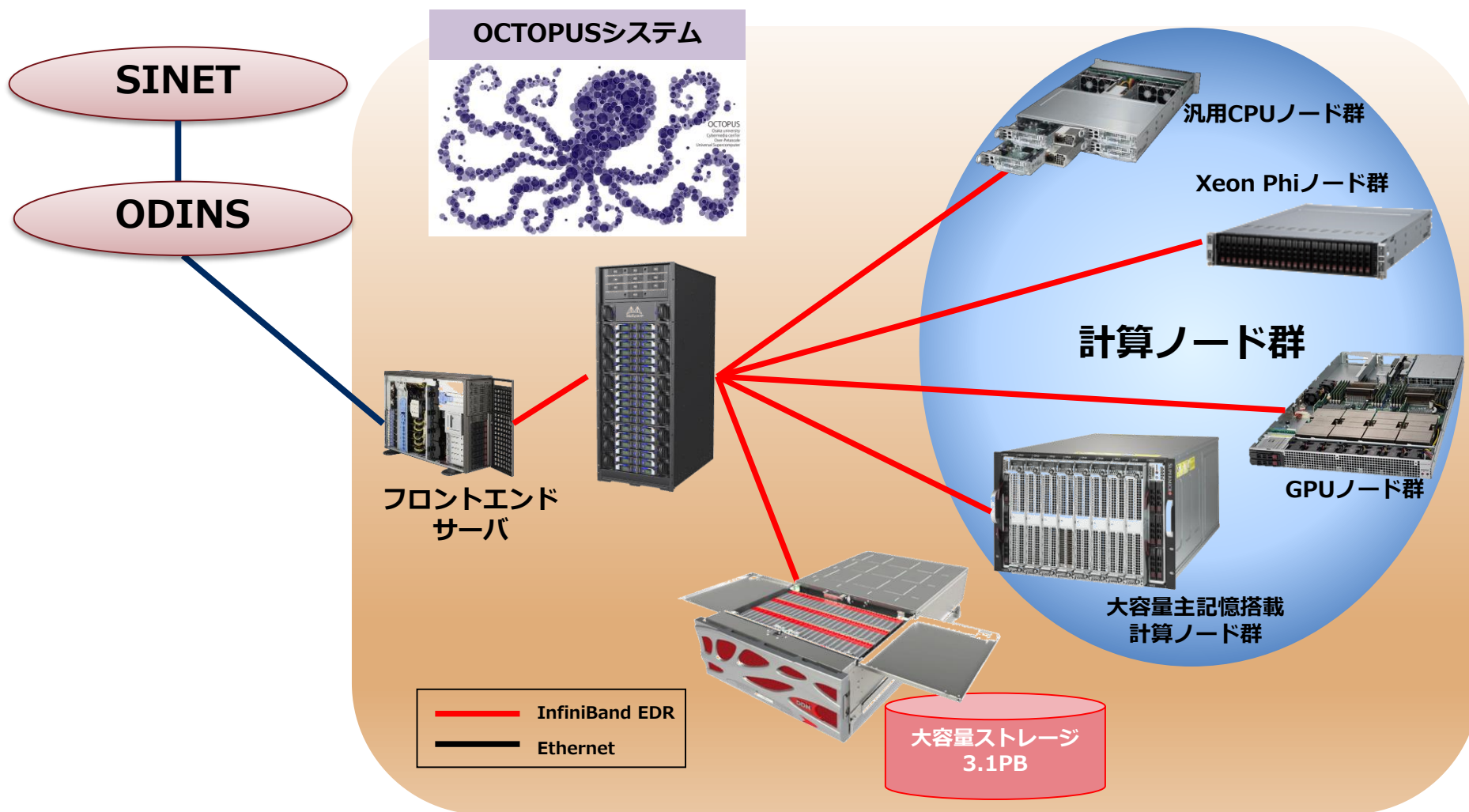
1. システム構成
2. 利用概要
3. 計算ノード群
4. 大容量ストレージ
5. ソフトウェア環境
6. OCTOPUSの利用方法
7. 利用状況表示
8. 利用の手引き

アジェンダ

1. システム構成
2. 利用概要
3. 計算ノード群
4. 大容量ストレージ
5. ソフトウェア環境
6. OCTOPUSの利用方法
7. 利用状況表示
8. 利用の手引き

1. システム構成

OCTOPUSシステムは、フロントエンドサーバ、計算ノード群、大容量ストレージで構成しています。



1. システム構成

OCTOPUSシステムの特徴

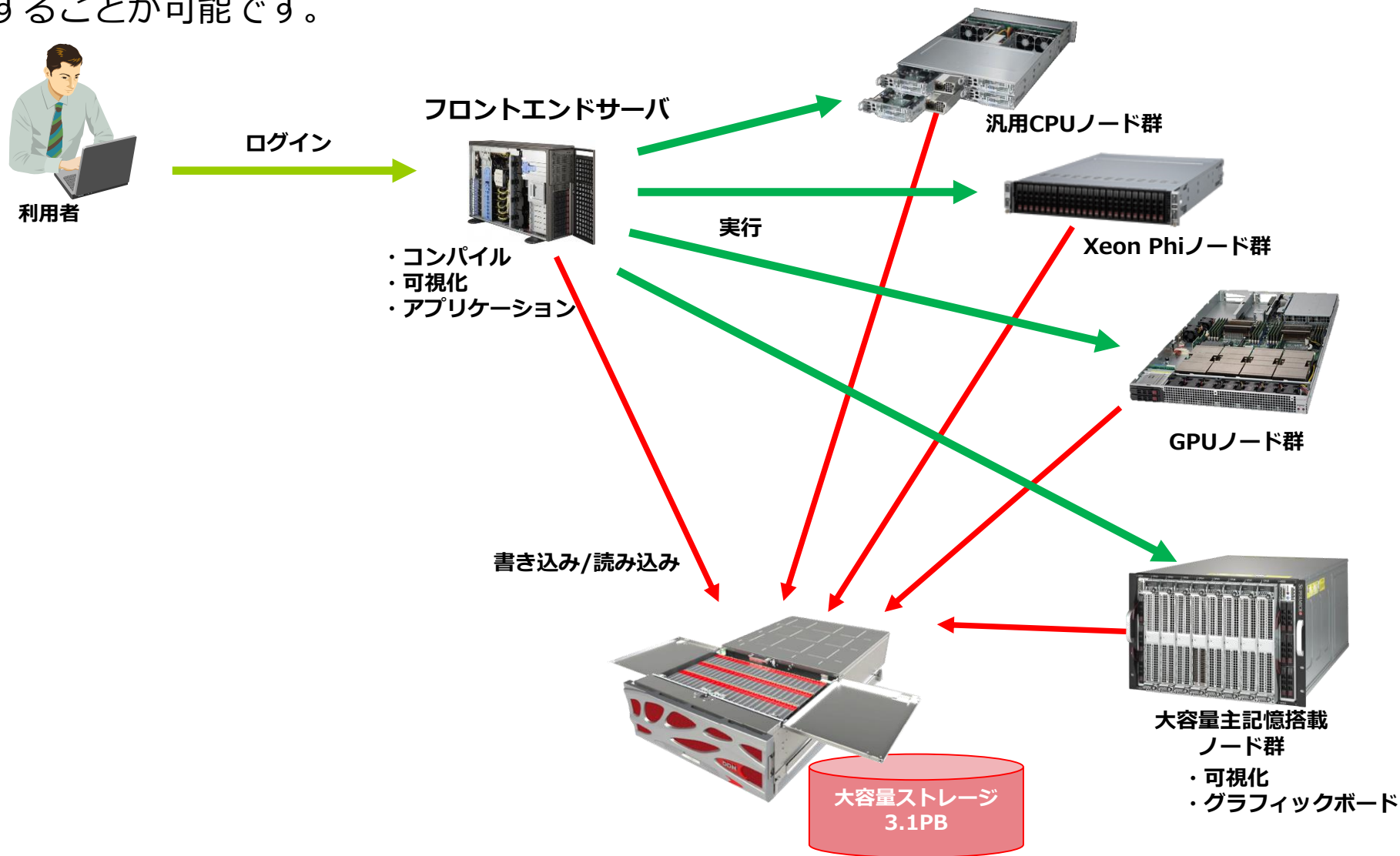
- 計算ノードには汎用CPUのXeonプロセッサ(Skylake)、メニーコアCPUのXeon Phiプロセッサ(Knights Landing)、GPU(NVIDIA Tesla P100)を搭載したサーバの取り揃え、総理論最大性能が1.463PFlops。
- フロントエンドサーバ、計算ノード群、大容量ストレージはすべて1台のスイッチにInfiniband EDR (100Gbps)で直収することで、すべてのノード間通信で0.5マイクロ秒未満の低レイテンシーを実現。
- 外部との通信は20Gbpsで接続、高速通信が可能。

アジェンダ

1. システム構成
2. 利用概要
3. 計算ノード群
4. 大容量ストレージ
5. ソフトウェア環境
6. OCTOPUSの利用方法
7. 利用状況表示
8. 利用の手引き

2. 利用概要

フロントエンドサーバにログインしてプログラム開発、計算ノードでプログラム実行することが可能です。

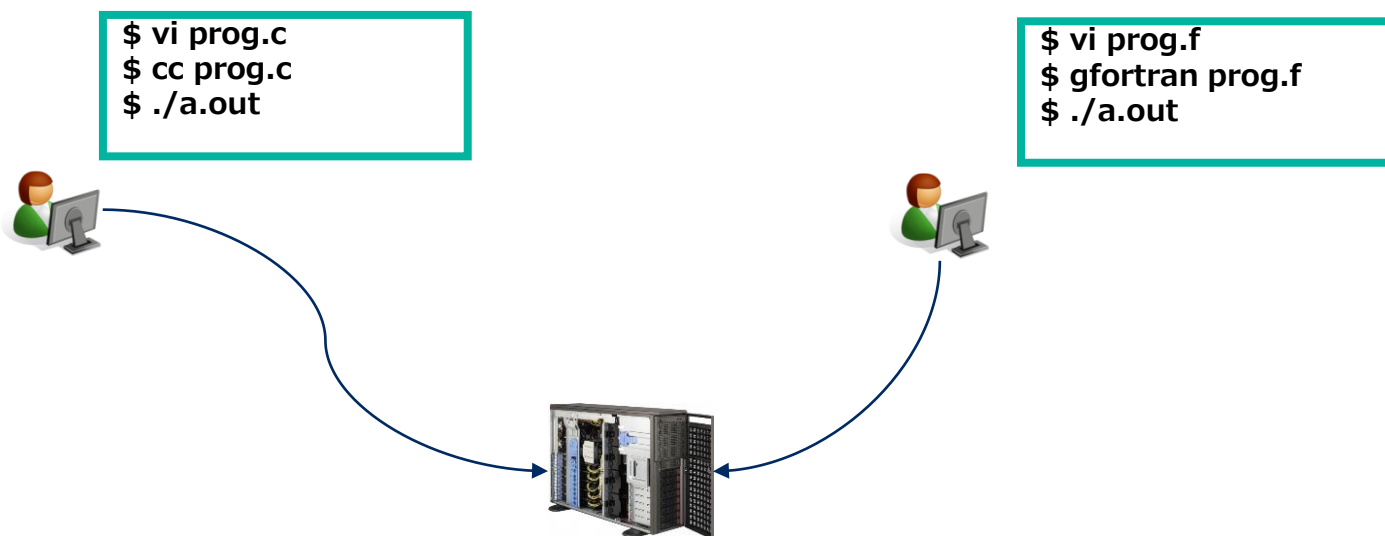


2. 利用概要

コンピューターの利用形態

利用の形態 ～対話型実行～

フロントエンドサーバにログインし、コマンドプロンプトから対話的にコマンドを実行する方法が「対話型実行」です。



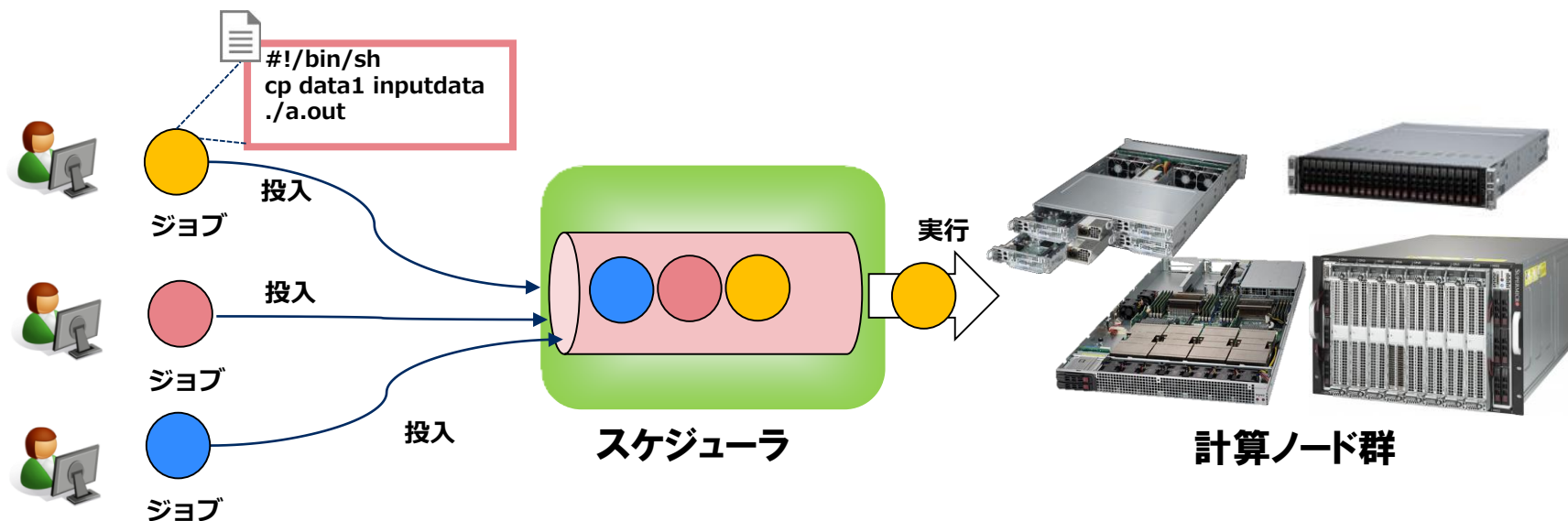
多数のユーザーが同時にプログラムを実行すると、CPU/メモリ等の資源不足や競合が発生することで、実行時間が延びる場合があります。

2. 利用概要

コンピューターの利用形態

利用の形態 ～バッチ型実行～

実行する一連のコマンドをシェルスクリプト化し、システムに実行指示して非対話的で、かつシステムが空きリソースを見て、順番制御、実行管理する方法が「バッチ型実行」です。



対話型実行のような即時実行性はありませんが、CPUやメモリ等の資源競合が起こらないように実行管理します。そのため、多くの方が同時に利用される場合においても、資源不足・競合が発生せず、同じプログラムの実行時間差異が少なくなります。

2. 利用概要

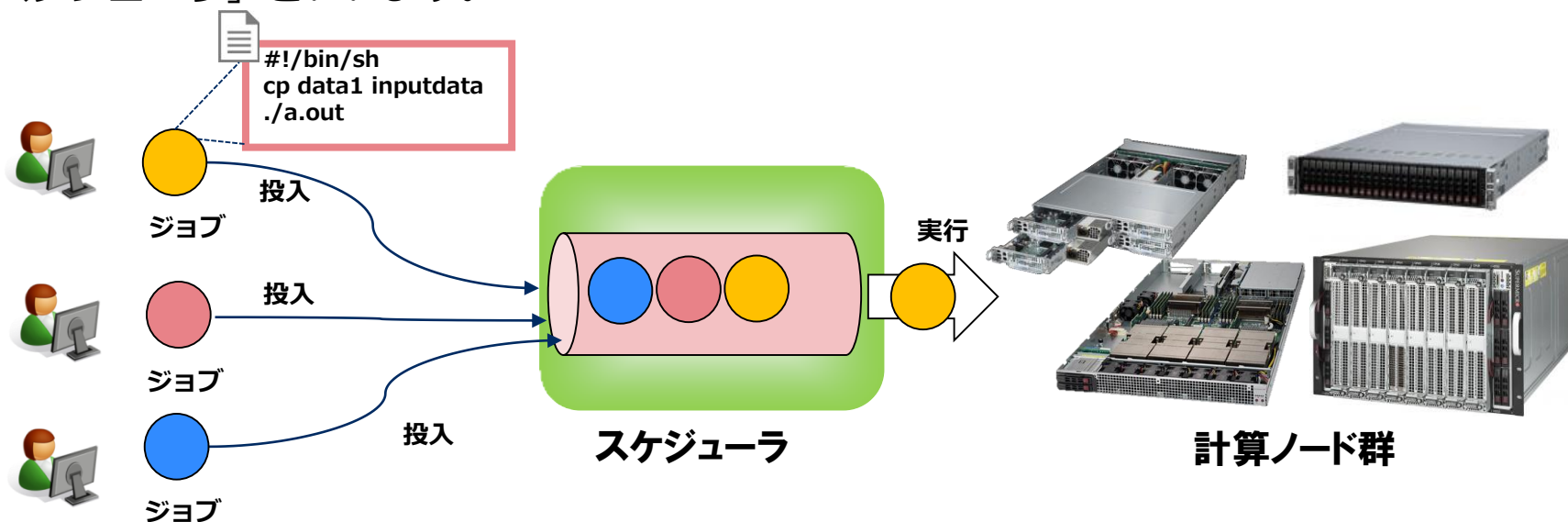
ジョブとスケジューラ

ジョブ

「バッチ型実行」の利用形態において、計算ノードでプログラム実行するためのスクリプトを「ジョブ」、ユーザが「ジョブ」をシステムに渡すことを「ジョブ投入」といいます。

スケジューラ

- 投入されたジョブの順番制御、空リソースの状況に応じて適時実行するシステムを「スケジューラ」といいます。



アジェンダ

1. システム構成
2. 利用概要
3. 計算ノード群
4. 大容量ストレージ
5. ソフトウェア環境
6. OCTOPUSの利用方法
7. 利用状況表示
8. 利用の手引き

3. 計算ノード群

計算ノード群全体構成

システム全体

CPUノード群理論総演算性能

GPU理論総演算性能

Xeon Phiノード群理論総演算性能

大容量主記憶搭載ノード理論総演算性能

1.463PFlops

545.1TFlops

784.4TFlops

117.1TFlops

16.0TFlops

● 汎用CPU計算ノード群 236ノード

- CPU : Intel Xeon Gold 6126 x2
1.9968 TFlops/node
- メモリ : 192 GB/node



● Xeon Phiノード群 44ノード

- メニーコア型CPU : Intel Xeon Phi 7210 x1
2.662TFlops/node
- メモリ : 192 GB/node



● GPUノード群 37ノード

- CPU : Intel Xeon Gold 6126 x2, 1.9968TFlops/node
- GPU : NVIDIA Tesla P100 x4, 21.2Tflops/node
- メモリ : 192 GB/node



● 大容量主記憶搭載ノード群 2ノード

- CPU : Intel Xeon Platinum 8153 x8
8.192TFlops/node
- メモリ : 6 TB/node



3. 計算ノード群

汎用CPUノード群構成サーバ

NEC LX 2U Twin2サーバ 406Rh-2

- 2Uシャーシに4ノードを搭載する高密度実装サーバ
- CPUにIntel Xeon Gold 6126（開発コード：Skylake）を採用
- InfiniBand EDRによる100Gbpsの高速インターコネクト
- 冷却方式には、水冷方式を採用

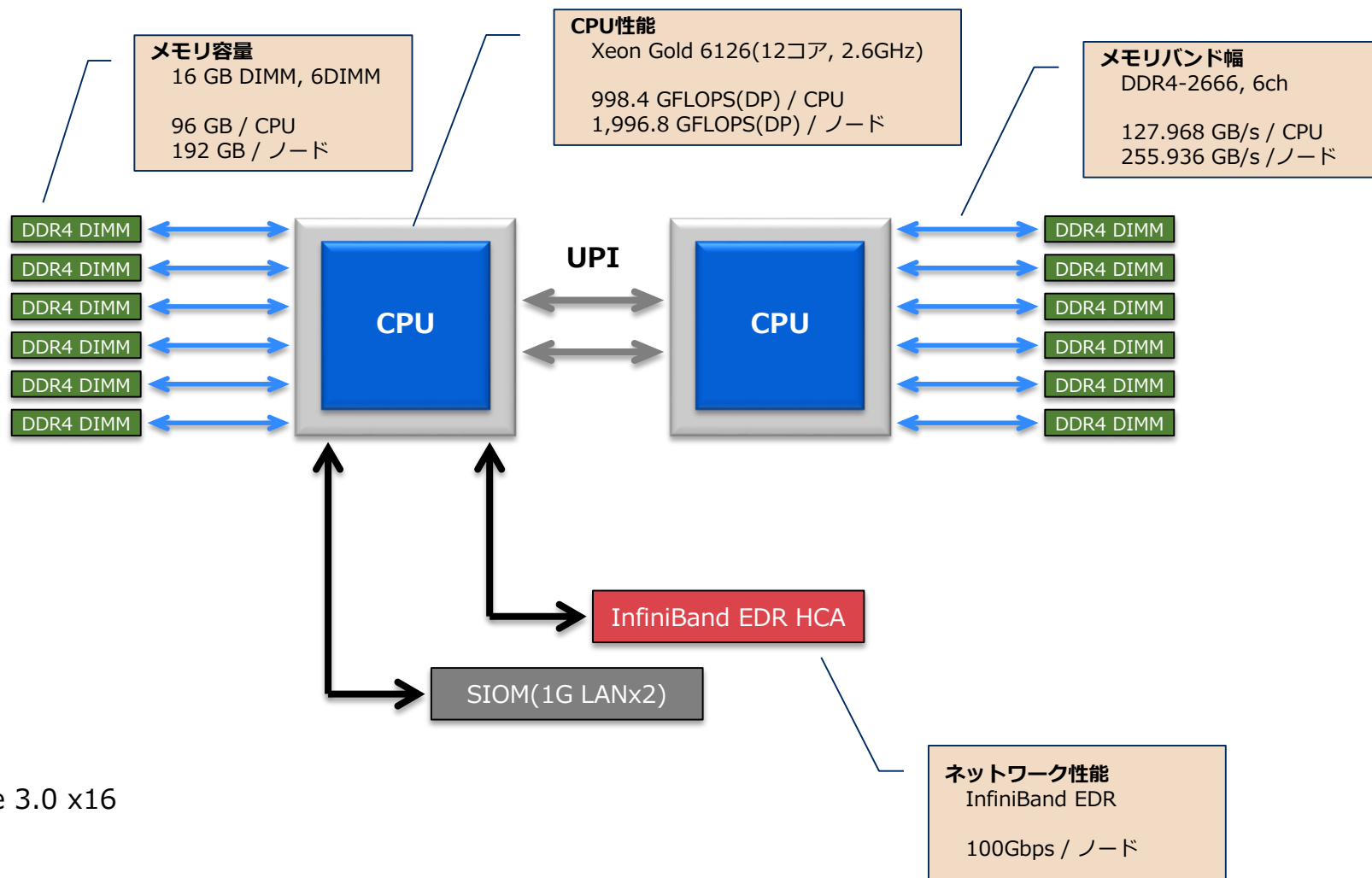


主な仕様

モデル		406Rh-2
CPU (1ノードあたり)	搭載CPU	Intel® Xeon® Gold 6126 プロセッサ
	動作周波数	2.6GHz
	キャッシュ(L3)	19.25 MB
	搭載数	2
コア(CPUあたり)	搭載数	12
メモリ (1ノードあたり)	搭載容量	192GB
	搭載メモリ	DDR4-2666
Infiniband		EDR Infiniband HCA
OS		Red Hat Enterprise Linux 7 HPC Compute Node

3. 計算ノード群

汎用CPUノード システムブロック図



3. 計算ノード群

GPUノード群構成サーバ

NEC LX 1U 4GPUサーバ 102Rh-1G

- 1UシャーシにCPU 2基、GPU 4基を搭載する高密度実装サーバ
- CPUにIntel Xeon Gold 6126（開発コード：Skylake）を搭載
- NVLinkに対応したNVIDIA社のTesla P100を搭載
- InfiniBand EDRによる100Gbpsの高速インターコネクト
- 冷却方式には、水冷方式を採用

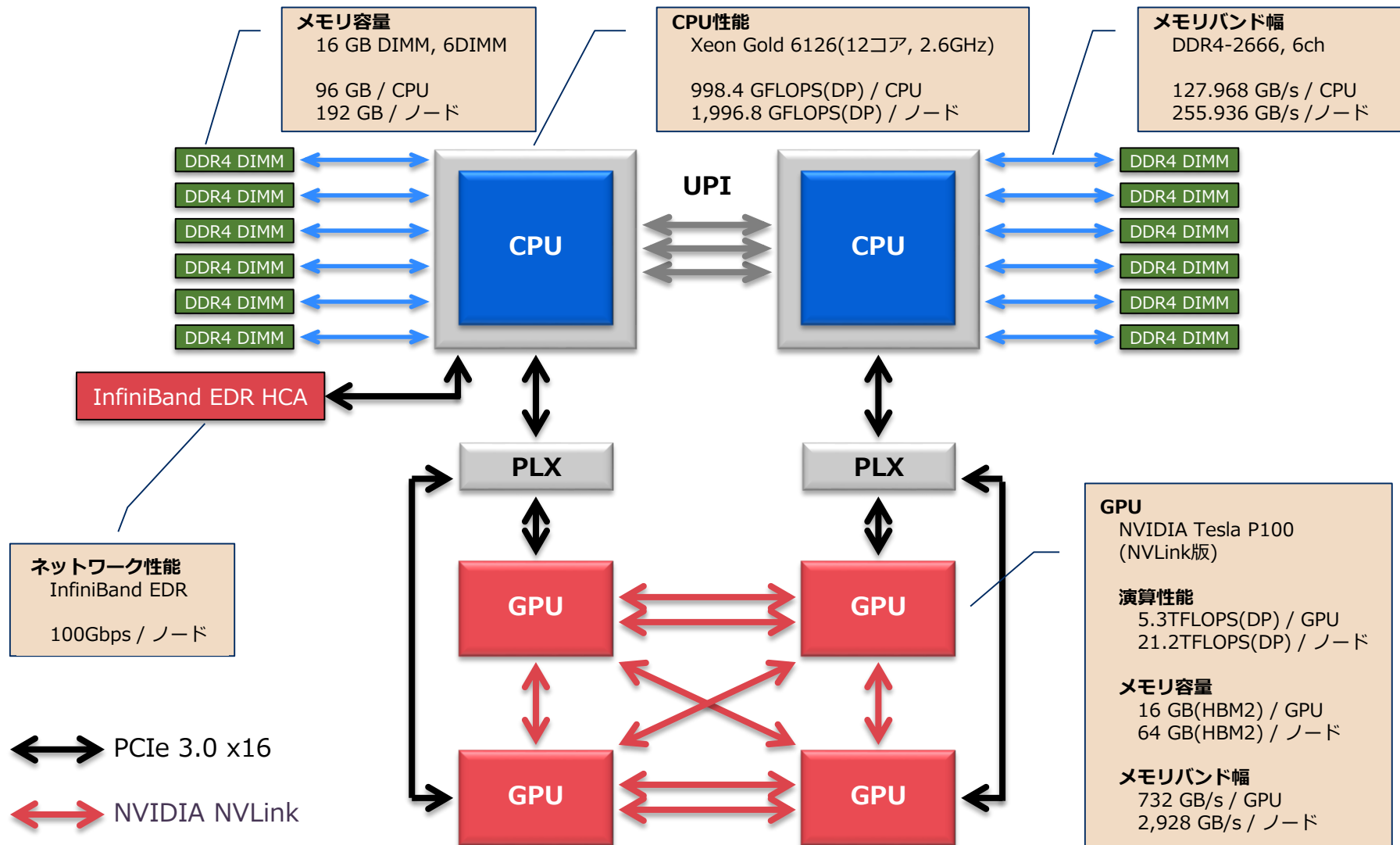


主な仕様

モデル		LX 102Rh-1G
CPU (1ノードあたり)	搭載CPU	Intel® Xeon® Gold 6126 プロセッサ
	動作周波数	2.6GHz
	キャッシュ(L3)	19.25 MB
	搭載数	2
コア(CPUあたり)	搭載数	12
	搭載容量	192GB
メモリ (1ノードあたり)	搭載メモリ	DDR4-2666
	搭載数	4
GPU (1ノードあたり)	NVIDIA® Tesla® P100 for NVLink Optimized	
	FP64	5.3TFLOPS
	Memory	16GB CoWoS HBM2 at 732 GB/s
	搭載数	4
Infiniband		EDR Infiniband HCA
OS		Red Hat Enterprise Linux 7 HPC Compute Node

3. 計算ノード群

GPUノード システムブロック図



3. 計算ノード群

Xeon Phiノード群構成サーバ

NECメニーコアサーバ Express5800/HR110c-M

- 2Uシャーシに4ノードを搭載する高密度実装サーバ
- CPUにIntel Xeon Phi 7210（開発コード：Knights Landing）を採用
- メモリは、DDR4 192GBに加え、MCDRAM 16GB(flatモード)を搭載
- InfiniBand EDRによる100Gbpsの高速インターコネクト
- 冷却方式には、水冷方式を採用

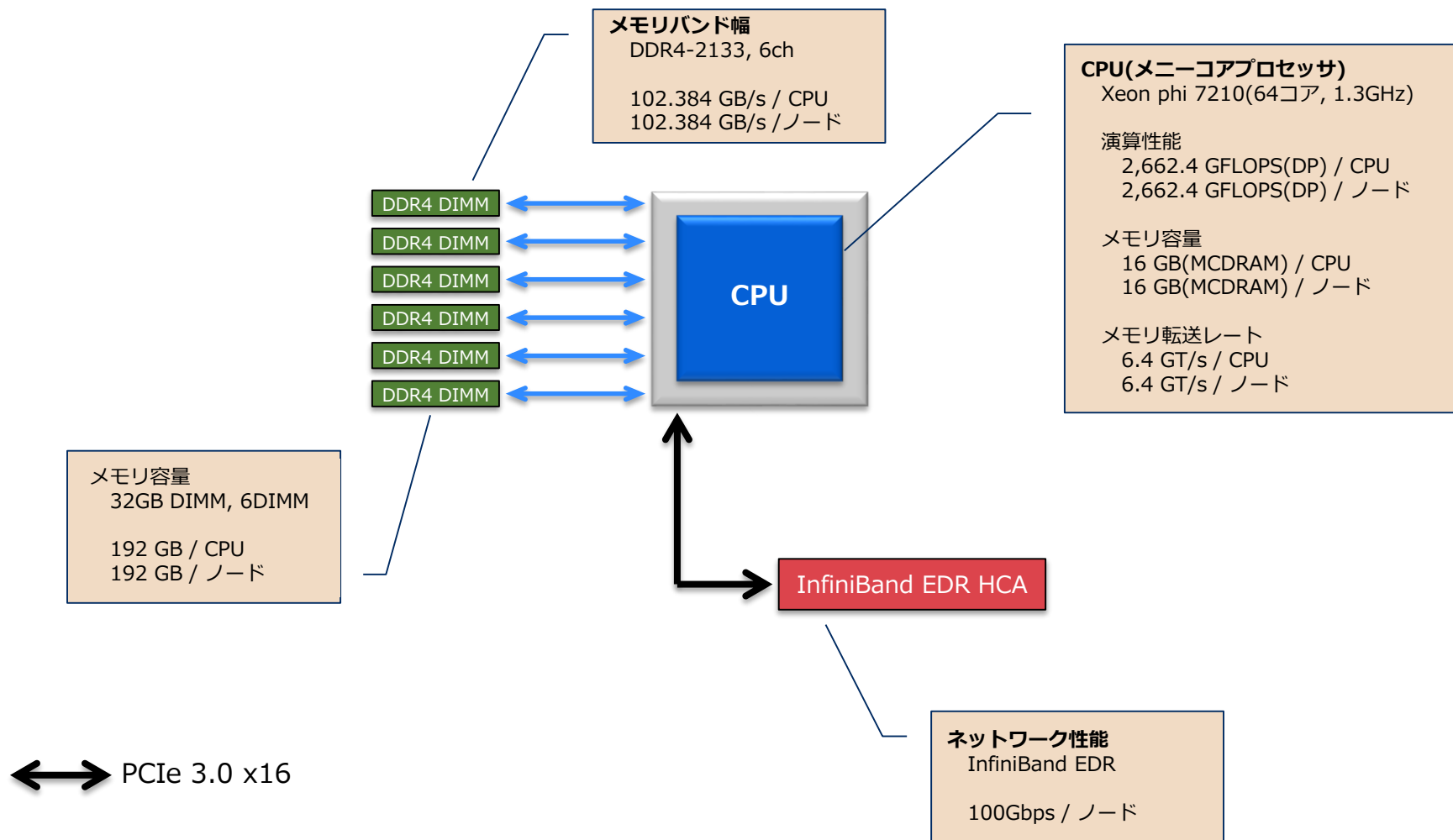
主な仕様

モデル		406Rh-2
CPU (1ノードあたり)	搭載CPU	Intel® Xeon® Phi 7210 プロセッサ
	動作周波数	1.3GHz
	MCDRAM	16 GB
	搭載数	1
コア(CPUあたり)	搭載数	64
メモリ (1ノードあたり)	搭載容量	192GB
	搭載メモリ	DDR4-2133
メモリ (1ノードあたり)	搭載容量	16GB
		MCDRAM
Infiniband		EDR Infiniband HCA
OS		Red Hat Enterprise Linux 7 HPC Compute Node



3. 計算ノード群

Xeon Phiノード システムブロック図



3. 計算ノード群

大容量主記憶搭載ノード群構成サーバ

大容量主記憶搭載 8way サーバ LX 116Rh-7

- CPUにIntel Xeon Platinum 8153（開発コード：Skylake）を8基搭載
- 6TBの大容量メモリを搭載
- InfiniBand EDRによる100Gbpsの高速インターコネクト



主な仕様

モデル		116Rh-7
CPU	搭載CPU	Intel® Xeon® Platinum 8153 プロセッサ
	動作周波数	2.0GHz
	キャッシュ(L3)	22 MB
	搭載数	8
コア(CPUあたり) 搭載数		16
メモリ (1ノードあたり)	搭載容量	6 TB
	搭載メモリ	DDR4-2666
Infiniband		EDR Infiniband HCA
OS		Red Hat Enterprise Linux 7 HPC Compute Node

3. 計算ノード群

大容量主記憶搭載ノード システムブロック図(CPU/メモリ)

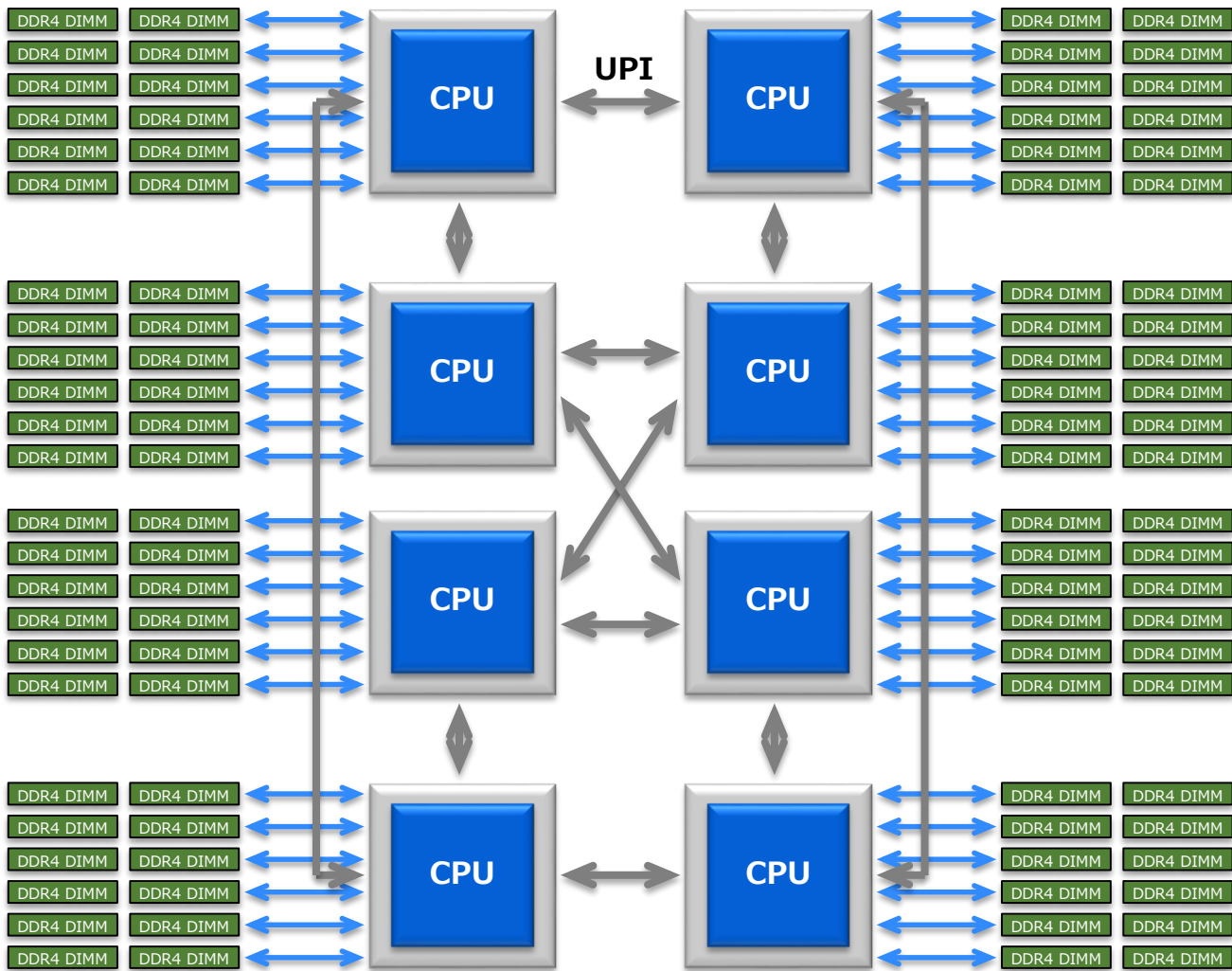
CPU性能
Xeon Platinum 8153
(16コア, 2.0GHz)
1,024 GFLOPS(DP) / CPU
8,192 GFLOPS(DP) / ノード

メモリ容量
64GB DIMM, 12DIMM

768 GB / CPU
6144 GB / ノード

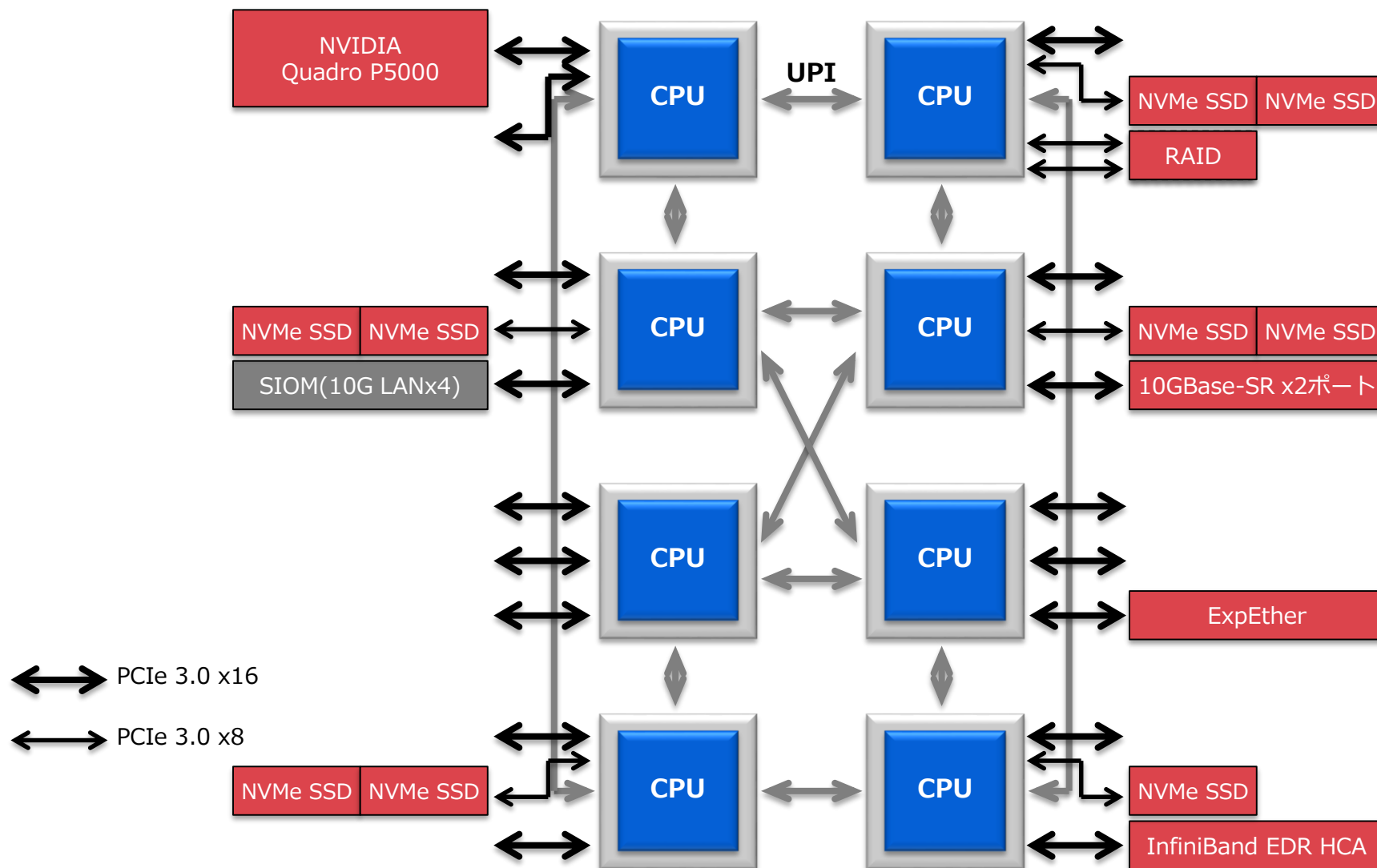
メモリバンド幅
DDR4-2666, 6ch

127.968 GB/s / CPU
1023.744 GB/s / ノード



3. 計算ノード群

大容量主記憶搭載ノード システムブロック図(I/O)



3. 計算ノード群

フロントエンドサーバ

NEC LXシリーズ 4U GPUサーバ 108Th-4G

- CPUにIntel Xeon Gold 6126（開発コード：Skylake）を搭載
- InfiniBand EDRによる100Gbpsの高速インターコネクト
- 10Gbit Ethernet × 2 でODINSと接続、20Gbpsの高速通信

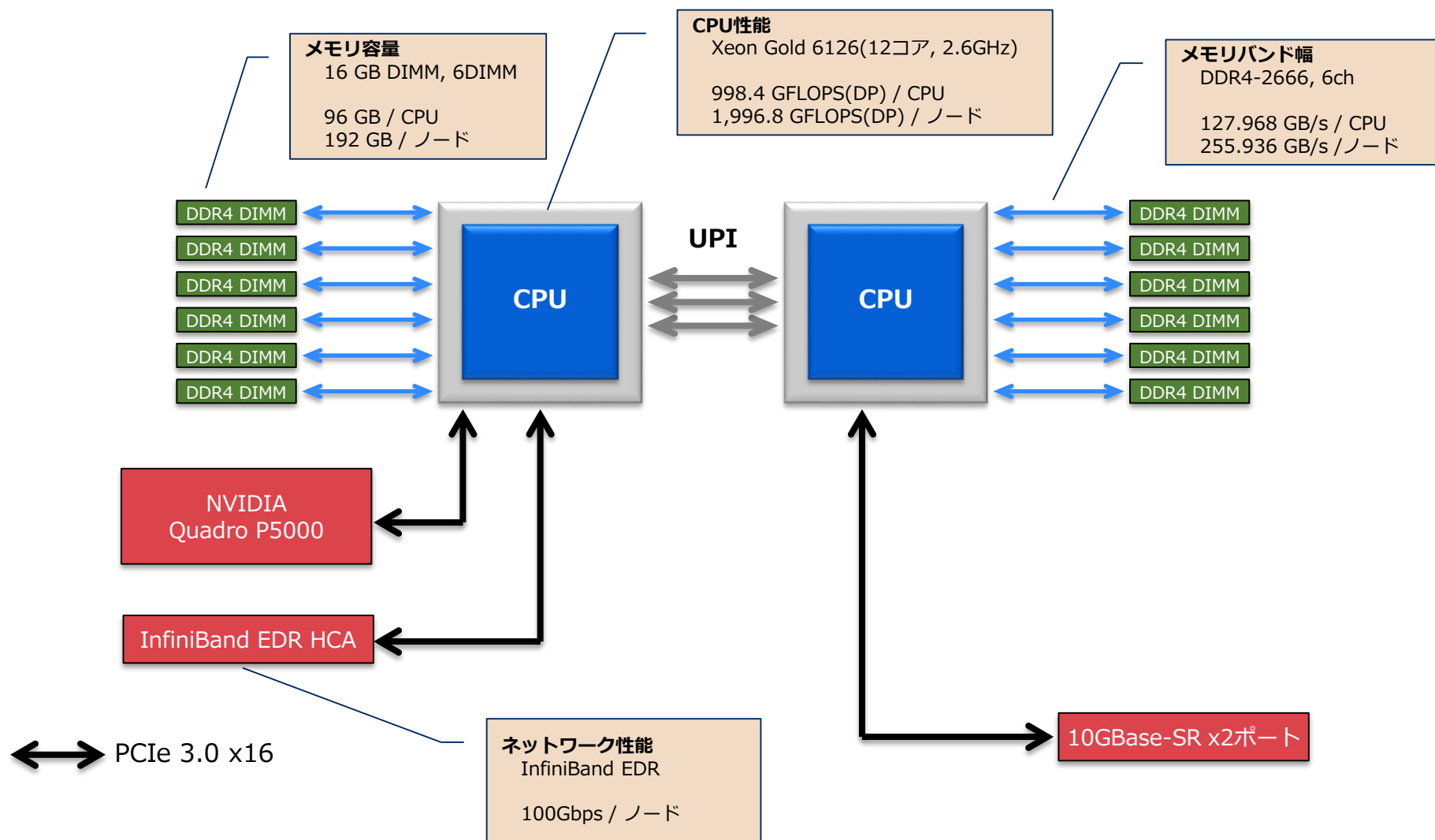


主な仕様

モデル		LX 108Th-4G
CPU (1ノードあたり)	搭載CPU	Intel® Xeon® Gold 6126 プロセッサ
	動作周波数	2.6GHz
	キャッシュ(L3)	19.25 MB
	搭載数	2
コア(CPUあたり)	搭載数	12
メモリ (1ノードあたり)	搭載容量	192GB
	搭載メモリ	DDR4-2666
グラフィックカード		NVIDIA® Quadro P5000
Infiniband		EDR Infiniband HCA
ネットワークインタフェース		2x 10GBASE-SR
OS		Red Hat Enterprise Linux Server 7

3. 計算ノード群

フロントエンドサーバシステムブロック図



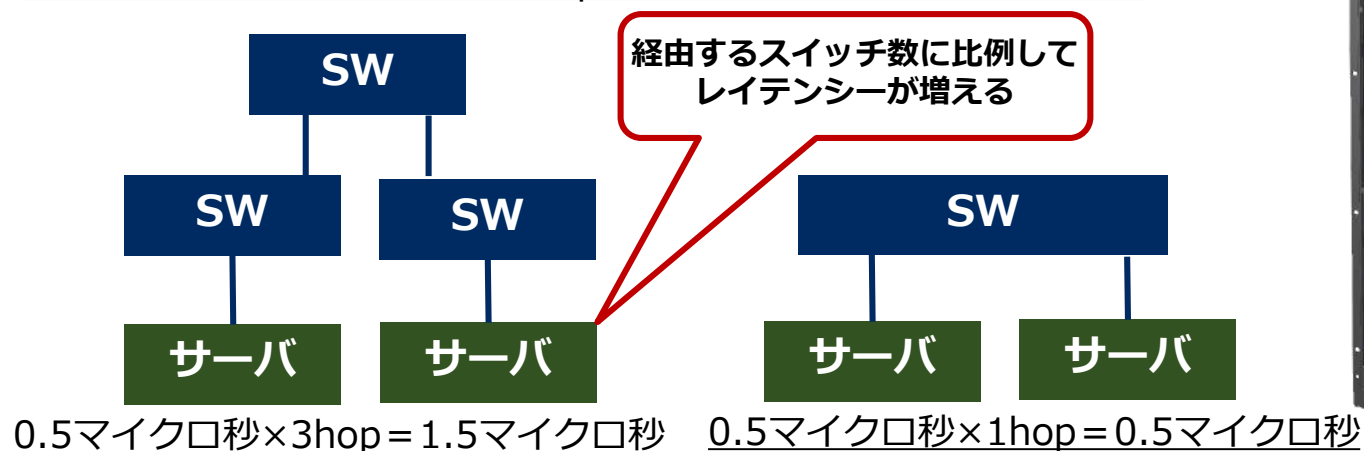
3. 計算ノード群

インターコネクト

Mellanox CS7500

- Infiniband EDR 100Gbps用のスイッチ
- すべてのノードを1つのスイッチに直接接続
- 72Tbpsのノン・ブロッキング帯域幅
- 0.5マイクロ秒未満のポート間レイテンシー

例) ポート間レイテンシーが1hop当たり0.5マイクロ秒とした場合



アジェンダ

1. システム構成
2. 利用概要
3. 計算ノード群
4. 大容量ストレージ
5. ソフトウェア環境
6. OCTOPUSの利用方法
7. 利用状況表示
8. 利用の手引き

4. 大容量ストレージ

ストレージシステム

DDN EXAScaler (Lustre)

- 3.1PB(2.8PiB)と大容量の高速並列ファイルシステム
- 計算ノードと同様にInfiniband EDR 100Gbps 接続
- すべてのフロントエンドサーバ、計算ノードは同一のファイルシステムにアクセス可能

PB : $1,000^5$ バイト

PiB : $1,024^5$ バイト

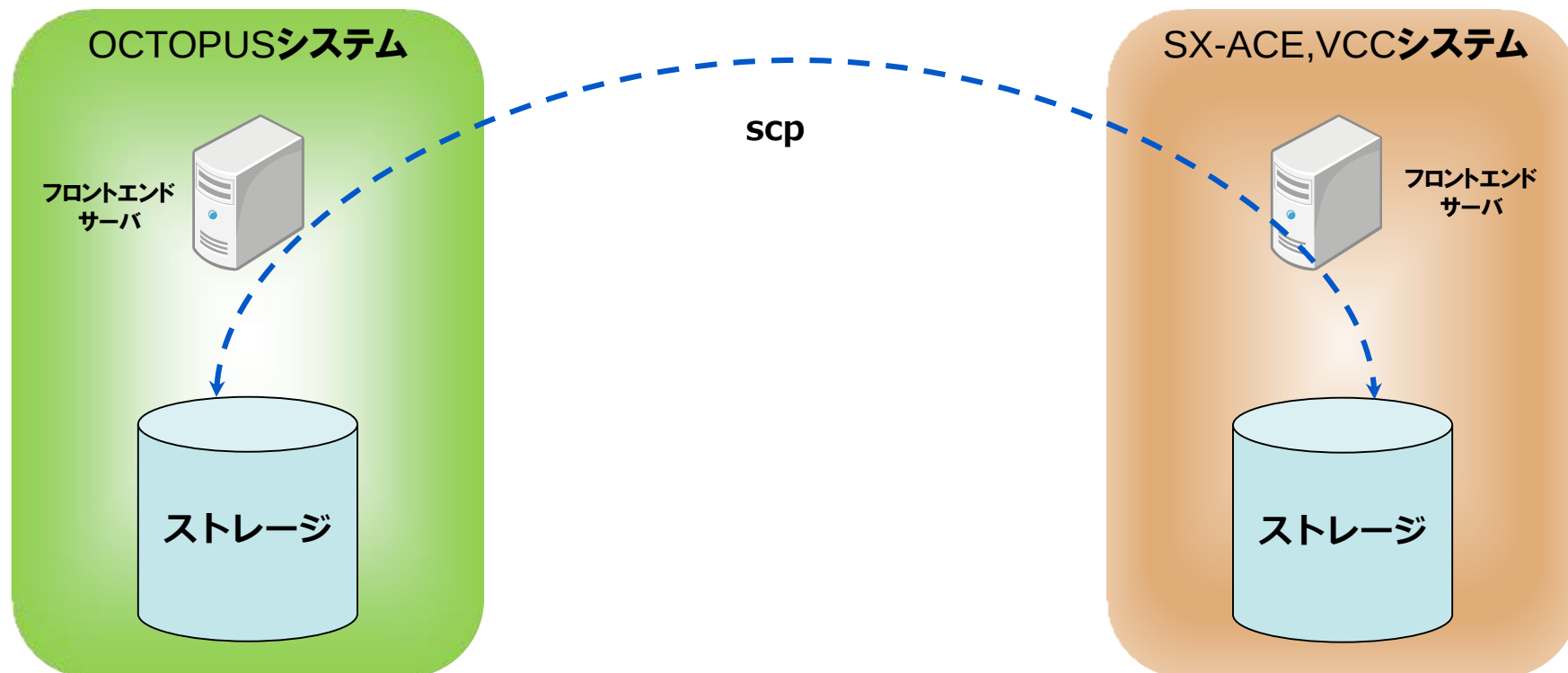


4. 大容量ストレージ

OCTOPUSとSX-ACE,VCCシステムとのファイル連携

ファイルの相互連携

- OCTOPUSとSX-ACE,VCCとではファイルシステムが別々
- 両システム間で「scpコマンド」等を使用してファイルコピー可能



アジェンダ

1. システム構成
2. 利用概要
3. 計算ノード群
4. 大容量ストレージ
5. ソフトウェア環境
6. OCTOPUSの利用方法
7. 利用状況表示
8. 利用の手引き

5. ソフトウェア環境

利用可能アプリケーション

アプリケーション名	バージョン	GPU対応
CTFFIND	4.1.8	
Digits	6.0.0	○
GAMESS	2017 Sept 30 R2	
Gaussian16	Rev. A03, Rev. B01	○ (Rev.B01)
GROMACS	2016.1	○
LAMMPS	20170811	○
MotionCor2	v1.0.5	
OpenFOAM	4.1	
Relion	2.1	○
ResMap	1.1.4	
VisIT	2.13.0.l	

5. ソフトウェア環境

■ 利用可能ライブラリ

ライブラリ名	バージョン
GNU Scientific Library	1.15-13
HDF5	5-1.10.1
Intel MKL	Parallel Studio XE Cluster Edition 2017
NetCDF	4.4.1
Parallel netcdf	1.8.1

5. ソフトウェア環境

利用可能コンパイラ、インタプリタ、フレームワーク等

ライブラリ名	バージョン
Caffe	1.0.0
Chainer	3.3.0
CUDA	9.0
FreeFem++	3.56-1
GCC	4.8.5, 4.9.4
Intel Compiler	Parallel Studio XE Cluster Edition 2017
Julia	0.6.1
MVAPICH2	2.2.2
Octave	3.8.2
PGI Compiler	17.9
Python	2.7.5, 3.4.5, 3.5.2
R	3.4.2
Tensorflow	1.4.1
Theano	0.9.0
Torch	7
XcalableMP	1.2.1

アジェンダ

1. システム構成
2. 利用概要
3. 計算ノード群
4. 大容量ストレージ
5. ソフトウェア環境
6. OCTOPUSの利用方法
7. 利用状況表示
8. 利用の手引き

6. OCTOPUSの利用方法

OCTOPUS利用の流れ

■ OCTOPUSでのジョブ利用の流れは以下の通りです。

- フロントエンドサーバへログイン



- フロントエンドサーバでプログラムのコンパイル



- ジョブスクリプトの準備



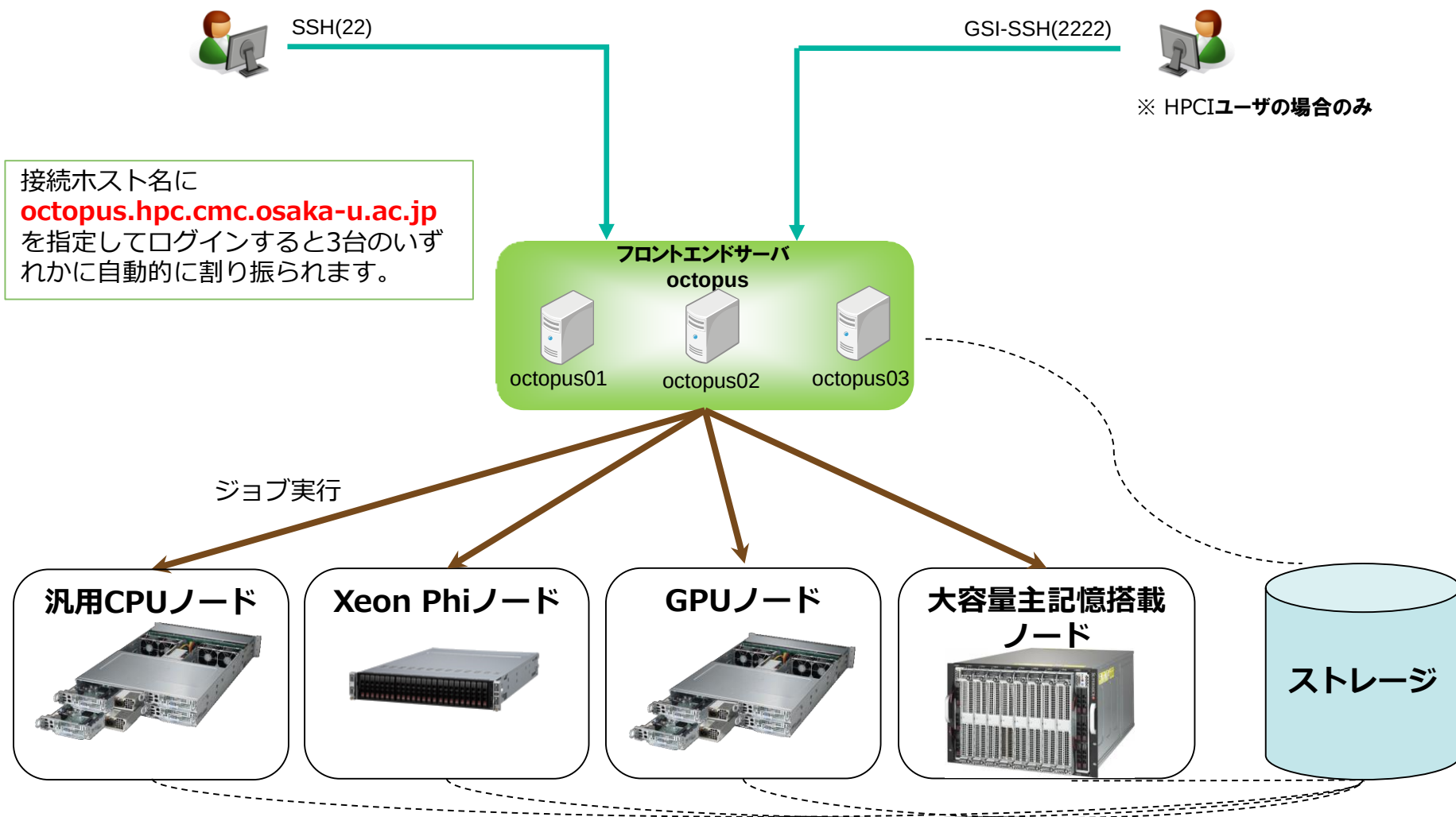
- ジョブ投入



- 結果の確認

6. OCTOPUSの利用方法

フロントエンドサーバへログイン ～利用イメージ～

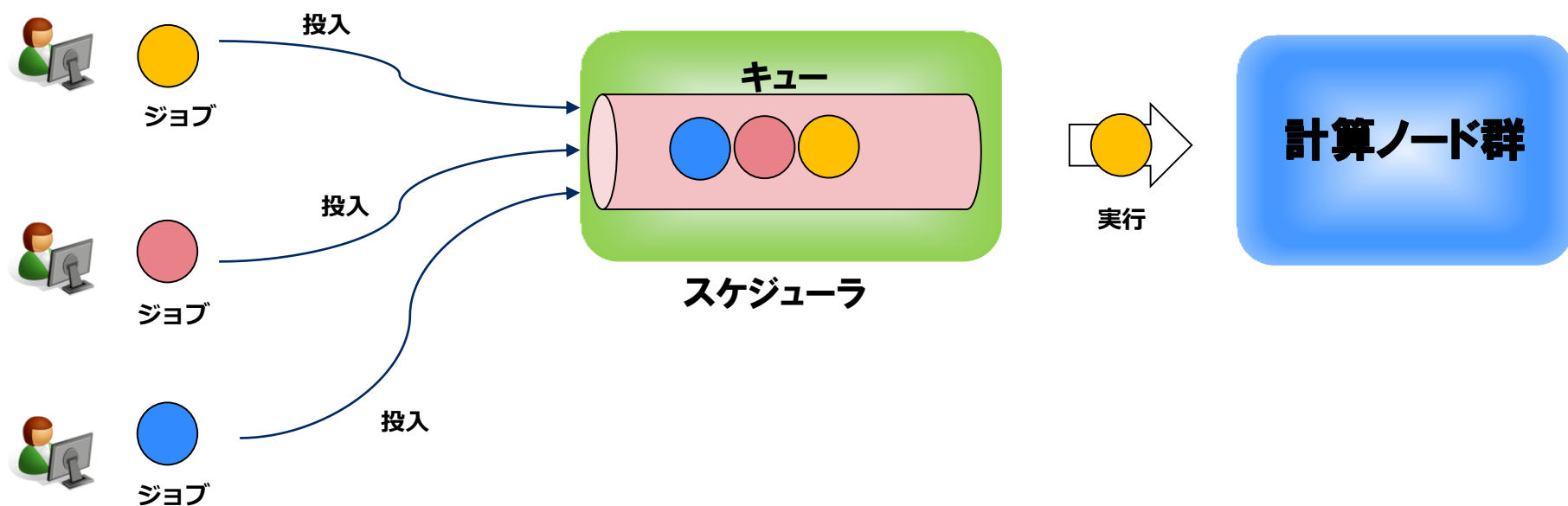


6. OCTOPUSの利用方法

利用イメージ

フロントエンドサーバでスケジューラにジョブ投入することで、スケジューラが空きリソース状況確認し、自動で実行を開始します。

原則、投入された順で実行開始しますが、ジョブ投入時に指定した必要リソースとリソースの空き状況により、実行順が前後する場合があります。



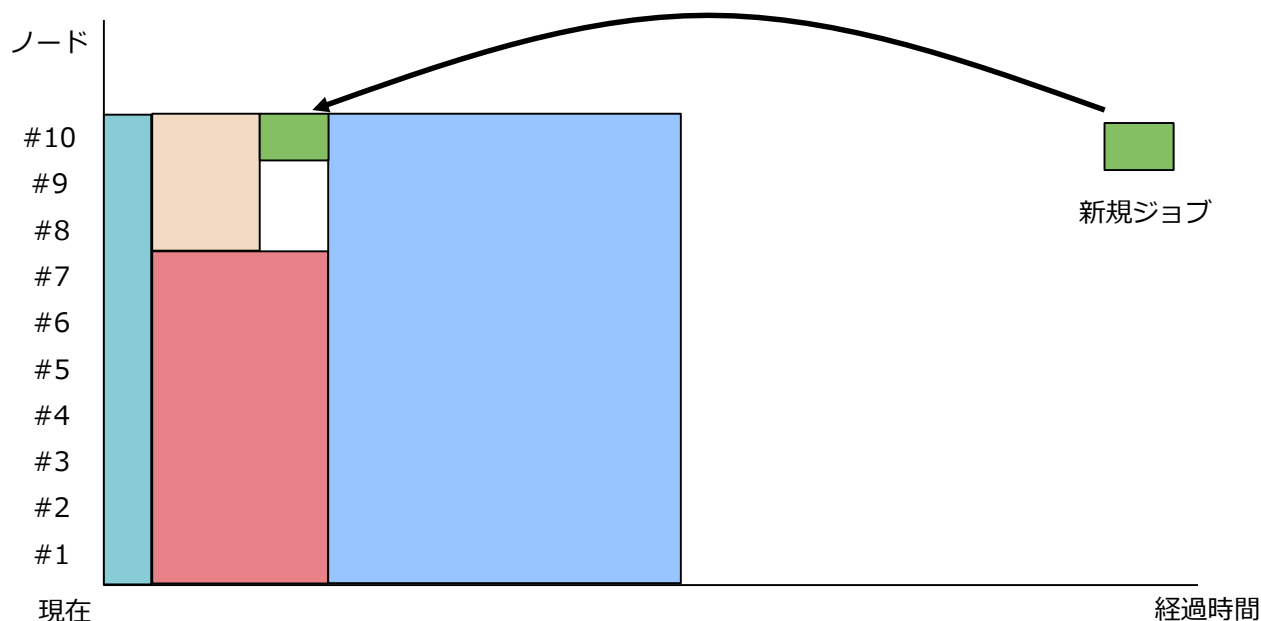
6. OCTOPUSの利用方法

ジョブの実行スケジュール

スケジューラはジョブの大きさを「ノード数」×「経過時間」で表し、未来時間に向かってスケジュール(アサイン)していきます。

新規ジョブが投入された場合に、スケジューラはそのジョブの大きさをリソースの空きを確認し、最短で実行可能な日時でアサインします。

以下の例では、アサイン済みのジョブの実行終了と次のジョブの実行開始までの間に空き状態があり、新規ジョブが先行して投入されたジョブを追い越して実行するスケジュールがされています。



6. OCTOPUSの利用方法

ジョブスクリプトの準備(1)

ジョブスクリプトの記述イメージ

```
#!/bin/sh  
#PBS -q ①  
#PBS -l cpunum_job=②,elapstim_req=③
```

- ① キュー名の指定
- ② 使用するコア数の指定（既定値は1ノードのフルコア数）
- ③ 使用計算時間（経過時間）の指定（HH:MM:SS形式で指定）

6. OCTOPUSの利用方法

ジョブスクリプトの準備(2)

汎用CPUノードのシングルノード用のジョブスクリプト例

```
#!/bin/sh  
#PBS -q OCTOPUS  
#PBS -l cpunum_job=①,elapstim_req=②
```

- ① 使用するCPUコア数の指定（範囲：1～24, 既定値：24）
- ② 使用計算時間（経過時間）の指定（HH:MM:SS形式で指定）

6. OCTOPUSの利用方法

ジョブスクリプトの準備(3)

GPUノードのシングルノード用のジョブスクリプト例

```
#!/bin/sh  
#PBS -q OCTOPUS  
#PBS -l cpunum_job=①,gpunum_job=②,elapstim_req=③
```

- ① 使用するCPUコア数の指定 (範囲：1～24, 既定値：24)
- ② 使用するGPU数の指定 (範囲：1～4, 既定値：なし)
- ③ 使用計算時間 (経過時間) の指定 (HH:MM:SS形式で指定)

6. OCTOPUSの利用方法

ジョブスクリプトの準備(4)

Xeon Phiノードのシングルノード用のジョブスクリプト例

```
#!/bin/sh  
#PBS -q OCTPHI  
#PBS -l cpunum_job=①,elapstim_req=②
```

- ① 使用するCPUコア数の指定 (範囲：1～64, 既定値：64)
- ② 使用計算時間（経過時間）の指定 (HH:MM:SS形式で指定)

6. OCTOPUSの利用方法

ジョブスクリプトの準備(5)

■ 大容量主記憶搭載ノードのシングルノード用のジョブスクリプト例

```
#!/bin/sh  
#PBS -q OCTMEM  
#PBS -l cpunum_job=①,elapstim_req=②
```

- ① 使用するCPUコア数の指定 (範囲 : 1~128, 既定値 : 128)
- ② 使用計算時間 (経過時間) の指定 (HH:MM:SS形式で指定)

6. OCTOPUSの利用方法

ジョブスクリプトの準備(6)

汎用CPUノードの複数ノード用のジョブスクリプト例

```
#!/bin/sh
#PBS -q OCTOPUS
#PBS -l cpunum_job=①,elapstim_req=②
#PBS -T intmpi
#PBS -b ③
```

- ① 使用するCPUコア数の指定（範囲：1～24, 既定値：24）
注) 1ノード内で使用するコア数を指定します。
- ② 使用計算時間（経過時間）の指定（HH:MM:SS形式で指定）
- ③ 使用するノード数の指定（範囲：1～256）

6. OCTOPUSの利用方法

ジョブの投入

qsubコマンドでジョブスクリプトを投入します。

(例) CPUコア数を「1」、実行時間(ElapseTime)を「2分」で「Hello, world!」を表示するプログラム「a.out」を実行する。

```
$ cat cpujob1
#!/bin/sh
#PBS -q OCTOPUS
#PBS -l cpunum_job=1, elapstim_req=00:02:00

./a.out

$ qsub cpujob1
Request 123456.oct submitted to queue: OCTOPUS.
$
```

投入に成功した際のメッセージ

リクエストID

6. OCTOPUSの利用方法

ジョブの投入（複数ノード使用の場合）

qsubコマンドでジョブスクリプトを投入します。

(例) CPUコア数を「24」、実行時間(ElapseTime)を「10分」、使用ノード数「2」で IntelMPIでコンパイルしたプログラム「a.out」を実行する。

```
$ cat cpujob2
#!/bin/sh
#PBS -q OCTOPUS
#PBS -l cpunum_job=24, elapstim_req=00:10:00
#PBS -T intmpi
#PBS -b 2

mpirun ${NQSII_MPIOPTS} -np 48 ./a.out

$ qsub cpujob2
Request 123456.oct submitted to queue: OCTOPUS.
$
```

cpunum_jobで指定するコア数はノード内で使用するコア数を指定

mpirunで指定するコア数(-np)はプログラム実行で使用する全コア数を指定

MPI実行の場合は、「\${NQSII_MPIOPTS}」を記述が必須

6. OCTOPUSの利用方法

ジョブの投入確認

ジョブ投入が成功したか確認します。

```
$ qstat
RequestID      ReqName  UserName Queue      Pri STT S   Memory      CPU   Elapse R H M Jobs
-----
123456. oct    cpujob1  v6a000  OC1C         0 QUE -    0.00B      0.00      0 Y Y Y 1
```

投入に成功していれば表示される

実行開始後の状態

```
$ qstat
RequestID      ReqName  UserName Queue      Pri STT S   Memory      CPU   Elapse R H M Jobs
-----
123456. oct    cpujob1  v6a000  OC1C         0 RUN -    4.0MB     10.00     10 Y Y Y 1
```

実行が開始されると「**RUN**」になる

6. OCTOPUSの利用方法

ジョブの状況確認

ジョブのスケジュール状況を確認します。

投入直後

```
$ sstat
RequestID      ReqName  UserName Queue          Pri STT PlannedStartTime
-----
123456. oct    cpujob1  v6a000  0C1C          0. 5002/ 0. 5002 QUE -
```

実行日時が決定した状態

```
$ sstat
RequestID      ReqName  UserName Queue          Pri STT PlannedStartTime
-----
123456. oct    cpujob1  v6a000  0C1C          0. 5002/ 0. 5002 ASG 2018-06-14 16:01:01
```

実行開始された状態

```
$ sstat
RequestID      ReqName  UserName Queue          Pri STT PlannedStartTime
-----
123456. oct    cpujob1  v6a000  0C1C          0. 5002/ 0. 5002 RUN Already Running...
```

6. OCTOPUSの利用方法

ジョブの実行結果確認

ジョブが終了すると、ジョブ投入した際のカレントディレクトリにジョブの標準出力、エラー出力ファイルが返却されます。

ファイル名を指定しない場合は下記の規則で命名されます。

標準出力ファイル : ジョブスクリプト名.oリクエストIDの数字部分

標準エラー出力ファイル : ジョブスクリプト名.eリクエストIDの数字部分

```
$ ls
a.out  cpujob1  cpujob1.o123456  cpujob1.e123456
$ cat cpujob1.o123456
Hello, world!
$
```


6. OCTOPUSの利用方法

ジョブの削除

不要となったジョブを削除する場合、以下のように実施します。

(例) 実行中の「123456.oct」を削除する

```
$ qstat
RequestID      ReqName  UserName Queue      Pri STT S   Memory    CPU    Elapse R H M Jobs
-----
123456.oct     cpujob1  v6a000  0C1C         0 RUN -    4.0MB     1.23    12 Y Y Y    1
$
$ qdel 123456.oct
Request 123456.oct was deleted.
```

✓ 待ち状態(QUE)の場合でも同様です。

アジェンダ

1. システム構成
2. 利用概要
3. 計算ノード群
4. 大容量ストレージ
5. ソフトウェア環境
6. OCTOPUSの利用方法
7. 利用状況表示
8. 利用の手引き

7. 利用状況表示

利用実績表示コマンド usage_view

自身が所属するグループおよびグループに所属するユーザーの共有利用での利用実績を表示

-----+ Group: G19999 +-----

[Group summary]

	OCTOPUS points	Disk (GiB)
usage	5.5	7.2
limit	1000.0	1024.0
remain	994.5	1016.8
rate (%)	0.6	0.7

所属するグループ全体の利用実績

[Detail]

	OCTOPUS points	Home (GiB)	Work (GiB)
v6a000	0.0	0.0	0.0
v6b000	0.0	0.0	0.0
v6c000	0.0	0.0	0.0
v6d000	5.6	8.3	7.2

所属するグループのユーザごとの利用実績

[Node-hours]

	Usage	Available*
General purpose CPU node	50.32	19124.80
GPU node	0.00	4576.57
Xeon Phi node	71.25	23791.62
Large-scale shared-memory node	0.00	2685.63

* Available Node-hour is estimated based on remains of OCTOPUS point.

残ポイントで利用可能な時間(参考値)

所属するグループ全体の利用実績

所属するグループのユーザごとの利用実績

残ポイントで利用可能な時間(参考値)

7. 利用状況表示

所属グループのジョブ状態表示コマンド `qstatgroup/sstatgroup`

自身が所属するグループおよびグループに所属するユーザーのジョブの状態を表示 (`qstatgroup`)

RequestID	ReqName	UserName	Queue	Pri	STT	S	Memory	CPU	Elapse	R	H	M	Jobs
291581.oct	gpujob10	v6b000	OC16C	0	RUN	-	1.63G	46958024.99	1964672	Y	Y	Y	1
293027.oct	gpujob21	v6a000	OC16C	0	RUN	-	2.82G	12501732.65	611672	Y	Y	Y	1
294071.oct	gpujob22	v6a000	OG16C	0	RUN	-	2.70G	3604797.51	234122	Y	Y	Y	1
294268.oct	gpujob23	v6a000	OG16C	0	RUN	-	2.70G	4356096.14	203672	Y	Y	Y	1
294497.oct	gpujob24	v6a000	OG16C	0	RUN	-	2.69G	3743532.37	166052	Y	Y	Y	1
294532.oct	gpujob25	v6a000	OG16C	0	QUE	-	0.00B	0.00	0	Y	Y	Y	1
294533.oct	gpujob26	v6a000	OG16C	0	QUE	-	0.00B	0.00	0	Y	Y	Y	1

自身が所属するグループおよびグループに所属するユーザーのジョブスケジュールの状態を表示 (`sstatgroup`)

RequestID	ReqName	UserName	Queue	Pri	STT	PlannedStartTime
291581.oct	gpujob10	v6b000	OC16C	0.5002/	0.5002	RUN Already Running...
293027.oct	gpujob21	v6a000	OG16C	0.5002/	0.5002	RUN Already Running...
294071.oct	gpujob22	v6a000	OG16C	0.5002/	0.5002	RUN Already Running...
294268.oct	gpujob23	v6a000	OG16C	0.5002/	0.5002	RUN Already Running...
294497.oct	gpujob24	v6a000	OG16C	0.5002/	0.5002	RUN Already Running...
294532.oct	gpujob25	v6a000	OG16C	0.5002/	0.5002	QUE -
294533.oct	gpujob26	v6a000	OG16C	0.5002/	0.5002	QUE -

7. 利用状況表示

システム全体のジョブ状態表示コマンド qstatall/sstatall

システム全体のジョブ状態を表示 (qstatall)

自身および自身が所属するグループ以外のジョブ状況表示については、一部マスクされます

RequestID	ReqName	UserName	Queue	Pri	STT	S	Memory	CPU	Elapse	R	H	M	Jobs
295005. oct			OC1C	0	RUN	-	237. 78M	28334. 78	1200	Y	Y	Y	1
294985. oct			OC32C	0	RUN	-	20. 57G	2594205. 19	3659	Y	Y	Y	32
294767. oct			OC8C	0	RUN	-	4. 45G	7534902. 42	88349	Y	Y	Y	4
295012. oct			OC8H	0	RUN	-	59. 90G	37050. 51	390	Y	Y	Y	4
294908. oct			OG1C	0	RUN	-	162. 48M	15575. 20	15629	Y	Y	Y	1
295001. oct			OG8C	0	RUN	-	11. 25G	123041. 72	1290	Y	Y	Y	4
291581. oct	gpu job10	v6b000	OG16C	0	RUN	-	1. 63G	46958024. 99	1964672	Y	Y	Y	1
293027. oct	gpu job21	v6a000	OG16C	0	RUN	-	2. 82G	12501732. 65	611672	Y	Y	Y	1
294532. oct	gpu job25	v6a000	OG16C	0	QUE	-	0. 00B	0. 00	0	Y	Y	Y	1
294533. oct	gpu job26	v6a000	OG16C	0	QUE	-	0. 00B	0. 00	0	Y	Y	Y	1

システム全体のジョブスケジューリング状態を表示 (sstatall)

自身および自身が所属するグループ以外のジョブ状況表示については、一部マスクされます

RequestID	ReqName	UserName	Queue	Pri	STT	PlannedStartTime
295005. oct			OC1C	0. 5002/	0. 5002	RUN Already Running...
294985. oct			OC32C	0. 5002/	0. 5002	RUN Already Running...
294767. oct			OC8C	0. 5002/	0. 5002	RUN Already Running...
295012. oct			OC8H	0. 5002/	0. 5002	RUN Already Running...
294908. oct			OG1C	0. 5002/	0. 5002	RUN Already Running...
295001. oct			OG8C	0. 5002/	0. 5002	RUN Already Running...
291581. oct	gpu job10	v6b000	OG16C	0. 5002/	0. 5002	RUN Already Running...
293027. oct	gpu job21	v6a000	OG16C	0. 5002/	0. 5002	RUN Already Running...
294532. oct	gpu job25	v6a000	OG16C	0. 5002/	0. 5002	ASG 2018-06-14 16:30:20
294533. oct	gpu job26	v6a000	OG16C	0. 5002/	0. 5002	QUE -

7. 利用状況表示

ジョブのスケジューリング情報表示 Web

スケジューラによるジョブの実行予定の状況をWebブラウザで表示可能です。

(大阪大学 サイバーメディアセンター 大規模計算機システムポータル内にあります)

[CPU Nodes]		OCTOPUS	CPU Scheduling Condition	(MM/DD/YY hh:mm)
* oct-c001 - 099		# : Assigned Node . : Unassigned Node X : Reserved Node		
* oct-c100 - 199		Node oct-p001 - 044		
* oct-c201 - 236		0000000001111111112222222223333333344444444 12345678901234567890123456789012345678901234		
[GPU Nodes]		(DATE)		
* oct-g001 - 037		02-06 14:00	#	#
		02-06 15:00	#	#
		02-06 16:00	#	#
		02-06 17:00	#	#
		02-06 18:00	#	#
		02-06 19:00	#	#
		02-06 20:00	#	#
		02-06 21:00	#	#
		02-06 22:00	#	#
		02-06 23:00	#	#
		02-07 00:00	#	#
		02-07 01:00	#	#
		02-07 02:00	#	#
		02-07 03:00	#	#
		02-07 04:00	#	#
		02-07 05:00	#	#
		02-07 06:00	#	#
		02-07 07:00	#	#
		02-07 08:00	#	#
		02-07 09:00	#	#
		02-07 10:00	#	#
		02-07 11:00	#	#
		02-07 12:00	#	#
		02-07 13:00	#	#
		02-07 14:00	#	#
		02-07 15:00	#	#
		02-07 16:00	#	#
		02-07 17:00	#	#
[Xeon Phi Nodes]				
* oct-p001 - 044				
[Large Memory Nodes]				
* oct-mm001 - 002				

アジェンダ

1. システム構成
2. 利用概要
3. 計算ノード群
4. 大容量ストレージ
5. ソフトウェア環境
6. OCTOPUSの利用方法
7. 利用状況表示
8. 利用の手引き

8. 利用の手引き

■ スカラ型スーパーコンピューター「OCTOPUS」

<http://www.hpc.cmc.osaka-u.ac.jp/octopus/>

■ OCTOPUSの利用方法

<http://www.hpc.cmc.osaka-u.ac.jp/system/manual/octopus-use/>

■ 大規模計算機システム ポータル

<https://portal.hpc.cmc.osaka-u.ac.jp/portal/>

(参考) ジョブクラス表

共有利用の実行用ジョブクラス

	汎用CPUノード群		GPUノード群		Xeon Phiノード群		大容量主記憶搭載ノード群
利用ノード数	実行用 ジョブクラス	実行用 ジョブクラス (HPCI利用者)	実行用 ジョブクラス	実行用 ジョブクラス (HPCI利用者)	実行用 ジョブクラス	実行用 ジョブクラス (HPCI利用者)	実行用 ジョブクラス
256-129ノード	OC256C	OC256H	-	-	-	-	-
128-65ノード	OC128C	OC128H					
64-33ノード	OC64C	OC64H					
32-17ノード	OC32C	OC32H	OG32C	OG32H	OP32C	OP32H	OCTMEM
16-2ノード	OC16C	OC16H	OG16C	OG16H	OP16C	OP16H	
1ノード	OC1C	OC1H	OG1C	OG1H	OP1C	OP1H	
投入クラス	OCTOPUS				OCTPHI		OCTMEM

Empowered by Innovation

NEC