

2021/4/26

第152回お試しアカウント付き並列プログラミング講習会
「スーパーコンピューター超入門」

より進んだ利用に向けて

お試しアカウント

- ✓ 利用者番号 tUVXYZ (t + 数字5桁)
- ✓ 利用グループ gt00
- ✓ 利用期限 5/26 (水) 9:00 まで有効

- ✓ 利用可能ノード数 8ノード以内
- ✓ ジョブあたり実行時間 15分以内

期限までは、利用規程で認められた目的範囲で自由にご使用ください

どのように使うか

- 一般利用

- 研究代表者（大学、公的研究機関の方）
- グループでの利用も可

➡ 研究実施に必要なトークン量を見積もる必要

システム	利用負担金 (年額, 税込)	利用可能 ノード数	割当資源 (トークン)	ディスク量 /work
Oakbridge-CX OBCX	100,000円 /年 (1セット)	最大256 ノード	8,640 ノード時間 (1ノード×360日相当)	4TB

電気代相当の経費を負担

一般利用の資源利用状況は `show_token, show_quota` コマンドで確認可

【企業の方は、企業利用制度により利用（詳細HP参照）】

利用予定プログラムで計算時間見積もり

OBCX の実機上でも、自分の研究室のマシンでも

- 担当講師の手元にあるプログラム

自分で書いた分子動力学コード, MPI + OpenMP に対応

やりたいこと：

16,384,000 粒子 100 ナノ秒 = 10^8 ステップ が必要

★OBCX 1ノードで実行, 10^4 ステップから確認

MPI + OpenMP ハイブリッド並列

OpenMP

= 指示文挿入による並列化

プログラムの大幅な書換えは
しなくて良いが、ノード内のみ

MPI

= 通信の明示的指定

ノードまたぎ並列には必須

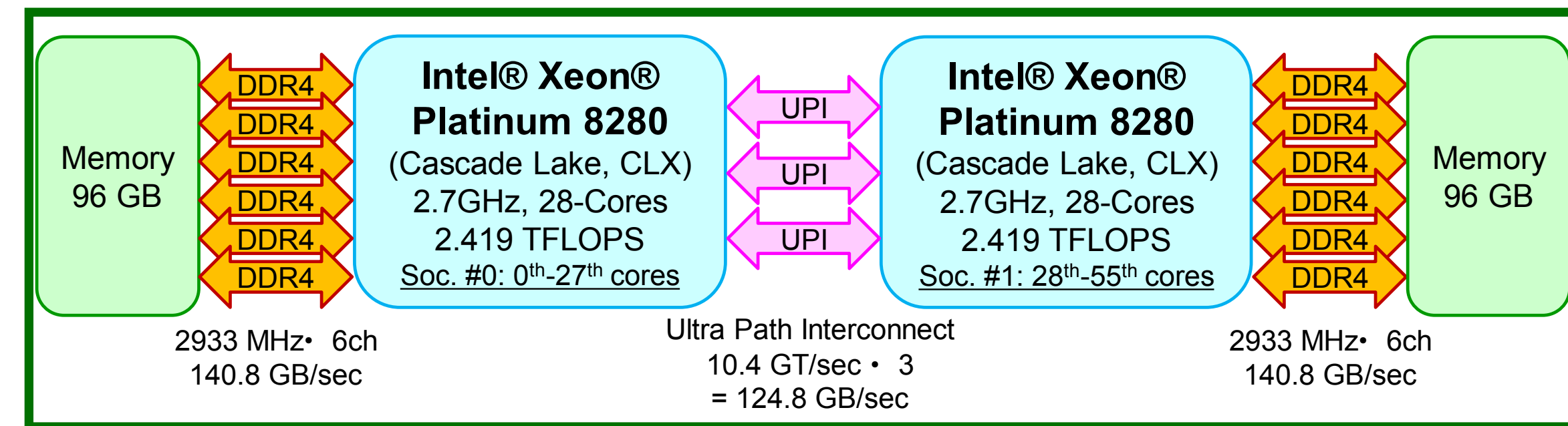
基本は1コアあたり
1スレッド, 1プロセス

OBCX 搭載のCPU

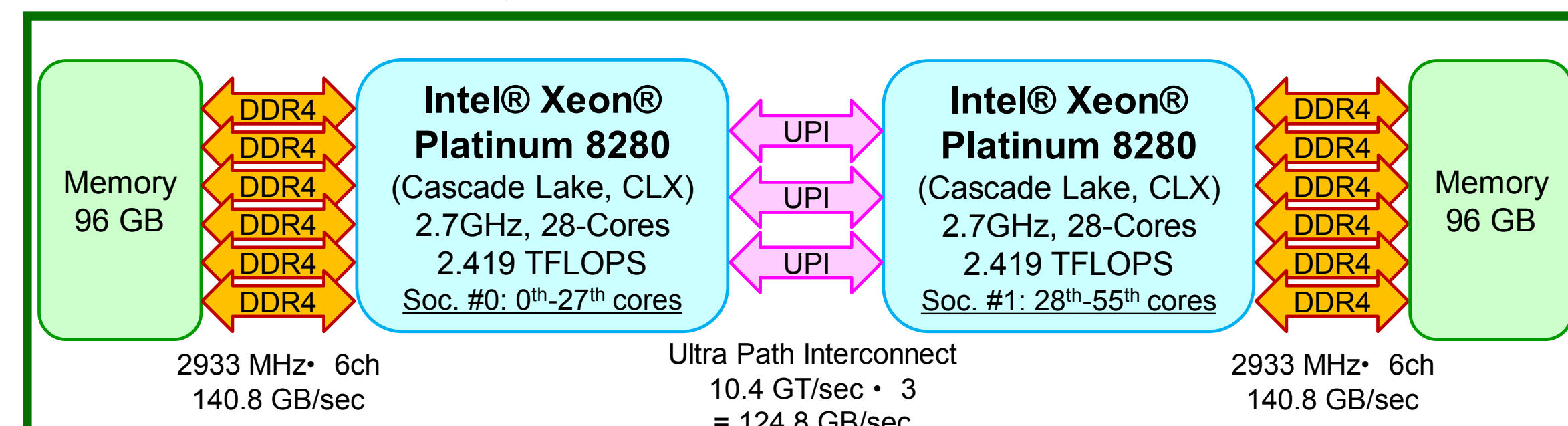
= Intel Xeon Platinum 8280 (CascadeLake)

1 CPU = 28 コア
OpenMP 7スレッド
x MPI 4プロセス

合計 2 ノード
OpenMP 7スレッド
x MPI 16プロセス



Intel Omni-Path (100 Gbps)



実行結果

1ノード, 56 コア (8 MPI x 7 OpenMP)

32,000 粒子 11.84 sec

- 単純に必要な計算資源を見積もると...

16,384,000 粒子 10⁸ ステップ

$$\begin{aligned} \Rightarrow & 11.84 \text{ sec} \times (16,384,000 / 32,000) \times (10^8 / 10^4) \\ & = 6.602 \times 10^7 \text{ sec} = 16389 \text{ hours} \end{aligned}$$

1ノード換算で トークン16400 ノード時間が必要 (約2セット)

Array Use Information:

MMD:

max of Nproc: 341003 (curr. lim=341936)
max of celenum[cn]: 45 (curr. lim=62)
max of nnlmmax[n]: 127 (curr. lim=149)
average of cell list duration: 0.018844

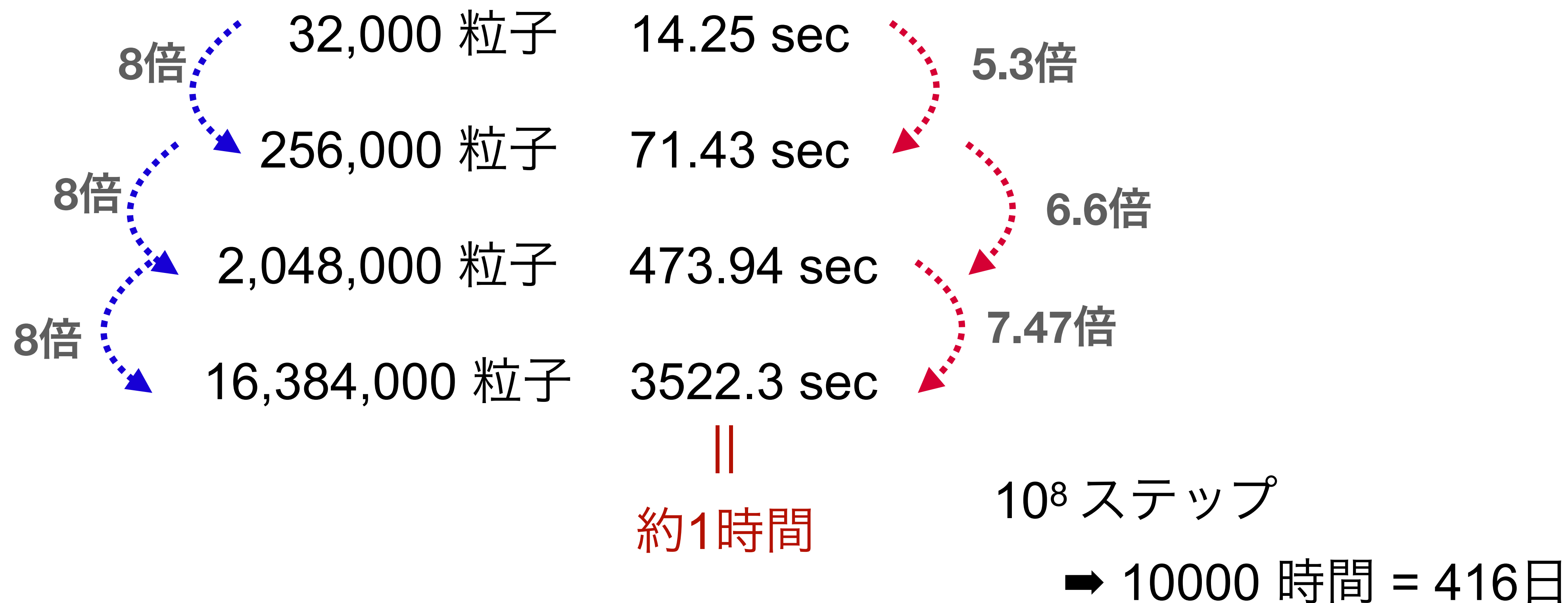
Function Load

514.293sec		
347.768sec	[67.6205%]	force()
0.00851301sec	[0.00165529%]	dist_rand()
6.53144sec	[1.26998%]	LFone()
3.96262sec	[0.770499%]	LFtwo()
87.233sec	[16.9617%]	comm_particles()
15.6167sec	[3.03654%]	[array_sendrecv]
53.7682sec	[10.4548%]	[comm_ispq]
17.8437sec	[3.46957%]	[communicateparticles]
67.4933sec	[13.1235%]	structlist()
50.0715sec	[9.73599%]	[makennlist]
0.358424sec	[0.0696925%]	[recvindexsearch]

まずは、1ノードで比較

10⁴ ステップ

全て 1ノード, 56 コア (8 MPI x 7 OpenMP)

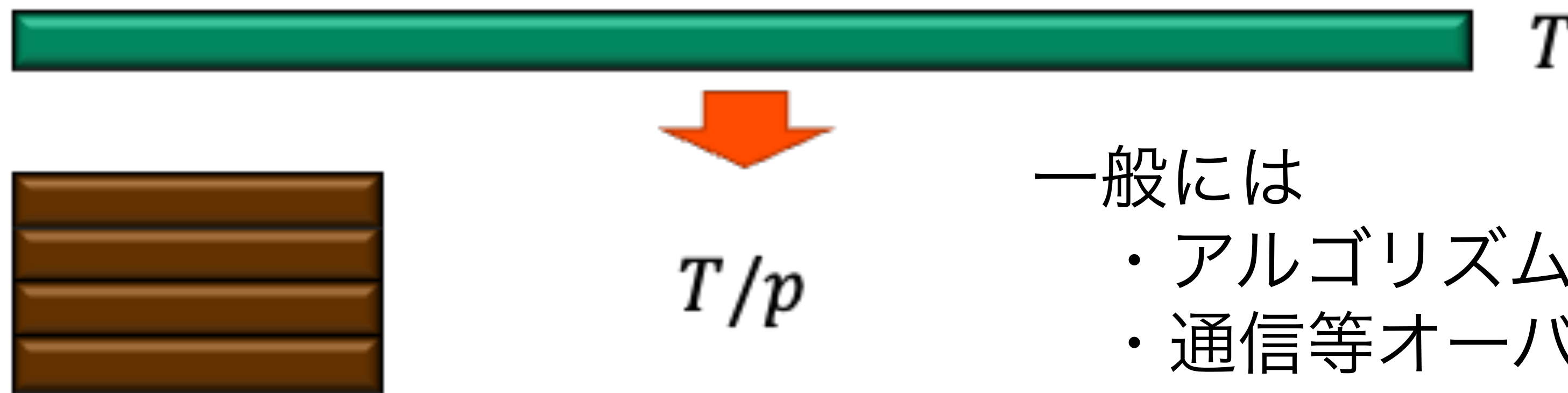


もう少し詳しい並列性能評価の話

多くのアプリケーションは並列度を増やすと性能低下

目的： 逐次実行のプログラム = 実行時間 T_s

p 台を用いて実行時間 T_p を T_s/p にしたい



一般には

- ・ アルゴリズム上の困難
- ・ 通信等オーバーヘッド

並列化効率 $E_p = 100 \times (T_p/T_s) / p$

並列化後の実行時間短縮 $S_p = T_p/T_s$

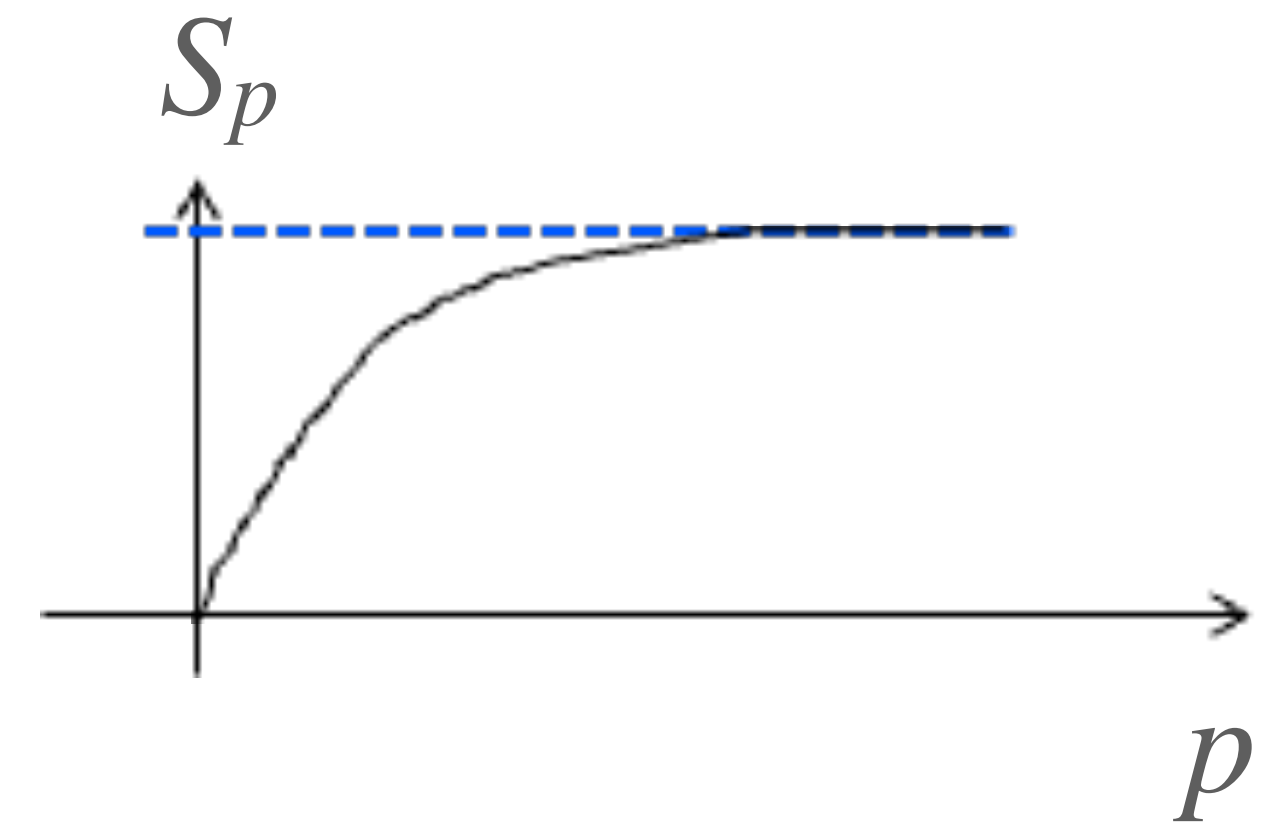
もう少し詳しい並列性能評価の話

アムダールの法則

- 逐次実行時間を T_s とし、このうち並列化できる割合を α とする
- 台数効果は以下のように計算できる：

$$S_p = \frac{T_s}{\frac{\alpha T_s}{p} + (1 - \alpha)T_s} = \frac{1}{\frac{\alpha}{p} + 1 - \alpha} = \frac{1}{1 + \alpha \left(\frac{1}{p} - 1\right)}$$

- 無限大の数のプロセッサを使っても、台数効果は $1/(1 - \alpha)$ が上限
(アムダールの法則)
 - 全体の90%が並列化できたとしても、台数効果は最大で $\frac{1}{1-0.9} = 10$ 倍
 - →高性能を達成するためには、少しでも並列化効率を上げる実装をすることが重要



並列化効率評価 = 多ノード実行性能を推定

16,384,000 粒子, 10^4 ステップに対して評価 (8 MPI x 7 OpenMP)

1 ノード	3522.26 sec	1.8倍	(8 MPI x 7 OpenMP)	
2 ノード	1929.47 sec	1.8倍	(8 MPI x 14 OpenMP)	
4 ノード	1091.22 sec	1.8倍	(8 MPI x 28 OpenMP)	
32 ノード	164.84 sec	6.6倍	(64 MPI x 28 OpenMP)	→ 20日 で 10^8 step
256 ノード	40.1173 sec	4.2倍	(512 MPI x 28 OpenMP)	→ 4日 で 10^8 step

注) お試しアカウントでは 8 ノードまでしか実行できません

台数効果が大雑把に逆算してみると。。。。

- 10^4 ステップに要する逐次実行時間 (1コアとしたときの概算)

$$T_s = 3522.26 \text{ sec} \times 56 \text{ コア} \sim 200000 \text{ sec.}$$

- 256 ノード = 14436 コア 使用した時の実行時間 40.1 sec ~ 5000 倍 = S_p

- 台数効果の式から逆算

$$S_p = \frac{T_s}{\frac{\alpha T_s}{p} + (1 - \alpha)T_s} = \frac{1}{\frac{\alpha}{p} + 1 - \alpha} = \frac{1}{1 + \alpha \left(\frac{1}{p} - 1\right)}. \quad \alpha \sim 0.99987$$

全体の計算のうちの99.987 % が並列化されていた

並列計算を開始する前に

- 99.99 % の並列化率というのには高い数字ですが、それでも100ノードを超えると、かなり計算速度は減少。
- この事情は、例えばオープンソースの並列プログラム（OpenFOAM, LAMMPS, Gromacs, etc...）を使用する場合も同じです。
- 並列化率だけでなく、並列度に応じたメモリレイアウトの変化、データの読み込みや書き出しなどに影響される場合もあります。
- 深層学習の並列計算というのもあります。

お役立ち情報 — GUI開発環境

一定以上の規模のコード開発には対話型開発環境 (IDE) が便利
— GUI 環境でコード開発から検証まで

Eclipse, Xcode (mac), Android Studio (androidアプリ),

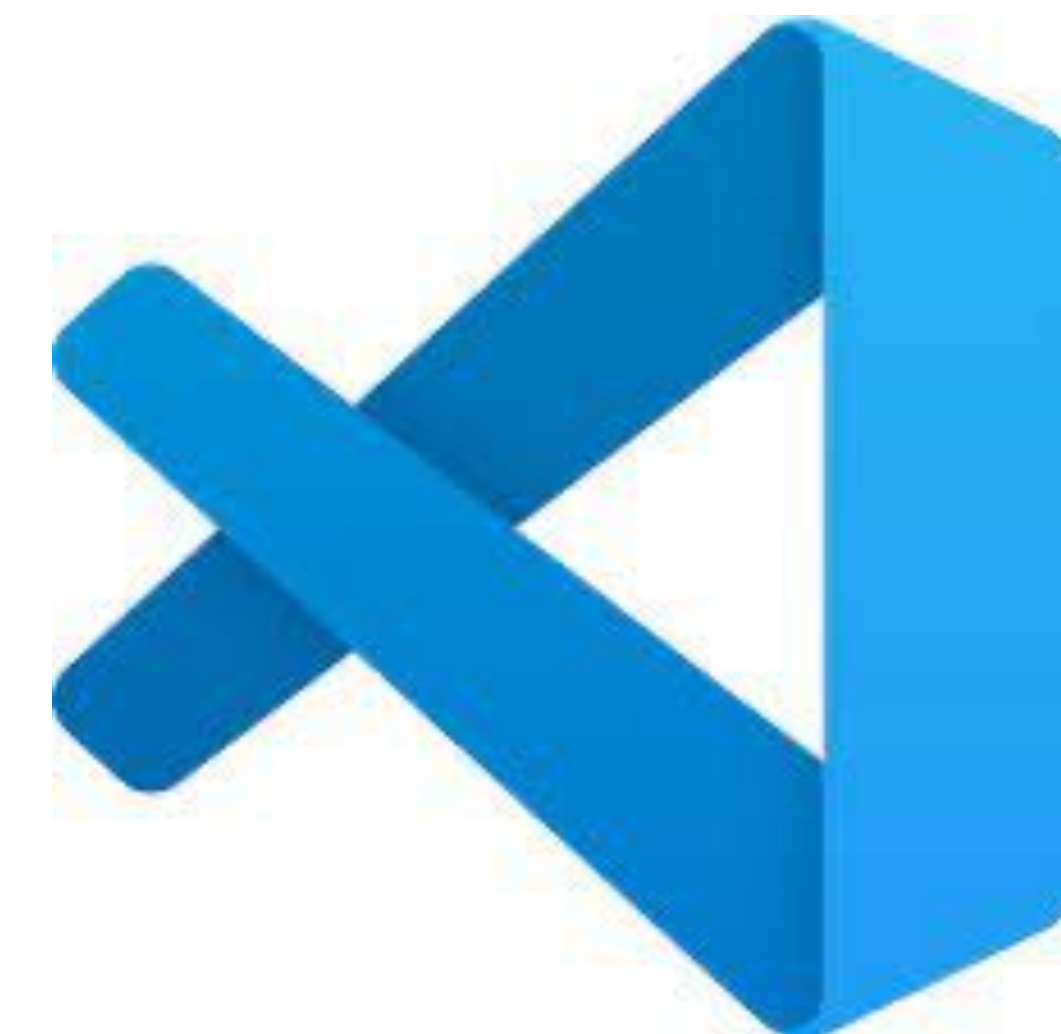
Python IDE = JupyterLab (後述)

Microsoft Visual Studio Code **フリー**

Remote-SSH 機能が2019年より追加

スパコン上での利用も便利に

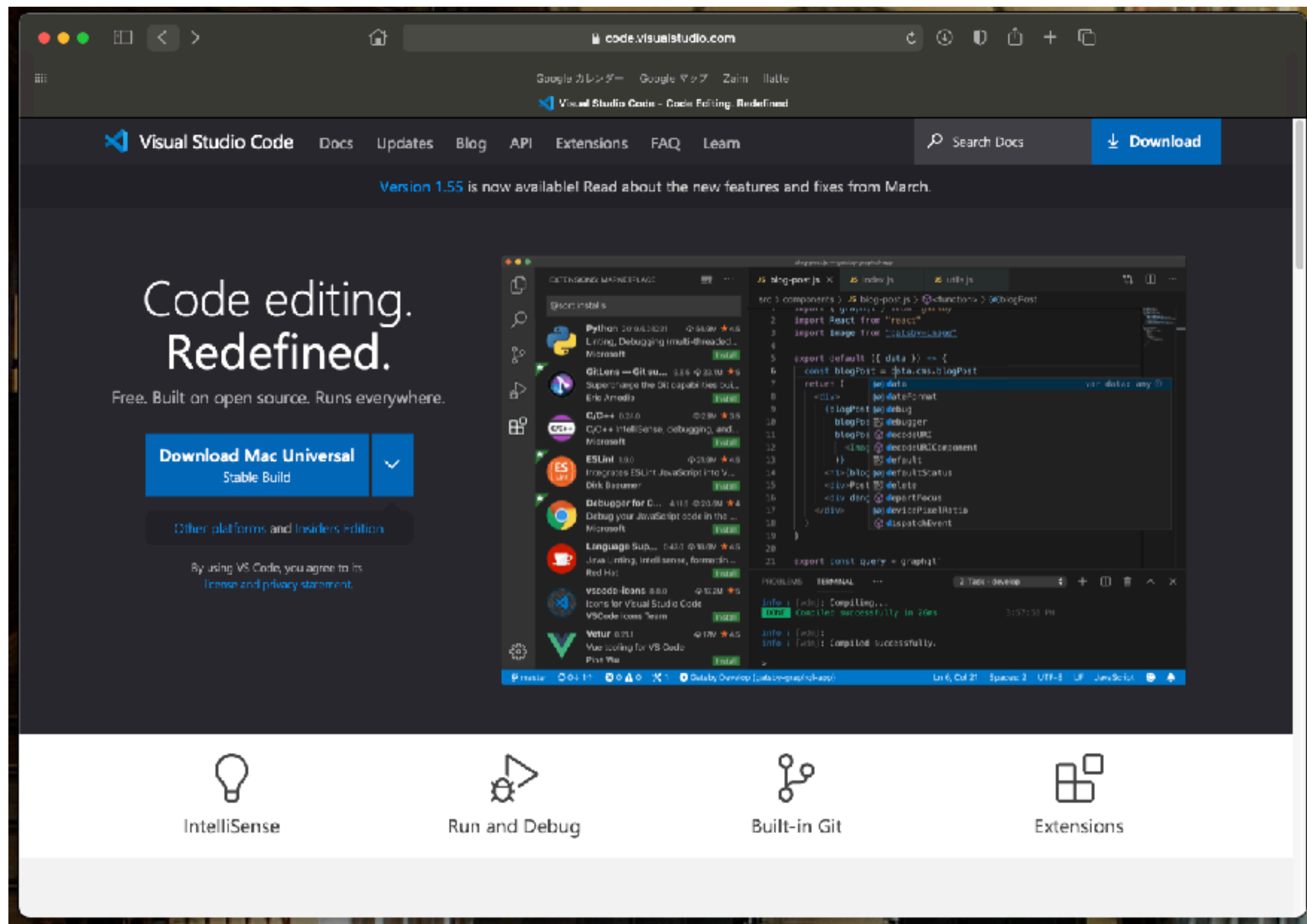
<https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.vscode-remote-extensionpack>



Visual Studio Code — ダウンロード

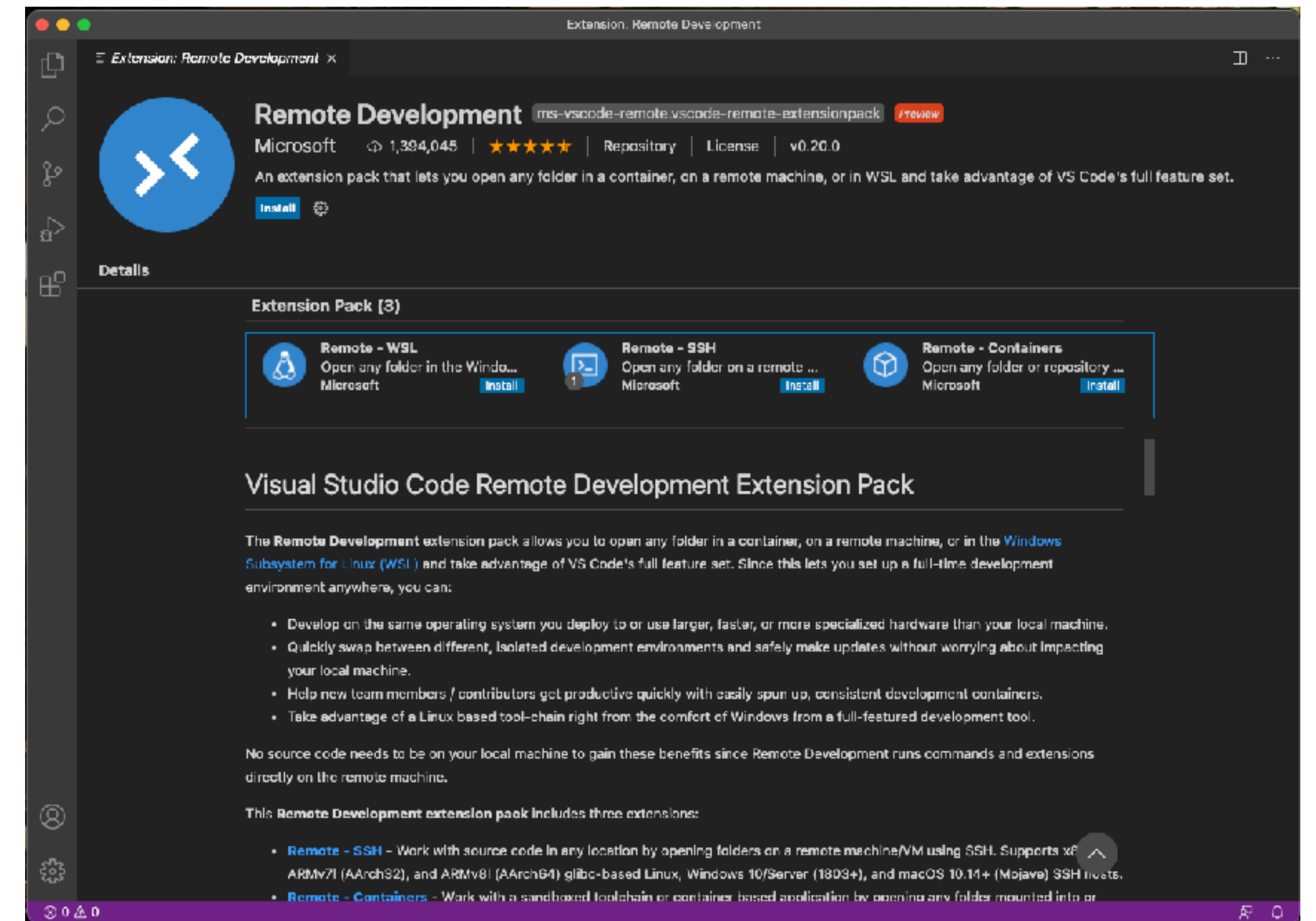
ウェブブラウザからダウンロード

<https://code.visualstudio.com>



拡張機能

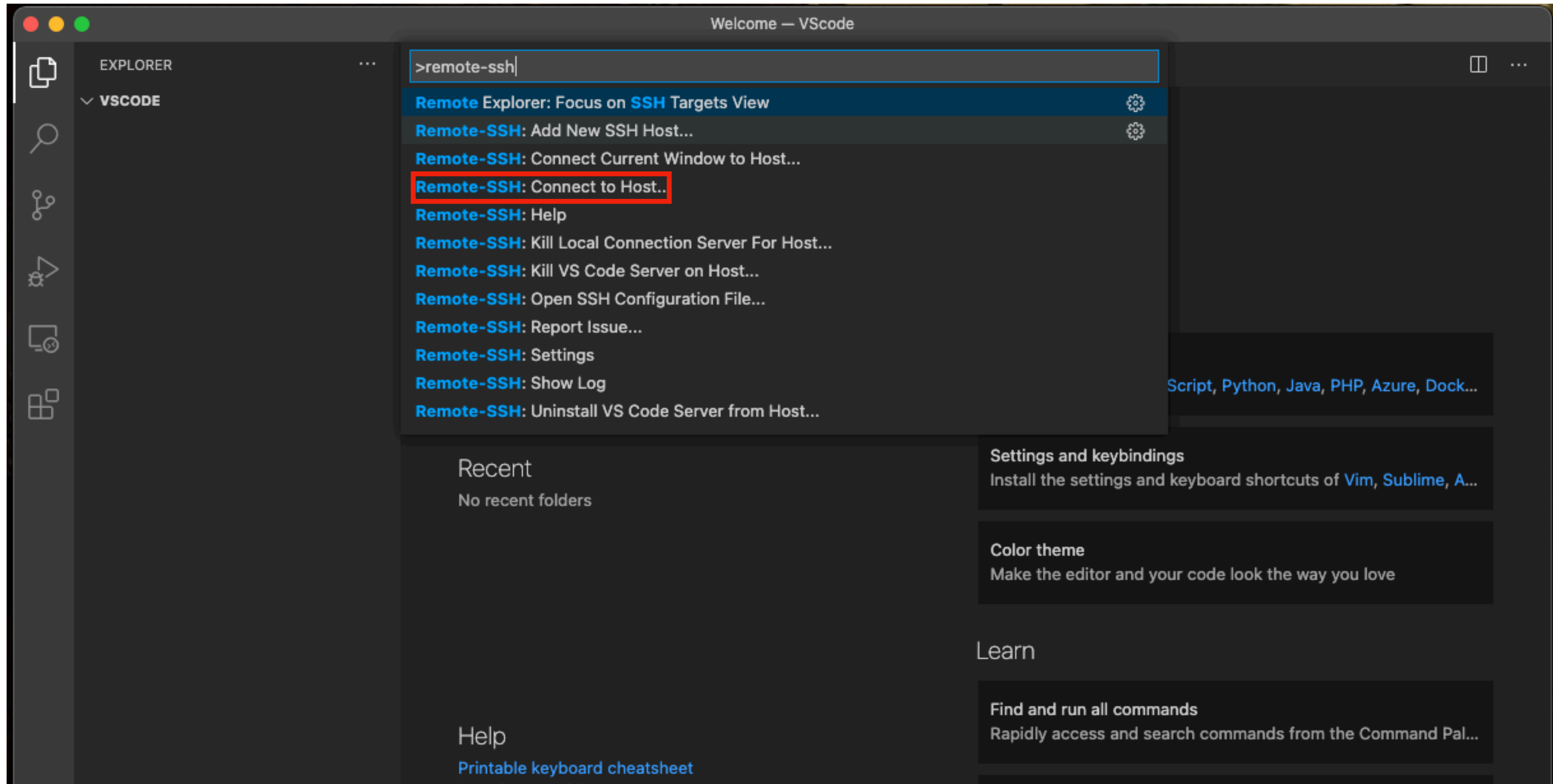
→ Remote Development → Install



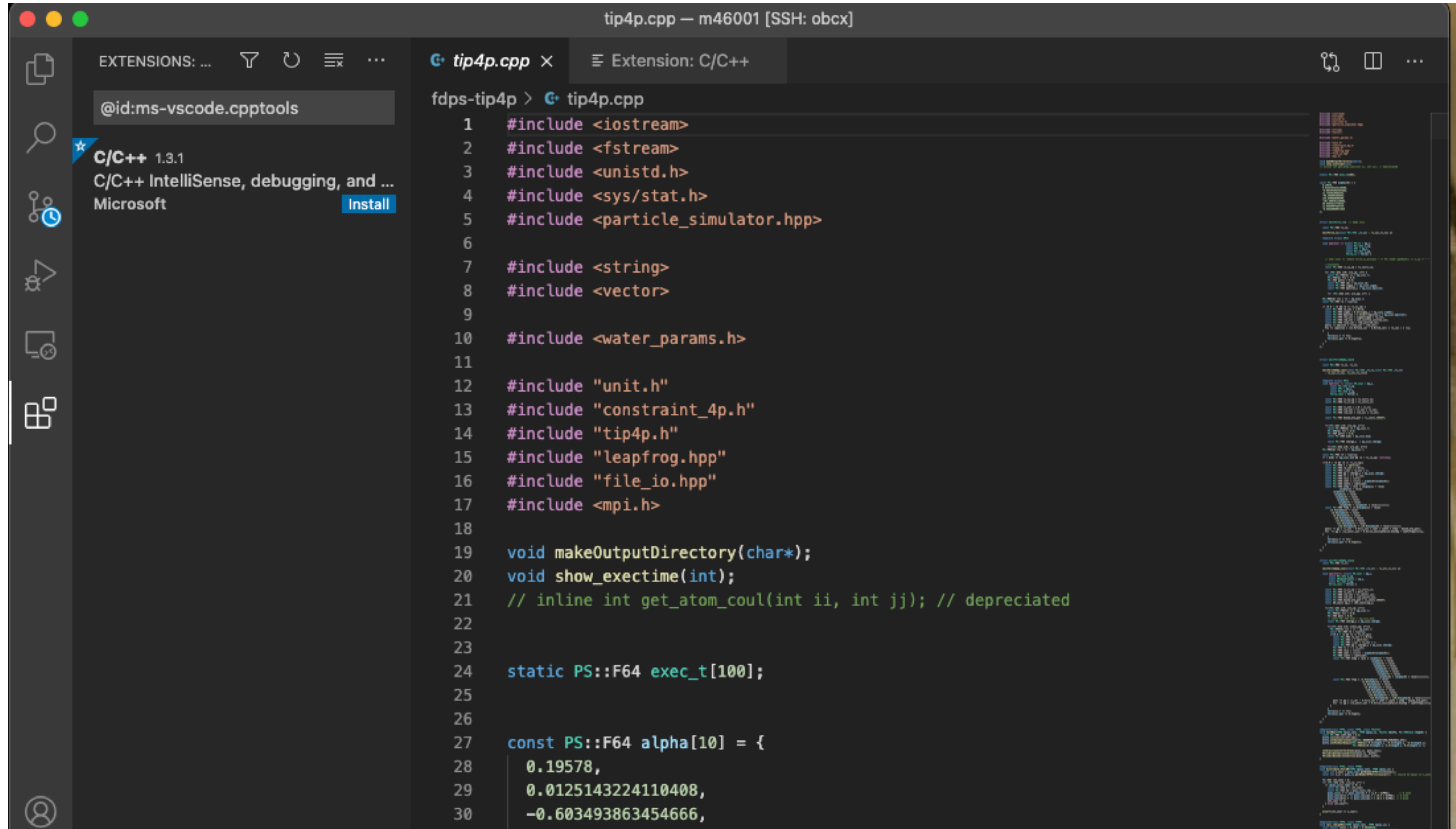
view > command palette > “remote-ssh” と入力 → 選択

接続先 tUVXYZ@obcx.cc.u-tokyo.ac.jp を入力

■ .ssh/config を使っている方は、インポートできて便利です。



コードブロックの全体も見えて、作業がしやすくなっています。



当センターのスーパーコンピューターの紹介

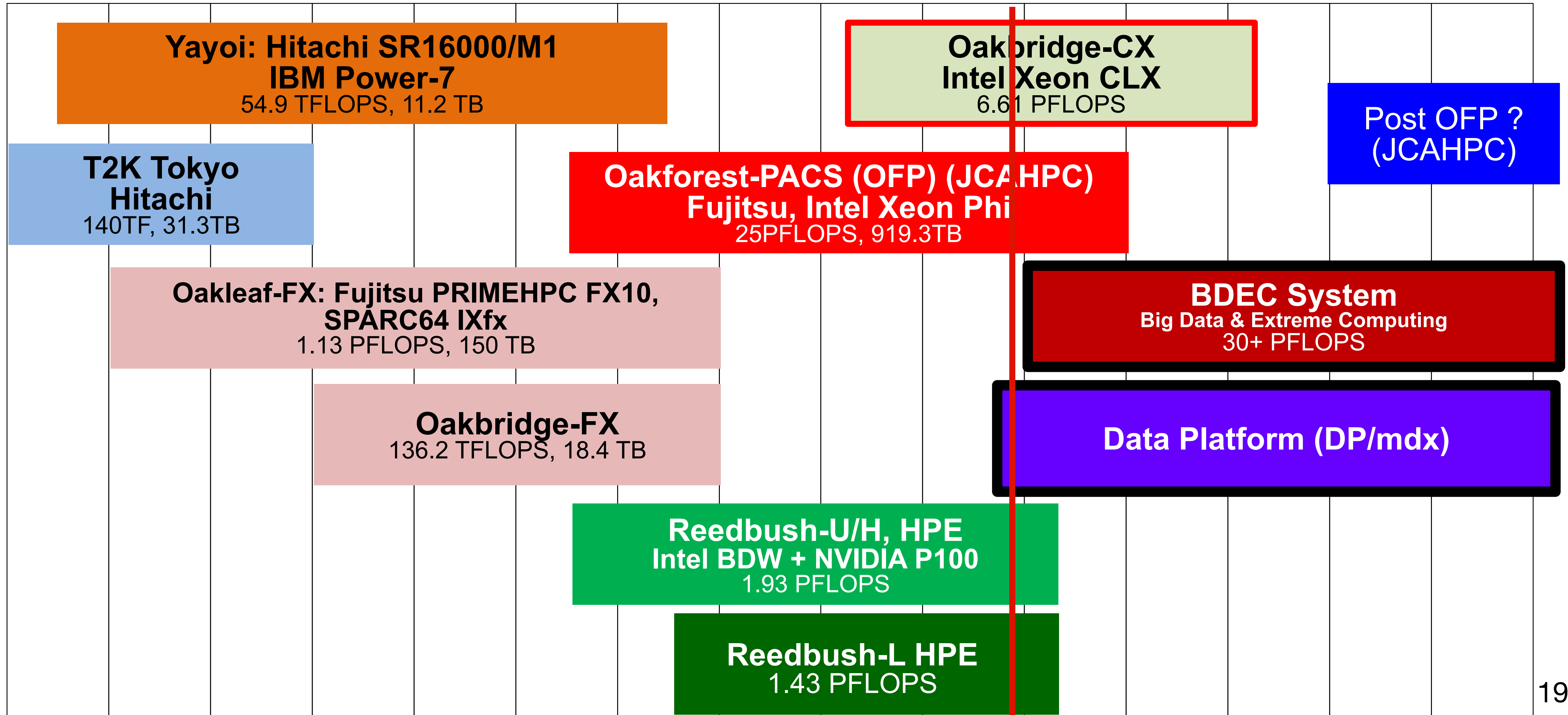


どのマシンを
使えばいいの？

Supercomputers @ ITC/U.Tokyo

Information Technology Center, The University of Tokyo

FY11 12 13 14 15 16 17 18 19 20 21 22 23 24 25



Now operating 3 Systems

2,600+ users (55+% from outside of U.Tokyo)

- **Reedbush (HPE, Intel BDW + NVIDIA P100 (Pascal))**
 - データ解析・シミュレーション融合スーパーコンピュータ
 - 3.36 PF, 2016年7月～2021年11月末 (予定)
 - 東大ITC初のGPUシステム (2017年3月より), DDN IME (Burst Buffer)
- **Oakforest-PACS (OFP) (Fujitsu, Intel Xeon Phi (KNL))**
 - JCAHPC (筑波大CCS&東大ITC)
 - 25 PF, TOP 500で6位 (2016年11月) (日本1位) (初登場時)
 - Omni-Path アーキテクチャ, DDN IME (Burst Buffer)
- **Oakbridge-CX (OBCX) (富士通, Intel Xeon Platinum 8280, CLX)**
 - 大規模超並列スーパーコンピュータシステム
 - 6.61 PF, 2019年7月～2023年6月
 - 全1,368ノードの内128ノードにSSDを搭載



Oakfoest-PACS (OFP)

東大柏キャンパスに設置

計算ノード

1ノード 68コア, **メニーコア**

Intel Xeon Phi

3TFLOPS×8,208ノード = 25 PFLOPS

メモリ (MCDRAM (高速, 16GB)

+DDR4 (低速, 96GB))

ノード間通信

フルバイセクションバンド幅を持つ
Fat-Treeネットワーク

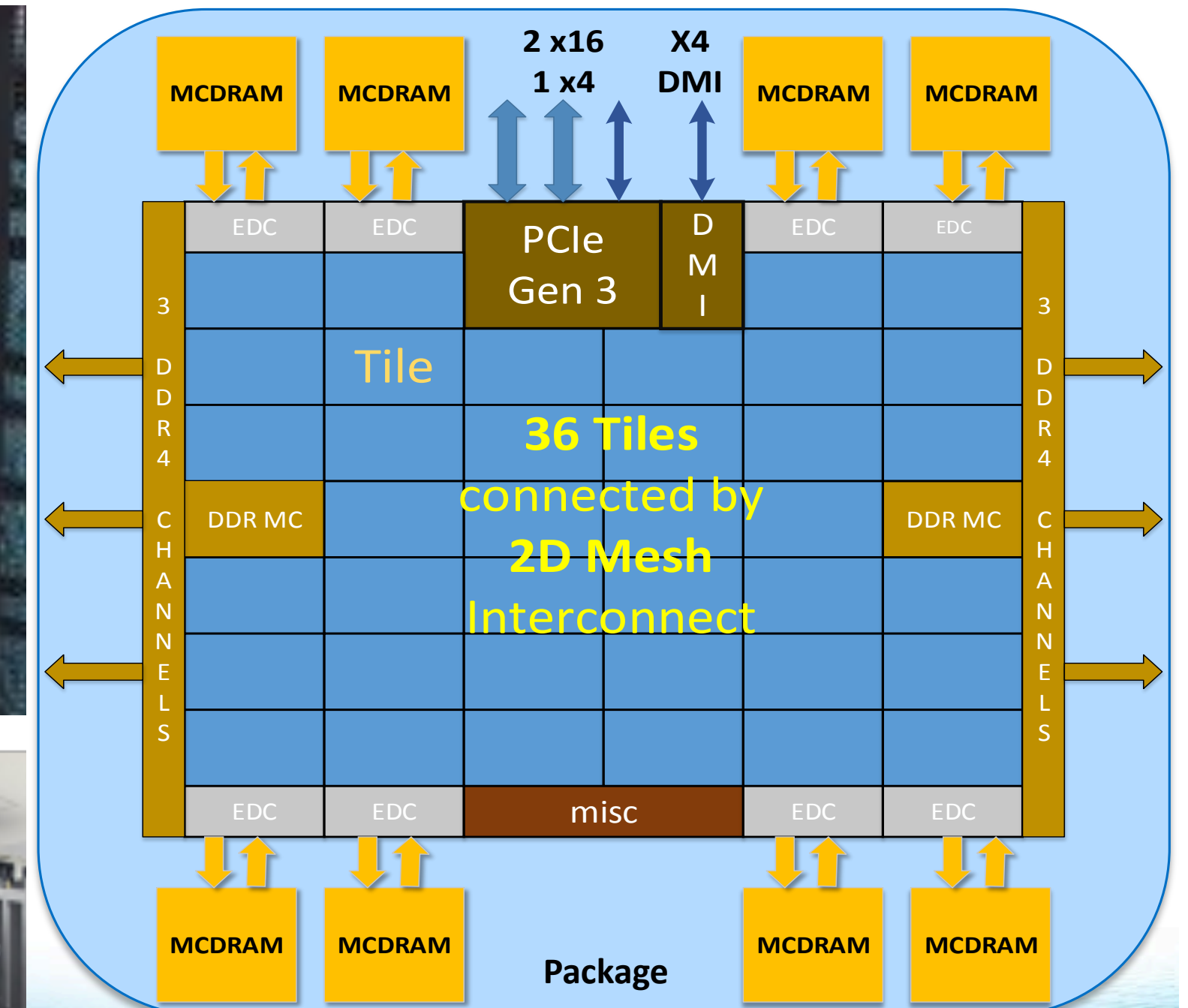
Intel Omni-Path Architecture

消費電力

Green 500でも世界6位 (2016年11月)



ラック当たり120ノード
の高密度実装



階層的プログラミングで高性能を発揮

マシン選択のガイドライン

→ プログラムが動作しますか？

受講者の皆さんは、今回のお試しアカウントでOBCXでの動作を確認可

→ 本番で必要な計算資源量（トークン量）を把握しましょう

→ メモリの必要量、データの保存量はどのくらいですか？

→ 使うプログラムはどのシステムで使えますか / 向いていますか？

GPU (Wisteria-A, Reedbush)

データ科学系、分子動力学系、その他

メニーコア (Wisteria-O, OFP)

プログラムとの相性、高速化作業が重要

大規模超並列機 (OBCX)

広い範囲のワークロードに対応します

GPUの利用

- Graphics Processing Unit
- もともと PC の3D描画専用の装置
- パソコンの部品として量産されている。
= 相対的に価格も低い



GPUの利用

- GPUはグラフィックスやゲームの画像計算のために進化を続けている。
- CPUがコア数が2-12個程度に対し、GPUは1000以上のコアがある。
- GPUを一般のアプリケーションの高速化に利用すること
「GPUコンピューティング」「GPGPU (General Purpose computation on GPU)」
などという。
- 2007年にNVIDIA社のCUDA言語がリリースされて大きく発展
- ディープラーニング（深層学習）、機械学習、AI（人工知能）**で注目



GPUの利用

■なぜGPUが使われるのか？

- ✓性能高い
- ✓NVIDIA P100 (Reedbush-H) 5,304 GFlops
- ✓Intel Xeon Phi (Oakforest-PACS) 3,046.4 GFlops
- ✓消費電力低い
- ✓スパコンに搭載されている

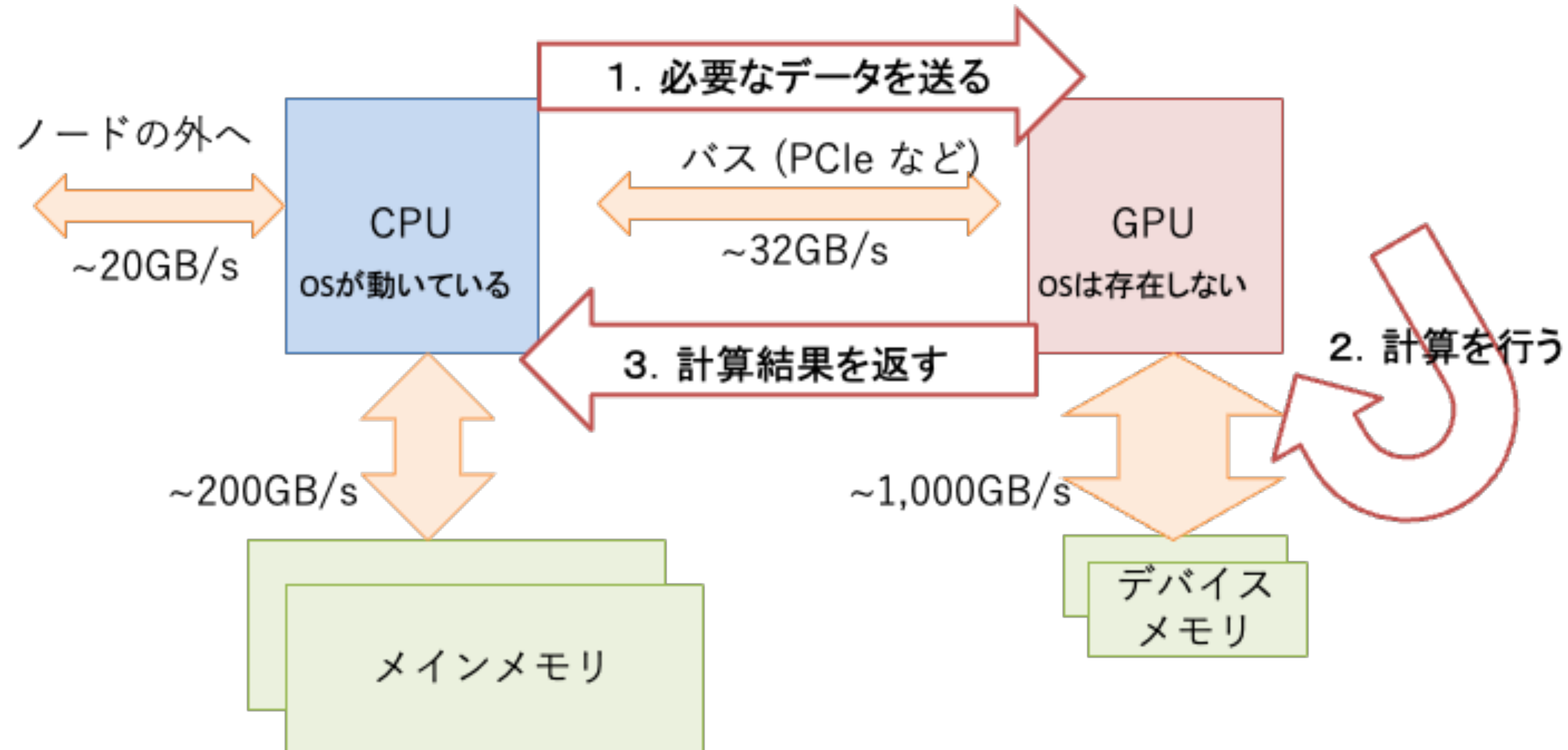
■コンピュータに取り付ける増設ボード

- ✓単体では動作できず、CPUからの指令が必要。
- ✓CPUとGPUはメモリが異なるため、メモリ間のデータ移動もプログラミングする必要がある。

■多数の小さなコア（1000以上）を搭載。これを有効に活用するため「並列計算」が必須。

- ✓CPUは大きなコア：分岐予測、パイプライン処理などできる。逐次処理が得意。
- ✓GPUは小さなコア：CPUの持つ機能がほとんどないか、全くない。

Reedbush — NVIDIA Tesla P100



- 計算はOSのあるCPUから始まる
- 物理的に独立のデバイスメモリとデータのやり取り必須

5月導入予定の新スーパーコンピューター Wisteria/BDEC-01

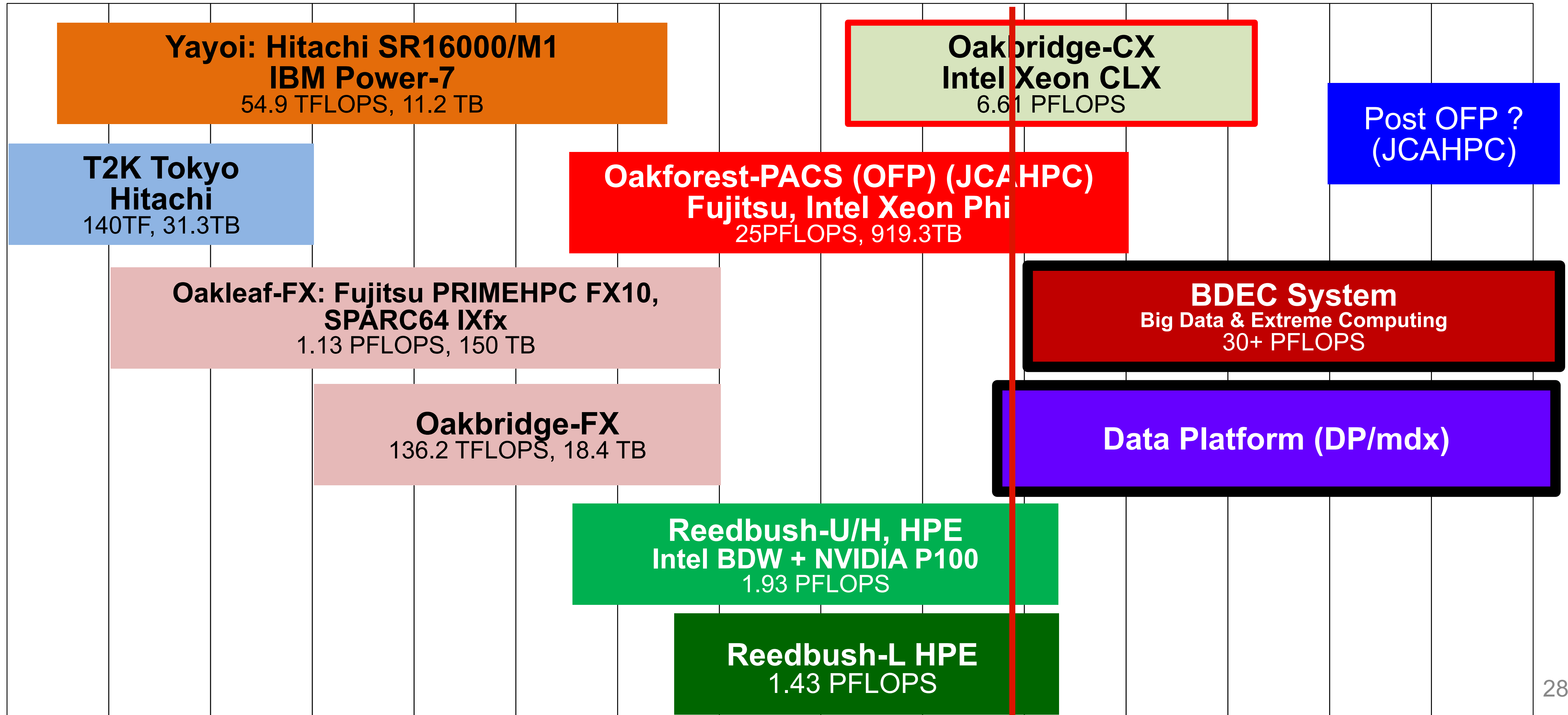
CPU + GPU 複合型

当センターの今後6年間のflagship システム

Supercomputers @ ITC/U.Tokyo

Information Technology Center, The University of Tokyo

FY11 12 13 14 15 16 17 18 19 20 21 22 23 24 25



東京大学のスーパーコンピュータは 柏キャンパス + 柏II キャンパスへ

柏地区キャンパス



**Oakforest-PACS
Oakbridge-CX**

柏IIキャンパス



**Wisteria/BDEC-01
mdx + 産総研 ABCI**

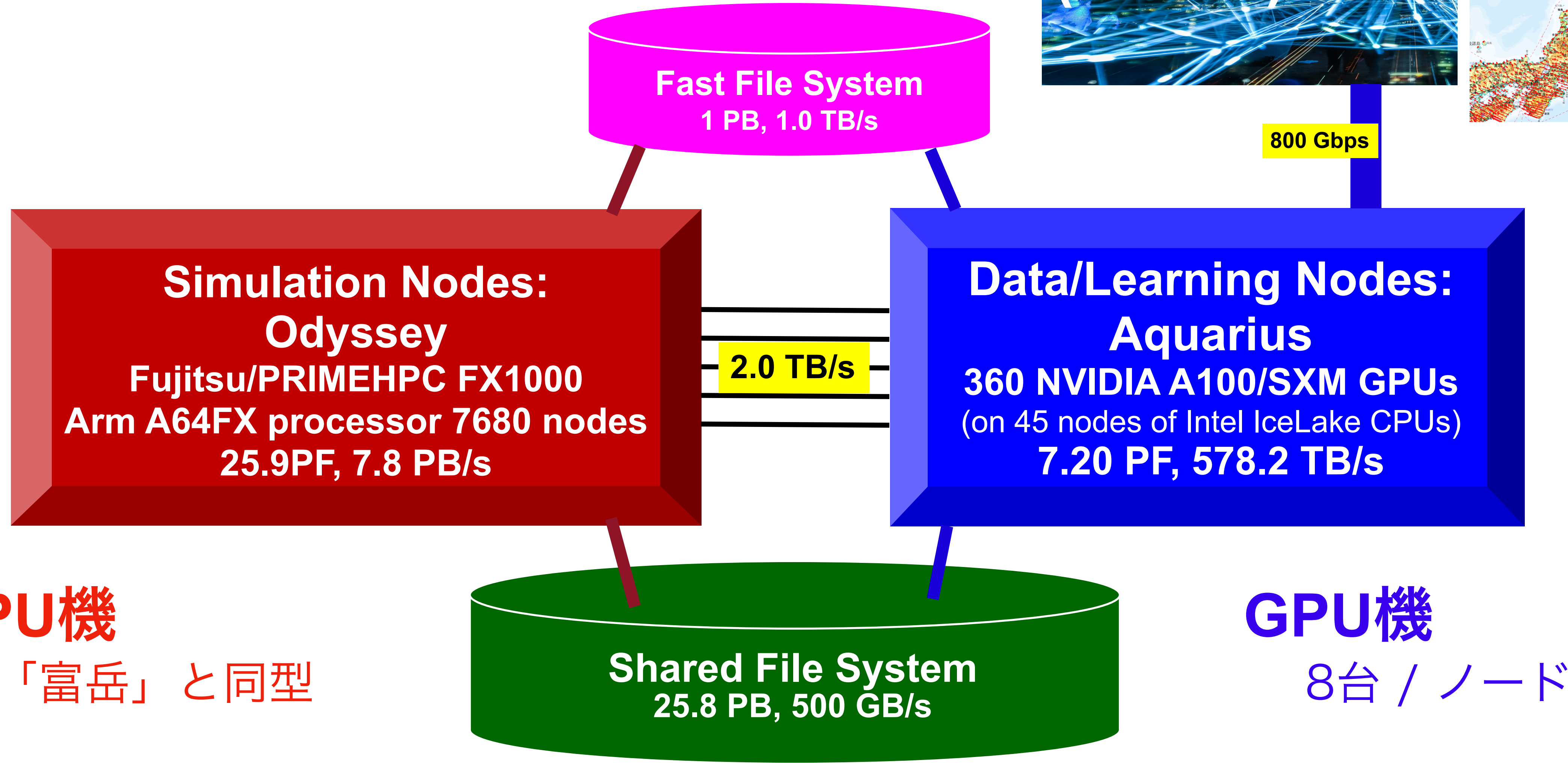
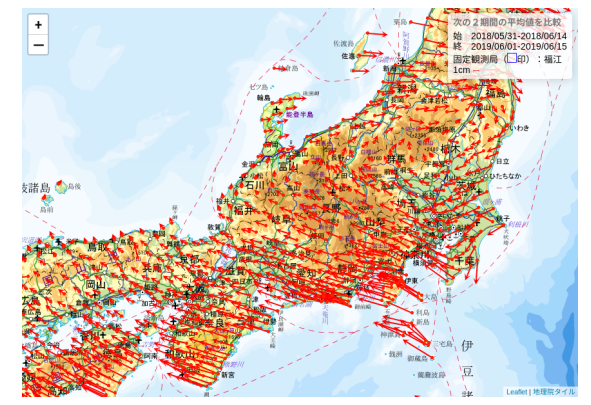
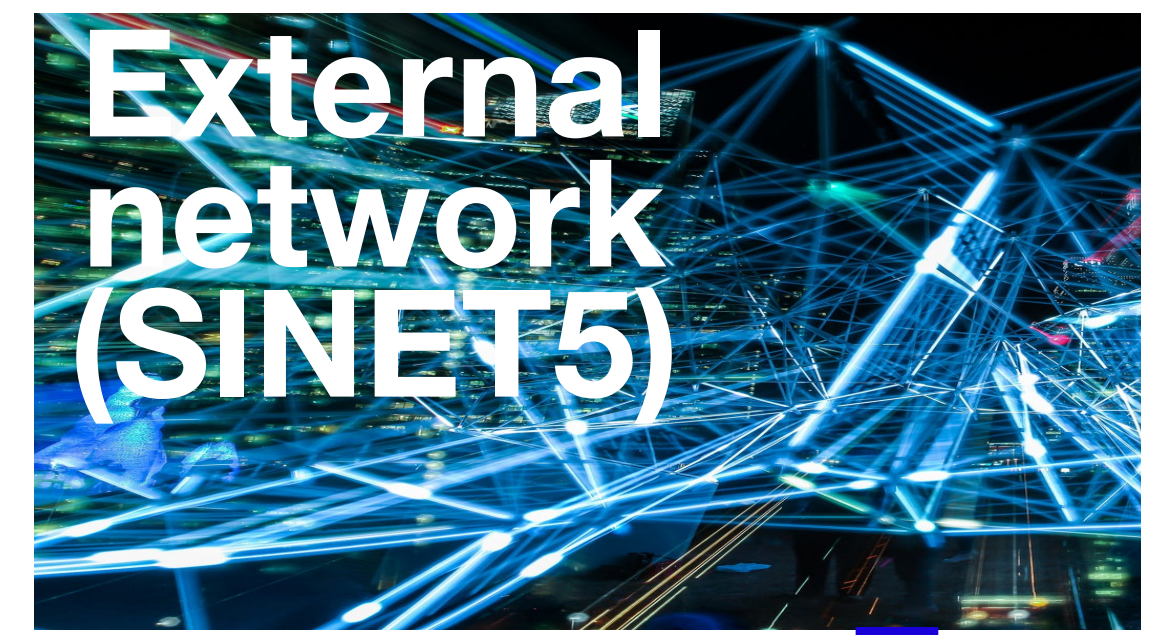


New system: **Wisteria/BDEC-01**

Wisteria = 紫藤

2021年5月14日運用開始予定

7月末までは試用期間、申込により無料で利用可能



CPU機
「富岳」と同型

GPU機
8台 / ノード



FUJITSU
PRIMEFLEX FX1000



全体構成

	Odyssey	Aquarius
理論演算性能	25.9 PFLOPS	7.2 PFLOPS
ノード数	7,680 (10x8x8x2x3x2)	45
総メモリ量	240 TiB	36.5 TiB
総メモリバンド幅	7.8 PB/s	578.2 TB/s
ネットワークトポロジ	6D Mesh Torus	Full-bisection BW Fat Tree
インタコネク	Tofu-D	InfiniBand HDR
バイセクションバンド幅	13 TB/s	4.5 TB/s
ノード間結合・ストレージ ネットワーク バンド幅	InfiniBand HDR, 2.0 TB/s	

ノード構成

項目		Odyssey	Aquarius
		Fujitsu PRIMEHPC FX1000	Fujitsu PRIMERGY GX257 (Successor model)
CPU	Processor name	Fujitsu A64FX	Next Intel Xeon IceLake processor
	#CPU (cores)	1 (48+2 or 4)	2 (36 + 36)
	Frequency	2.2 GHz	2.4GHz
	Peak performance	3.3 TFLOPS	5.53 TFlops
Memory		32 GB, 1TB/s	512 GB / 409.6 GB/s
GPU	GPU name	-	NVIDIA A100 Tensorcore
	Peak performance		19.5 TFlops (FP64 Tensor Core)
	Memory		40 GB, 1.55 TB/s
	#GPU / node		8, NVswitch connected
インターコネクト		Tofu-D	InfiniBand HDR (200 Gbps) x4
External connection		-	25 Gbps/node

Wisteria/BDEC-01 キーポイント

Odyssey

富士通 A64fx プロセッサを搭載（富岳と同型） 48 core / CPU

高いメモリバンド幅（ノードあたり 1024 GB/秒）

→ 数値流体力学・構造計算・電子状態計算などに特に威力

Aquerius

ノードあたり 8 台の NVIDIA A100 Tensorcore GPU を搭載、合計45ノード

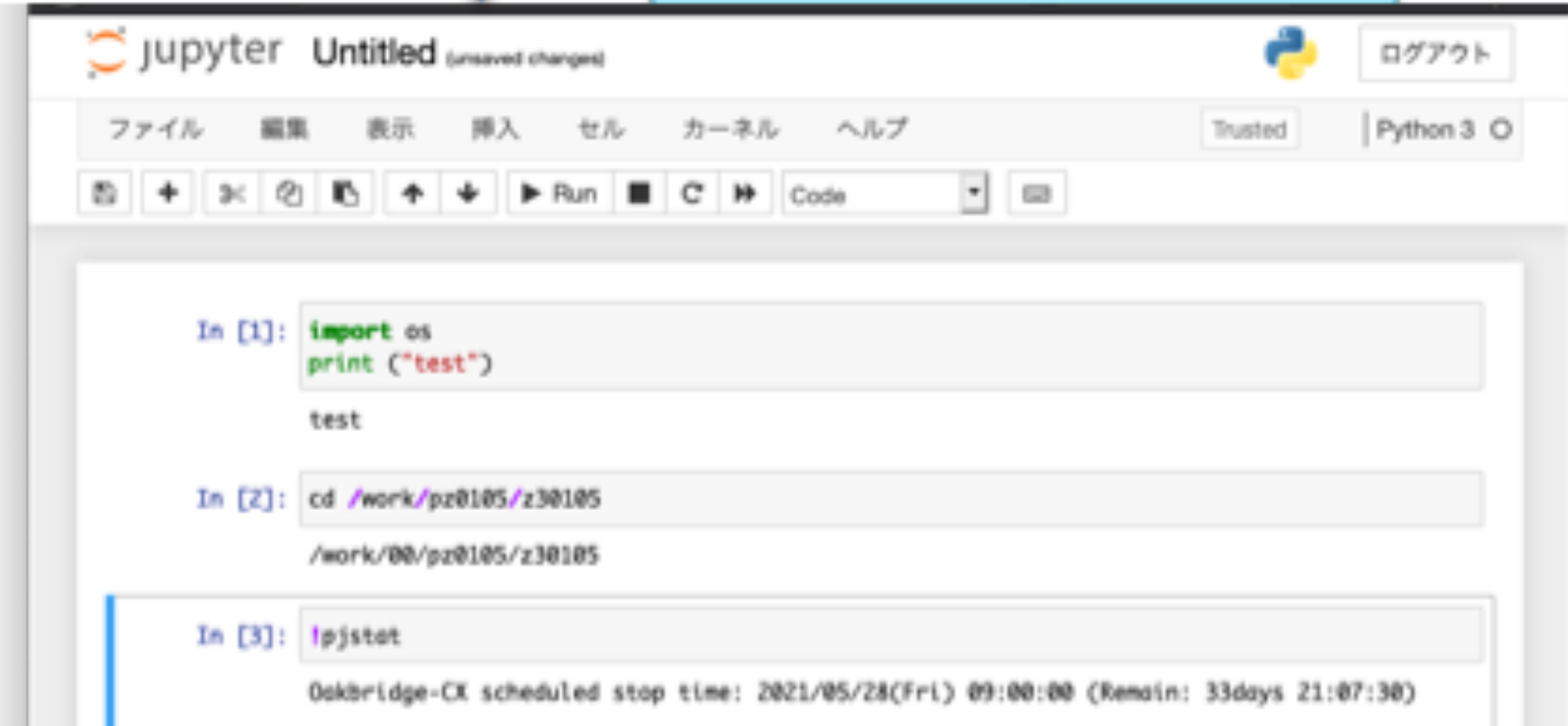
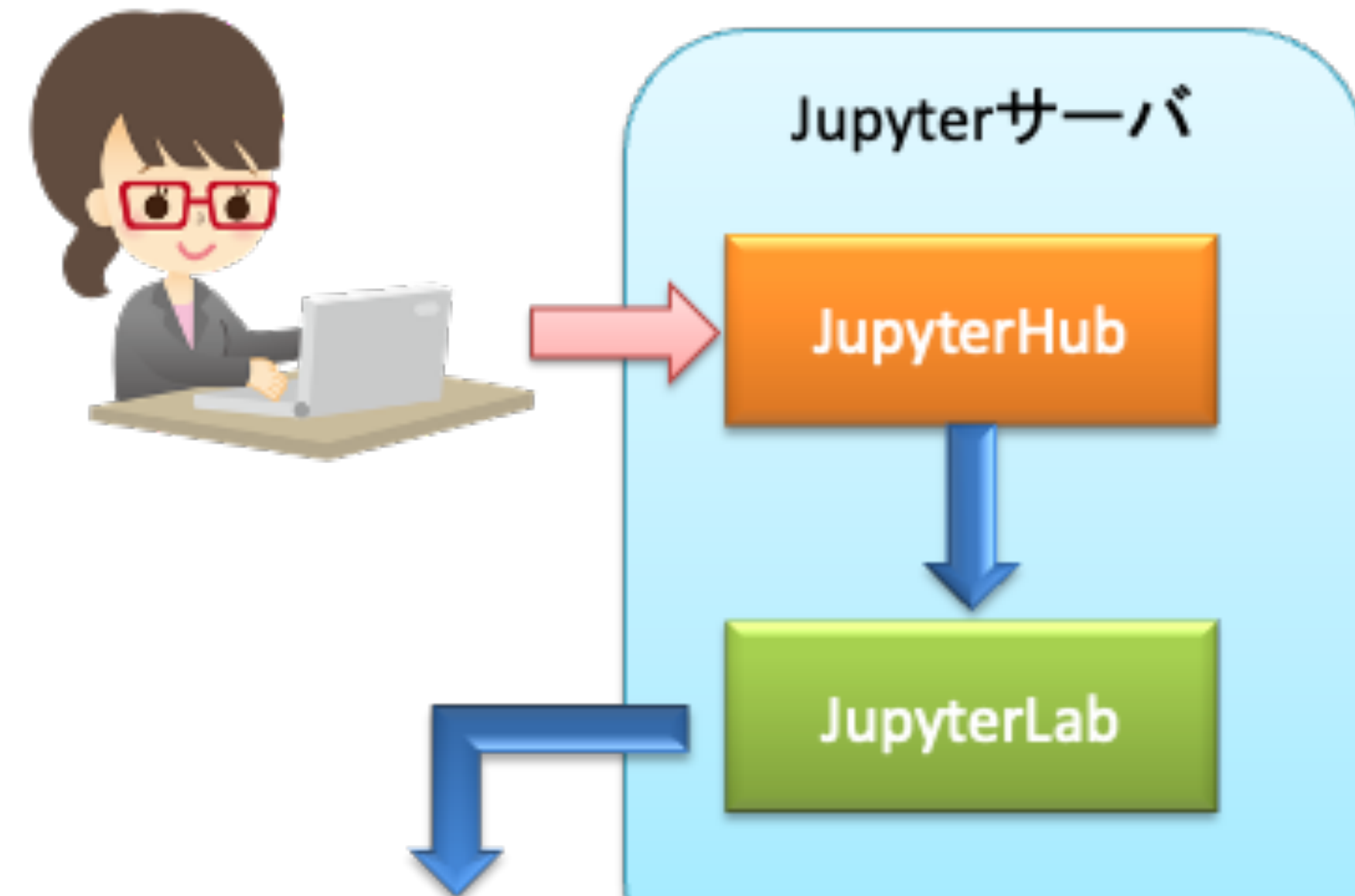
GPU 1台ずつで利用可能

JupyterLab の利用環境も提供します

JupyterLabの利用

- WisteriaでJupyterHubを用意
 - 自力でのPython環境構築不要
 - ユーザごとに用意したPython環境も利用できる
- 計算ノードとの連携
 - JupyterLabからバッチジョブを簡便に投入できる(予定)

! psub 行番号



GFLOPS (ピーク性能) 当たり利用負担 (円) : 電気代 GFLOPS/W (Green 500)

***1: Fugaku, *2 NVIDIA DGX SuperPOD, TOP/Green 500@Nov. 2020**

System	JPY/GFLOPS Small is Good	GFLOPS/W Large is Good
Oakleaf-FX/Oakbridge-FX (Fujitsu) (Fujitsu SPARC64 IXfx)	125	0.866
Reedbush-U (HPE) (Intel Xeon Broadwell (BDW))	61.9	2.310
Reedbush-H (HPE) (Intel BDW+NVIDIA P100x2/node)	15.9	8.575
Reedbush-L (HPE) (Intel BDW+NVIDIA P100x4/node)	13.4	10.167
Oakforest-PACS (Fujitsu) (Intel Xeon Phi/KNL)	16.5	4.986
Oakbridge-CX (Fujitsu) (Intel Xeon Cascade Lake)	20.7	5.076
Wisteria-Odyssey (Fujitsu/Arm A64FX)	17.8	14.66^{*1}
Wisteria-Aquarius (Intel Xeon Ice Lake + NVIDIA A100x8)	9.00	26.20^{*2}

一般利用料金

- 1セット = 60,000円 /年 (1年間 Odyssey 1 node 使用する量に相当)

	最大利用可能ノード数	トークン量	ディスク量
Wisteria-Odyssey	2,304 nodes = 110,592 cores (予定)	8640 (=24x360) / (node hours)	2TB / set
Wisteria-Aquarius	8 nodes = 64 GPUs (予定)	2880 (=24x120) / (GPU hours)	2TB / set

他、ノード固定 or GPU占有利用、外部接続ノード利用など
詳細の料金情報はホームページ 参照

<https://www.cc.u-tokyo.ac.jp>

情報基盤センターの資源を無料で使いたい場合

- 若手・女性利用課題

40歳以下または女性、年2回の募集（次回は8月末〆切）

- 学際大規模情報基盤共同利用・共同研究拠点課題

基盤センターとの共同研究， 2021年度は募集終了

- HPCI 課題（RIST）

2021年度分の一般課題について、各基盤センター分は募集終了