

ARMでHPCクラスタ

🔊ちからこぶ🔊（サーバ技術部第2課）

塩澤 広基（発表者）、柴崎 勝憲、
藤高 利文、田中 重臣、五味 学、大谷 和也、
今田 庸介、青山 政孝、立古 玲加、中村 匠、
山田 悠真、大野 薫

1. HPCとは ～検証のきっかけ～
2. 検証環境 ～構築したクラスタシステム～
3. クラウドバーステイングについて ～AWS & Slurm～
4. まとめ

1. HPCとは ～検証のきっかけ～

HPC（ハイ・パフォーマンス・コンピューティング）



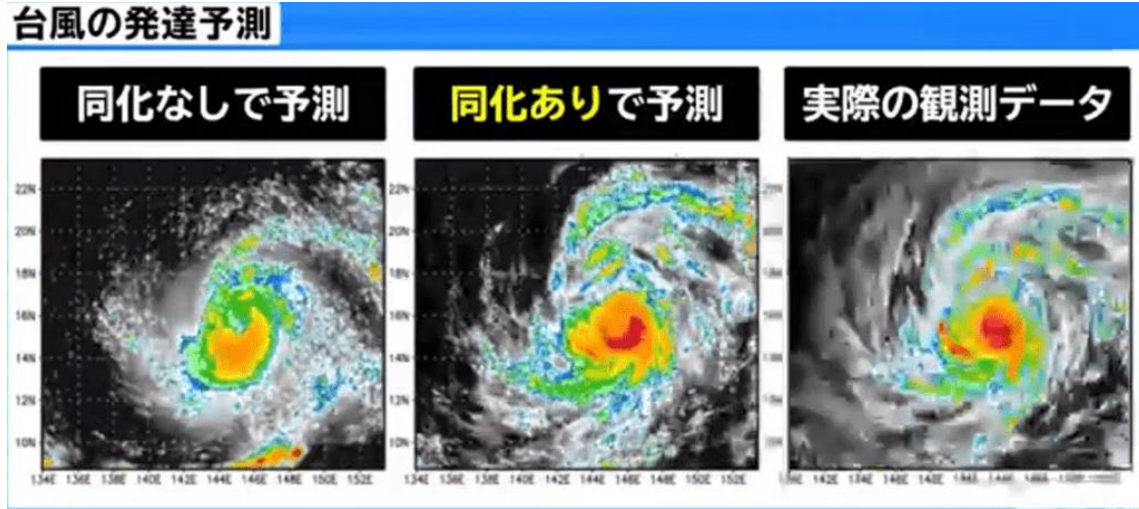
たくさんのコンピュータによる並列処理で大規模な計算を行う
システム（大規模なものでは1000台以上）

HPCはどんなことに使われているか

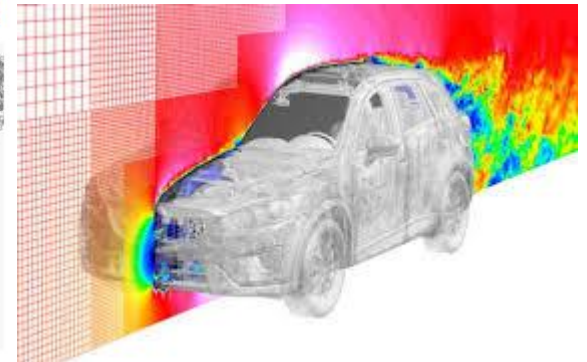
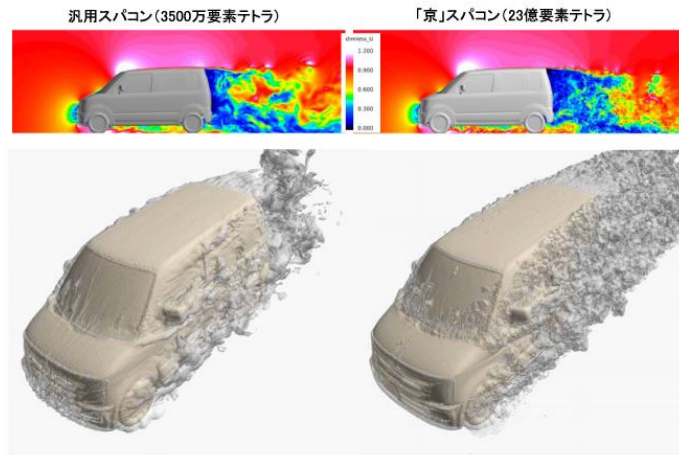
(創薬)



(気象予測)



(車の開発)

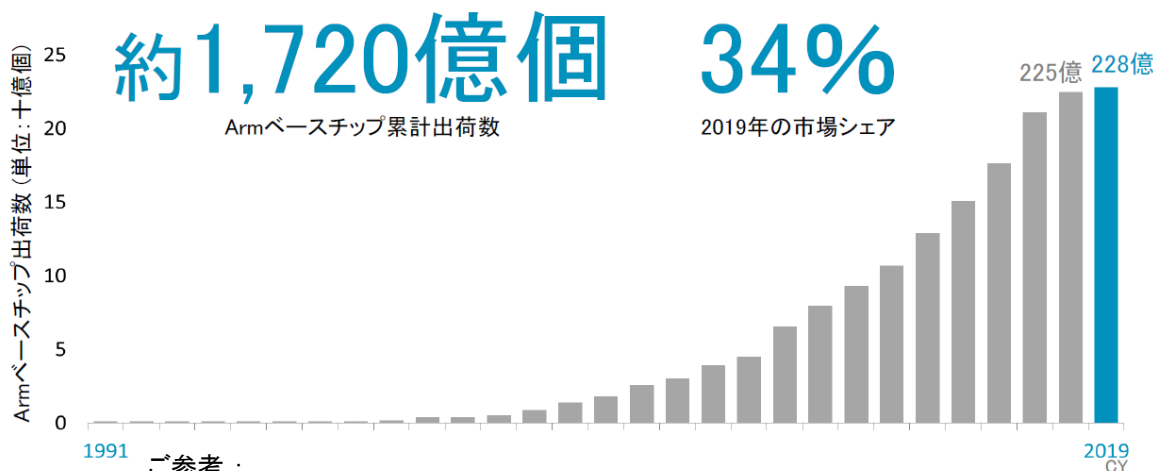


ARMアーキテクチャが脚光を浴びている

- ✓ **ARMベース**の理研と富士通が共同開発したスーパーコンピュータ「**富岳**_(※1)」が、再び「TOP500」リストの**第1位**に！
- ✓ HPC分野の急速な需要の進化に対する、**ARMテクノロジー**の独自の取り組みが浮き彫りに _(※2)



Armベースチップの出荷実績

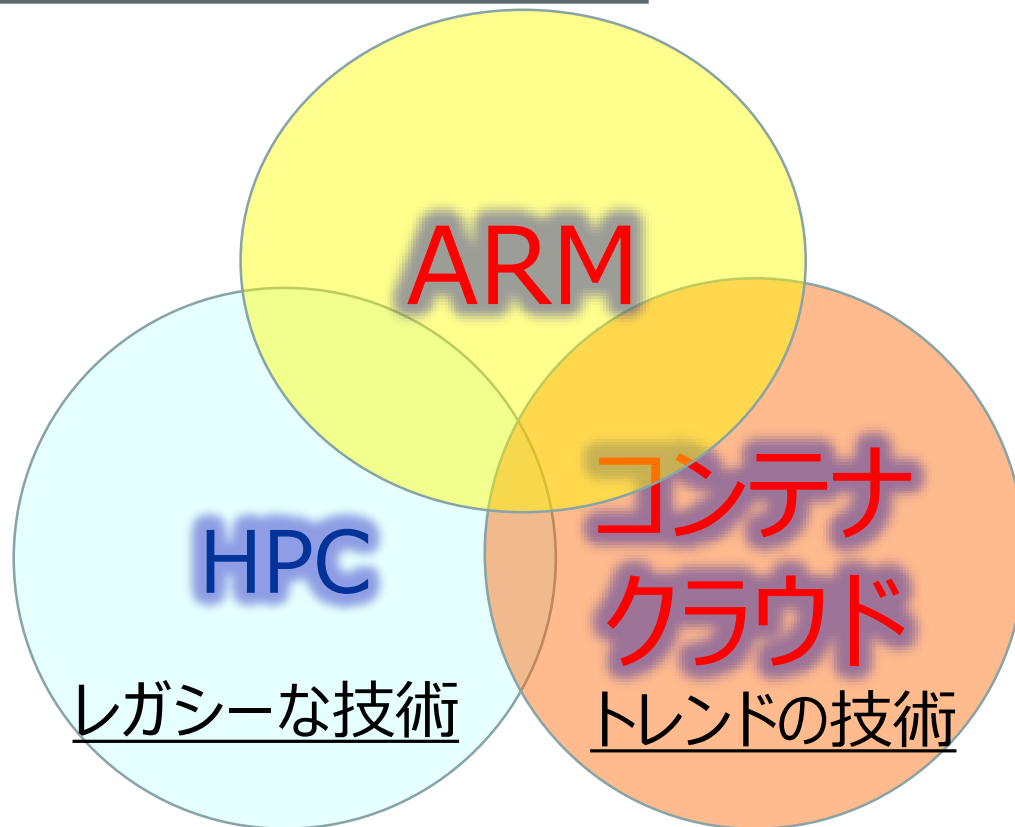


ご参考：

※1 https://www.r-ccs.riken.jp/exhibit_contents/SC20/fugaku

※2 <https://prtimes.jp/main/html/rd/p/000000116.000022759.html>

サーバ技術部第二課が得意とするHPCインフラ構築技術を
ARMを交えたトレンドのリソースにどう生かせるかを検証した



2. 検証環境 ～構築したクラスタシステム～

ARMデバイスとしてRaspberry Piを使用

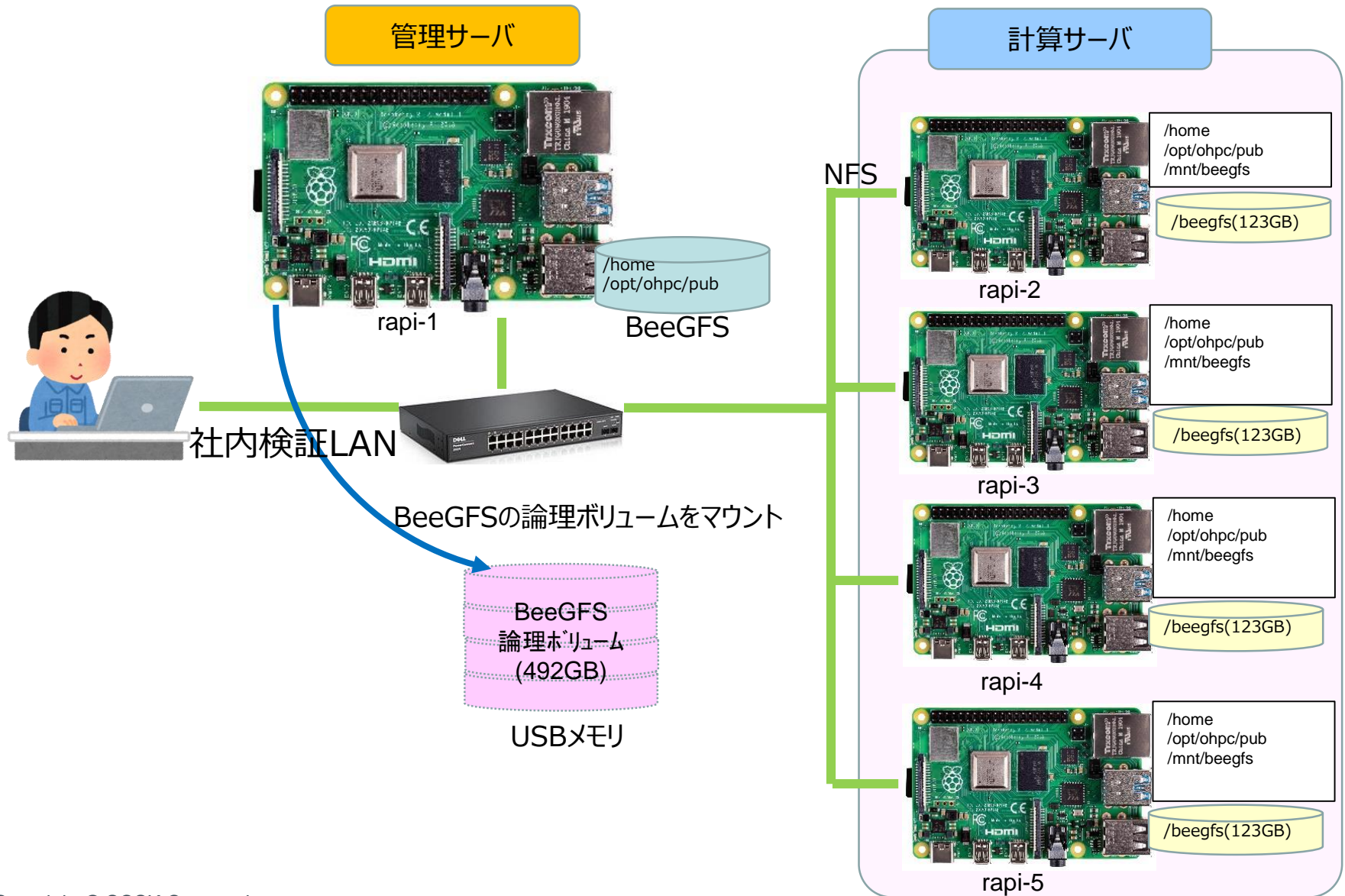
➤ 安価

→冷却用ファン、電源、OS用SDカード、ケース込みで、1台 1万円程度

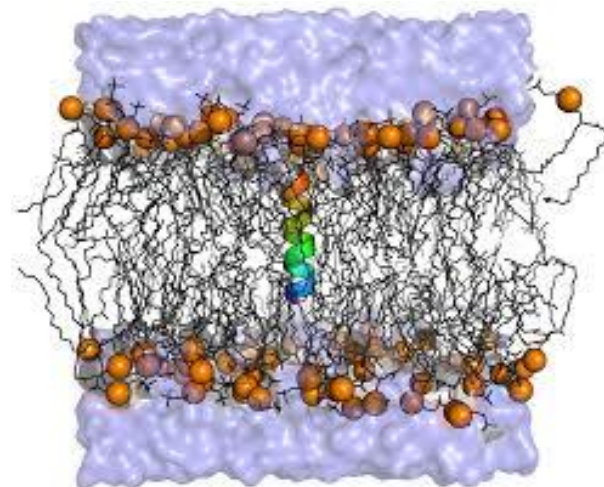
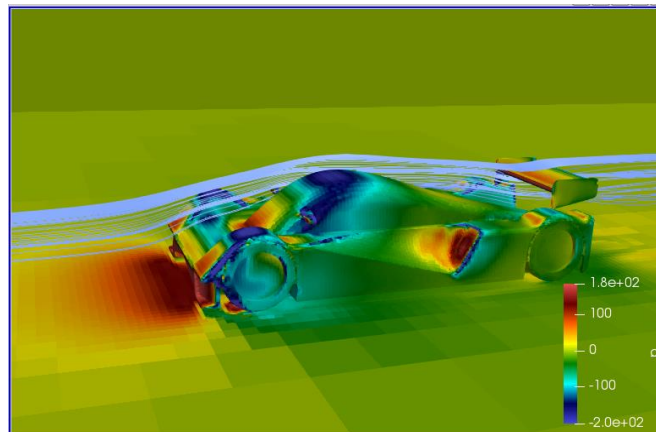
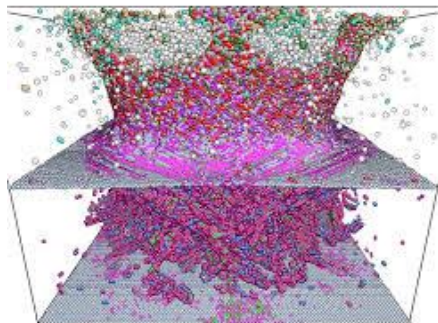
➤ HPC環境としての汎用OS(Linux)の利用が可能



| | |
|-------------------|---|
| モデル | Raspberry PI 4 Model B |
| CPU | クアッドコア 1.5GHz ARMv8 (64bit) |
| Memory | 4GB |
| Disk | 32GB Micro SDカード(OS領域) 128GB USBメモリ (BeeGFSストレージ) |
| OS | OpenSUSE Leap 15.2 |
| アプリケーション /サービス | BeeGFS 7.2(並列ファイルシステム) , NFS, NIS, chronyd,コンテナ(Singularity), slurm (ジョブスケジューラ) |
| 開発環境 | OpenHPC (Gnu v9,OpenMPI v4.0.4) |

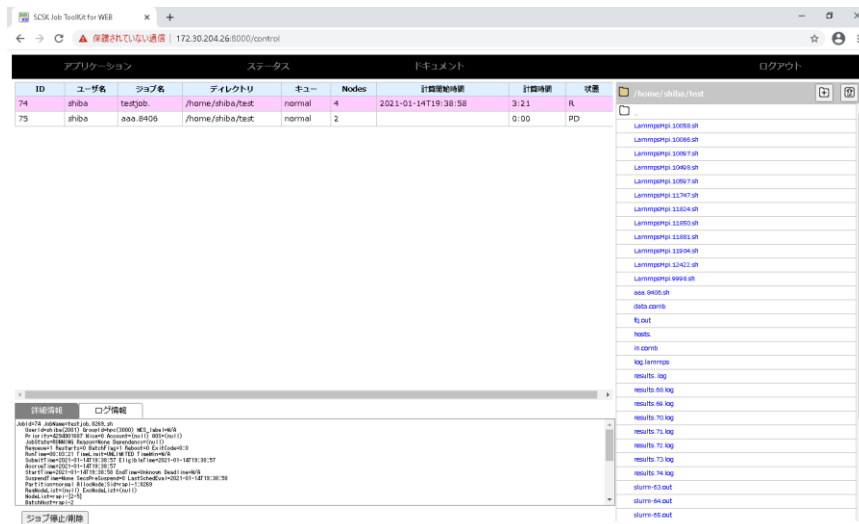


- HPCクラスタ向けに最適化されたOpenHPC開発環境をインストール
 - OpenHPC 2.0 for OpenSUSE Leap 15.2 (aarch64)
 - GNU Compiler 9.3.0
 - OpenMPI 4.0.4
- OpenHPC開発環境を利用して Lammmps と OpenFOAMをビルド
 - OpenFOAM v2006
 - Lammmps 2020(Nov19)
- OpenSUSE Leap 15.2 サイトから Gromacs ダウンロード&インストール
 - Gromacs 2019.6

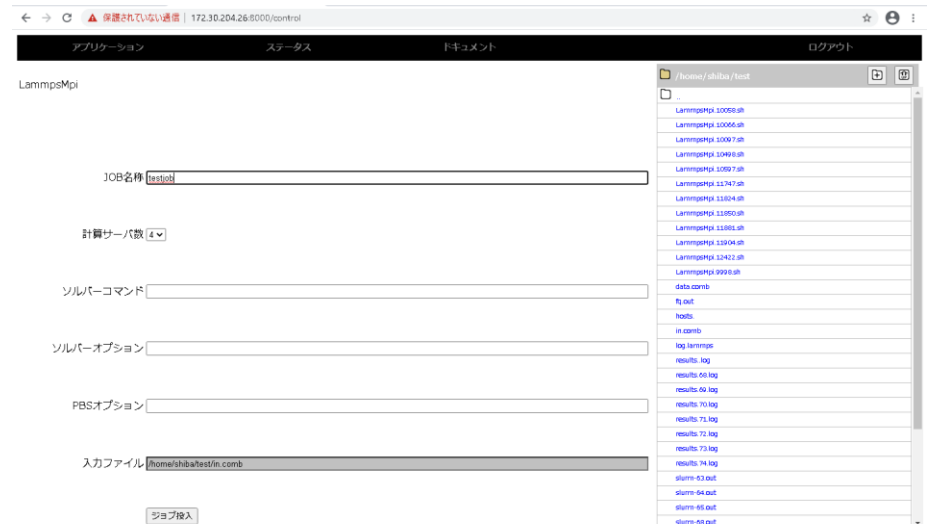


Webベースでのジョブ投入システム

- ジョブ投入システムの実装
 - WEBブラウザからジョブ投入する仕組みをRaspberry Pi上に作成
 - クライアントPCからブラウザで <https://rapi-1:8000> へアクセス



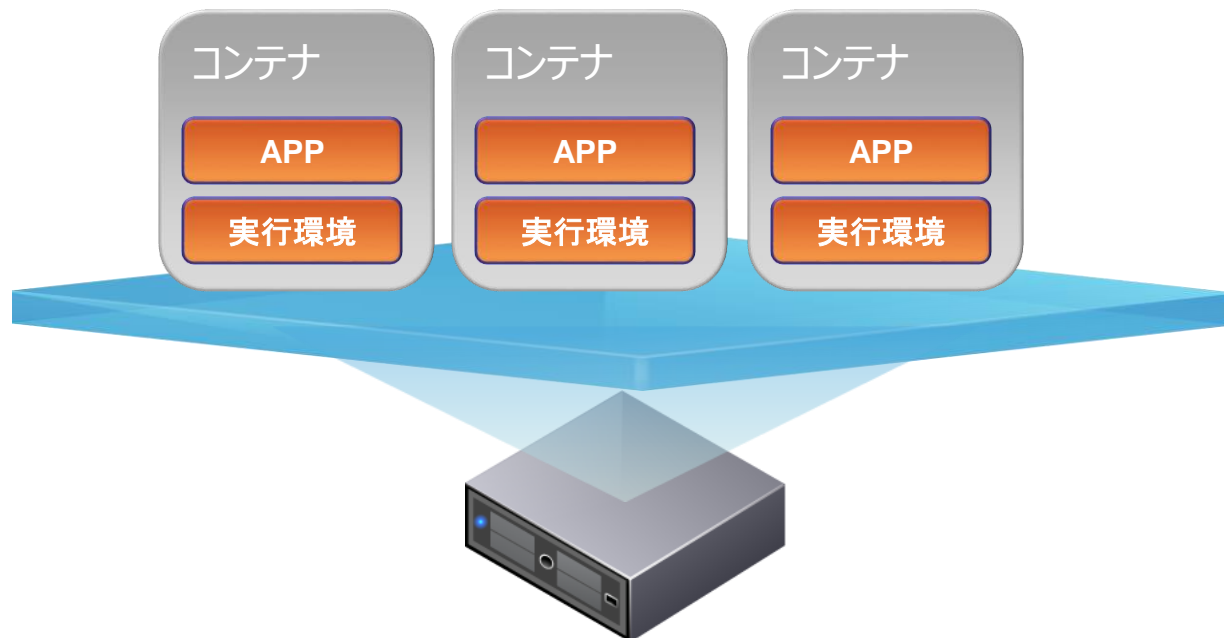
copyright (c) SCSK All right reserved.



copyright (c) SCSK All right reserved.

Singularityとは

- ✓ HPC 向けのコンテナ型仮想化プラットフォーム
- ✓ Dockerの課題(rootでの実行、MPI,GPUの利用)の改善
- ✓ HPC分野で広く利用されている



■ OpenFOAM の並列処理を実行

ジョブ投入

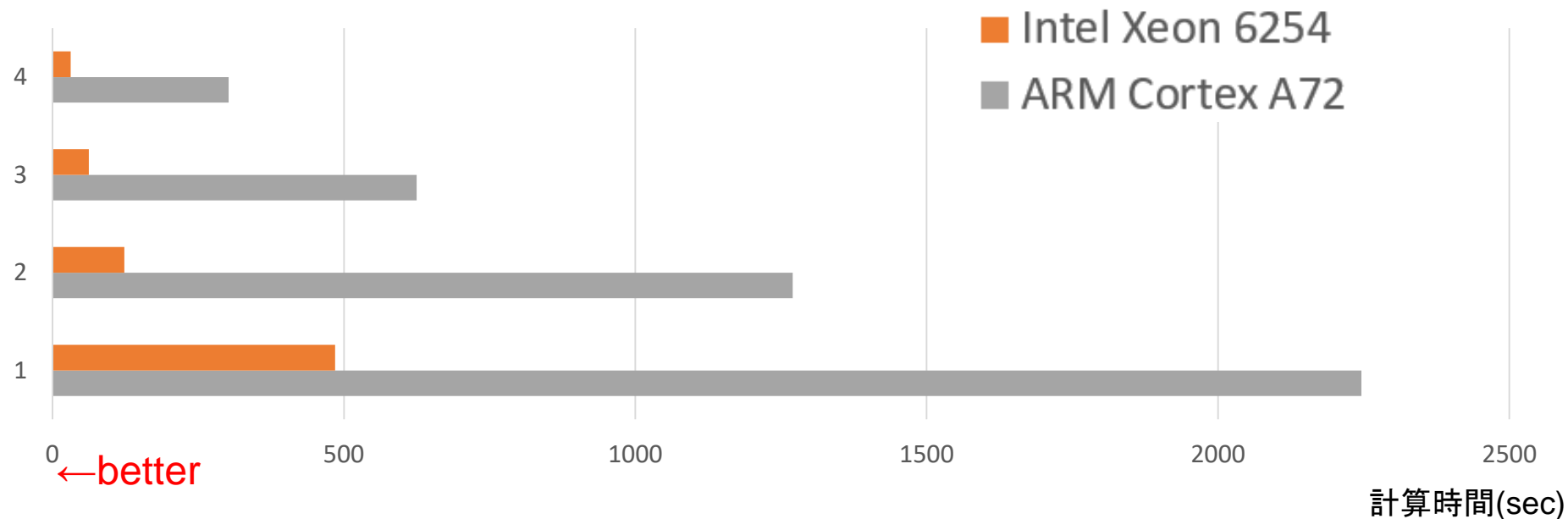
```
nakam@rapi-1:~> sbatch openfoam_pitzdaily.sh
```

実行結果

```
/*-----*¥
| ===== |
| ¥¥ / F ield | OpenFOAM: The Open Source CFD Toolbox |
| ¥¥ / O peration | Version: v1912 |
| ¥¥ / A nd | Website: www.openfoam.com |
| ¥¥/ M anipulation |
¥*-----*/
Build : _f3950763fe-20191219 OPENFOAM=1912
Arch : "LSB;label=32;scalar=64"
Exec : /opt/OpenFOAM-8/OpenFOAM-v1912/platforms/linuxARM64GccDPInt32Opt/bin/interFoam
-parallel
Date : Jan 15 2021
Time : 17:39:09
Host : rapi-2
... (中略) ...
End
Finalising parallel run
```

ベンチマーク比較 (Intel Xeon vs ARM Cortex)

CPUコア数
(core)



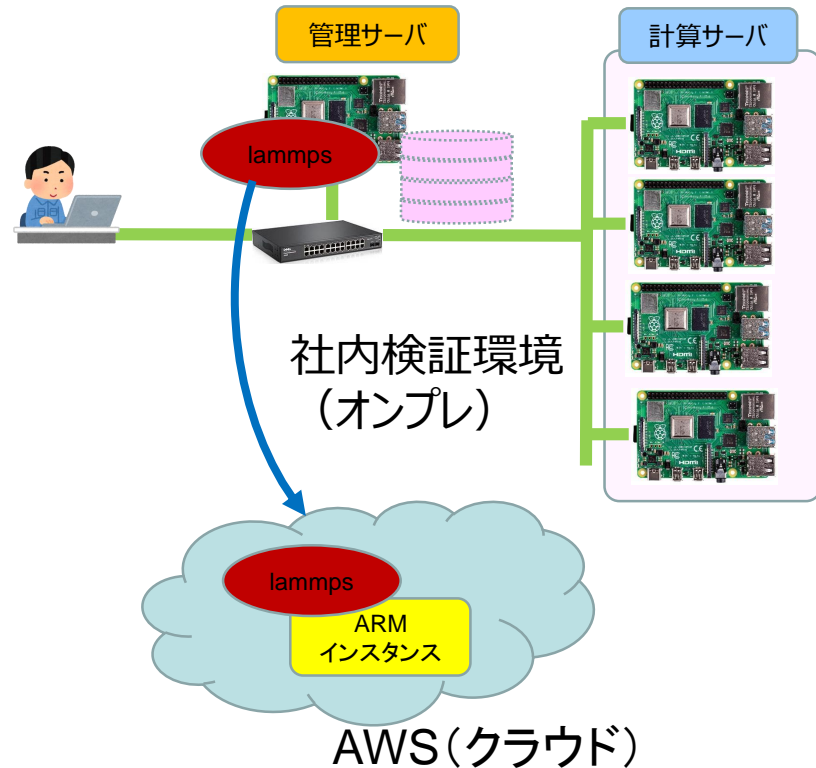
| CPU | 価格 | 性能 (小さいほど性能高い) |
|-----------------|--------|-------------------|
| ARM Cortex A72 | 8,900円 | 2246秒/core |
| Intel Xeon 6254 | 110万円 | 485秒/core |

124倍 (Price comparison: Intel Xeon 6254 is 124 times more expensive than ARM Cortex A72)

約5倍 (Performance comparison: ARM Cortex A72 is approximately 5 times more performant than Intel Xeon 6254)

3.クラウドバーステイングについて ～AWS & Slurm～

- モチベーション（目標、やりたいこと）
 - Raspberry Piではやっぱり**非力**。もっと**速く**計算を終らせたい。
 - AWSに**ARM**インスタンスがある
- 必要なこと
 - クラウド環境で Raspberry Piのバイナリが動く
 - Raspberry Piからクラウドにつながる
 - ARMインスタンスをバースティングするしくみ



● 検証結果

- LAMMPSをAWSのインスタンスで実行したら、3倍弱の性能向上を確認できた。

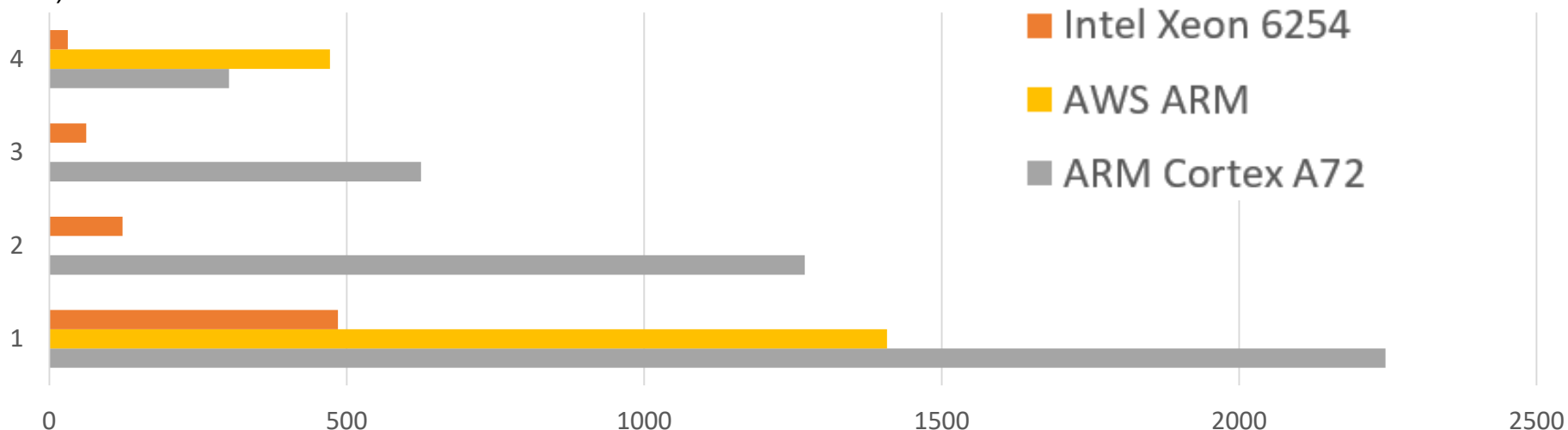
| 項目 | 評価 | 説明 |
|-----------|----|--|
| バイナリが動く | ○ | <p>Raspberry Piと同等の環境構築を行なって動作することを確認できた。</p> <ul style="list-style-type: none">• Raspberry Pi 1node,1core Total wall time: 0:37:26 1node,2cores Total wall time: 0:24:00 1node,4cores Total wall time: 0:21:10 2nodes,8cores Total wall time: 0:10:25 3nodes,12cores Total wall time: 0:06:47 4nodes,16cores Total wall time: 0:05:02• AWS 1node,1core Total wall time: 0:23:28 (a1.medium: 0.0321USD/時間) 1node,4core Total wall time: 0:07:52 (a1.xlarge: 0.1284USD/時間) |
| クラウドにつながる | △ | <p>検証LANを経由して、インターネット経由のsshアクセスはできたが、VPCとのVPN接続までは至らず。</p> |
| バースティング動作 | △ | <p>AWSが公開するSlurmを利用したバースティング環境CloudFormationテンプレート(※1)をつかって、VPC内に閉じた環境の構築に成功。 Raspberry Piからバースティングさせるには以下も実現する必要があるが、そこまでは実現できなかった。</p> <ul style="list-style-type: none">• Raspberry Pi同等環境であるOpenSUSEによるバースティング環境の実現• Raspberry Piでバースティング環境構築 |

※1 <https://github.com/aws-samples/aws-plugin-for-slurm/tree/plugin-v2>

ベンチマーク比較

(Intel Xeon vs ARM Cortex vs AWS ARM)

CPUコア数
(core)



←better

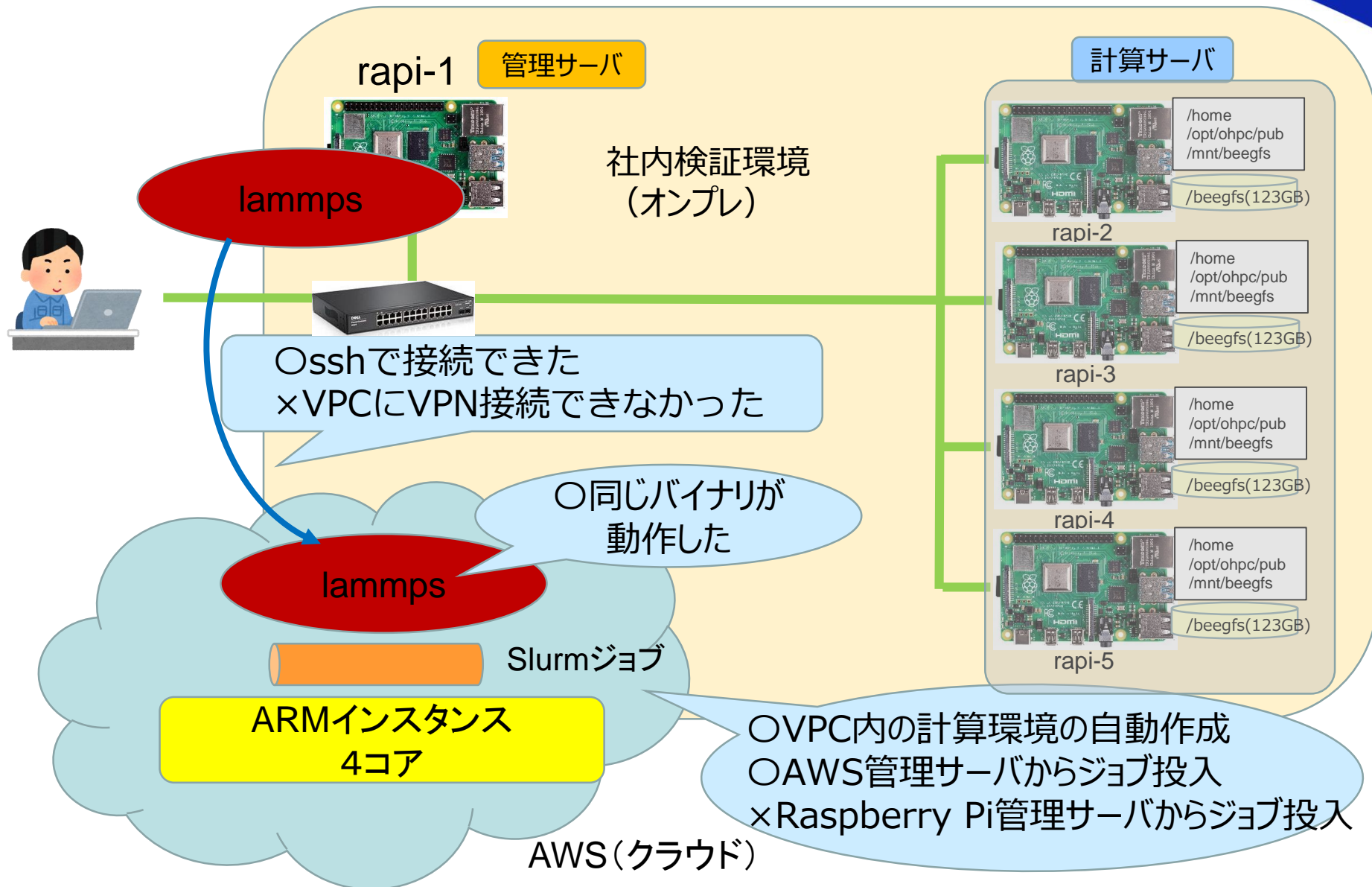
計算時間(sec)

| CPU | 価格 | 性能 (小さいほど性能高い) |
|-----------------|----------|-------------------|
| ARM Cortex A72 | 8,900円 | 2246秒/core |
| AWS a1 instance | 13.48円/h | 1480秒/core |
| Intel Xeon 6254 | 110万円 | 485秒/core |

約9年分

約3倍

ジョブ投入と並列計算



■ ふりかえり

- ✓ バイナリを動くようにするまでの手順が大変だった
- ✓ クラウドバーステイングの仕組みづくりがさらに大変だった
 - Cloudformationのテンプレートファイルを読んでデバッグ
 - AWSの制限事項の理解(使えないAMIがある、など)
- ✓ クラウドでは、**再現させるための努力(Infrastructure as Code的な努力)**が必要。

■ 今後

ARMでのHPCクラスタ環境を作成するために、今回失敗した以下2点を実現したい

- ✓ Raspberry Pi→AWSへ計算ノードをスケーリングしたい（社内手続き含む）
（ジョブの再投入ではなく**オンプレでリソース足りなくなったら自動でクラウドへ**）
- ✓ AWS上でバーステイングする仕組みの構築（オンプレと全く同じ環境作成）

■ HPC技術のRaspberry Piへの適用

- ✓ ARM環境を手軽に試すにはコンテナ含めてインフラ面で**問題なし**
- ✓ HPCアプリケーションも動作するが、**非力**だった

■ Raspberry Piからクラウドバースティング

- ✓ AWSを使って、Raspberry Piに似たARM環境を作成することはできた
- ✓ 自動でオンプレからバースティングする仕組みを構築することは**大変**
- ✓ Raspberry Piの**リソース不足を補う**手段の一つになる

■ ベンチマーク比較

- ✓ 現状は**Intel**に軍配が上がる
- ✓ 今後の**ARM**の開発動向に期待!!