



MISHIMA.SYK#18 ONLINE 2022.04.16

KNIMEをSlurmで動かす話

@tkochi0603

自己紹介：@tkochi0603

▶伊豆半島にある製薬会社で、ITインフラ周りを
担当しています

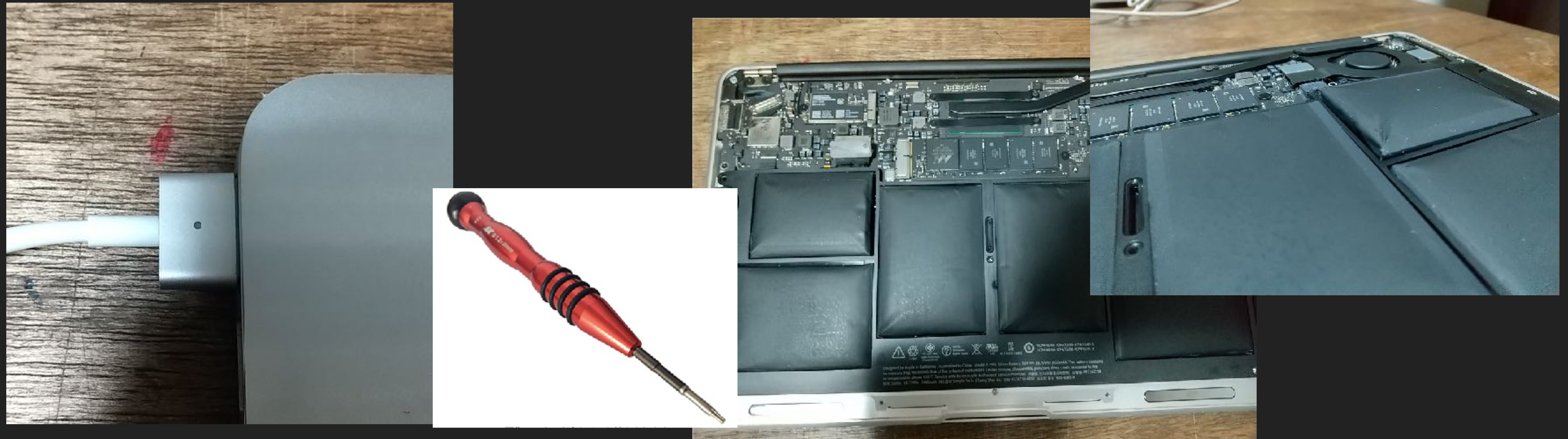
有明海とハイボール



▶ 今月、特急A列車で行こうに乗った。とっても良かった

小話：ある日、

- ▶ Mac book Air(A1465:2013年)が充電できなくなりました
- ▶ 星型ドライバを買って開けてみたら、バッテリーが超膨張してた
- ▶ 互換バッテリーに交換したら復活した(焦った)



今日の話

- ▶ Q. KNIME Analyticsで書かれた処理を、スケールアップしたいときどうする？
 - ▶ 最近のPCはスペックいいし、杞憂かもしれません

Q. KNIMEのスケールアップ

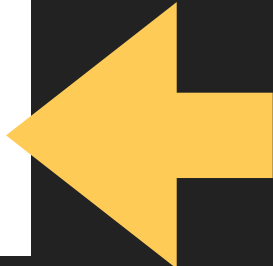
- ▶ A1. KNIME Analyticsを動かすPCのスペックを上げる
 - ▶ それでもいいけど



Q. KNIMEのスケールアップ

- ▶ A2. KNIME Serverを大きめなサーバで動かす？
 - ▶ KNIME Server持っていないし。。買う？
 - ▶ うちも買ったほうがいいかもと、思うけど、クラスタサーバで動かすとなると金かかりそう

ライセンスプラン	KNIME Server Small	KNIME Server Medium	KNIME Server Large
機能概要	少人数のチームで、リモートでワークフローを共有	Webブラウザからワークフローを使用、REST APIへのアクセス	複数マシンへのインストール、大規模なチームで、グローバルなコラボレーション
インストール可能数	1	1	複数台



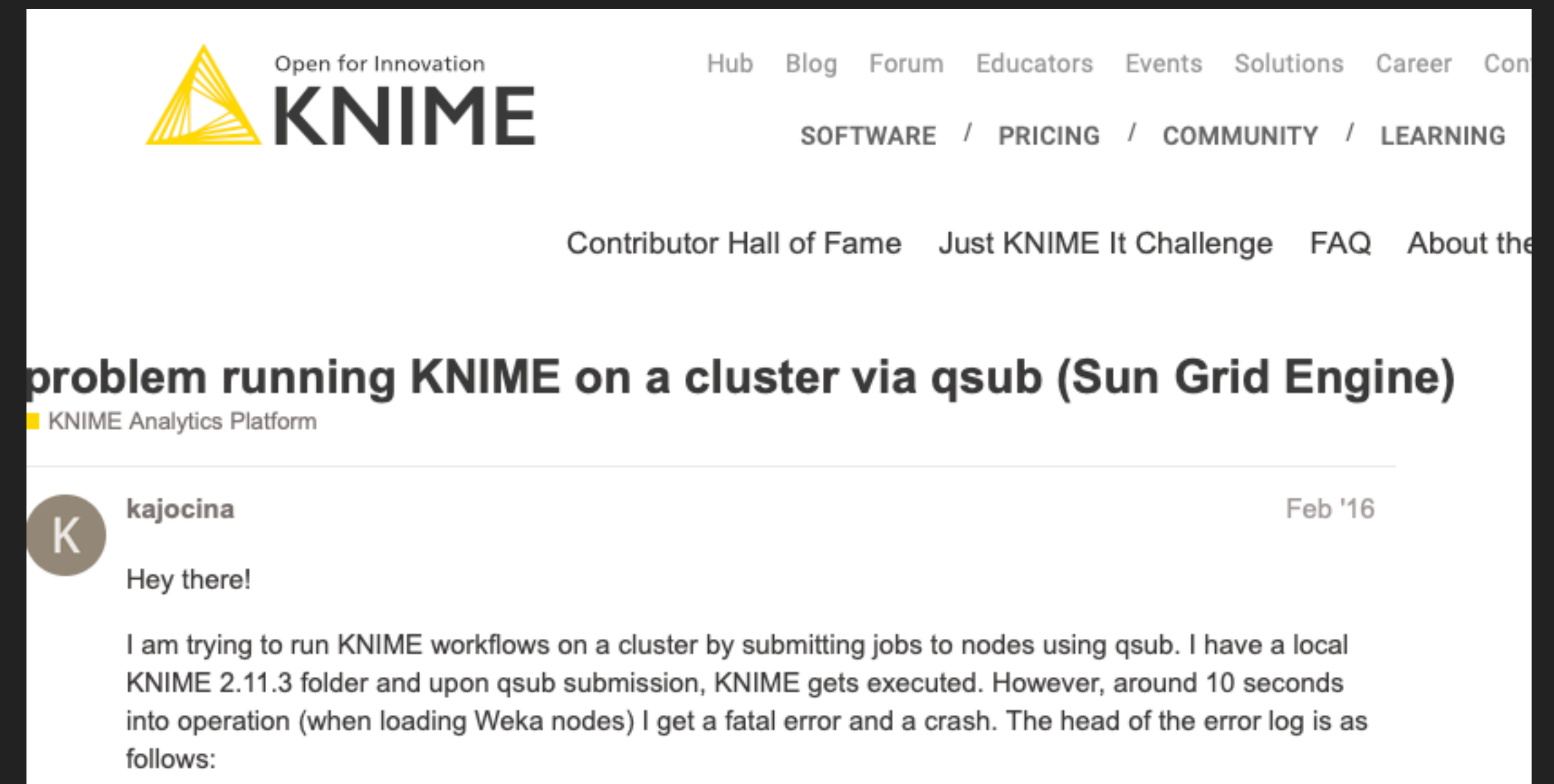
Q. KNIMEのスケールアップ

- ▶ A3. 他の言語に書き換えて、クラスタサーバで動かす？
 - ▶ 書き換えるのめんどくさいw、けど、やるかも

Q. KNIMEのスケールアップ

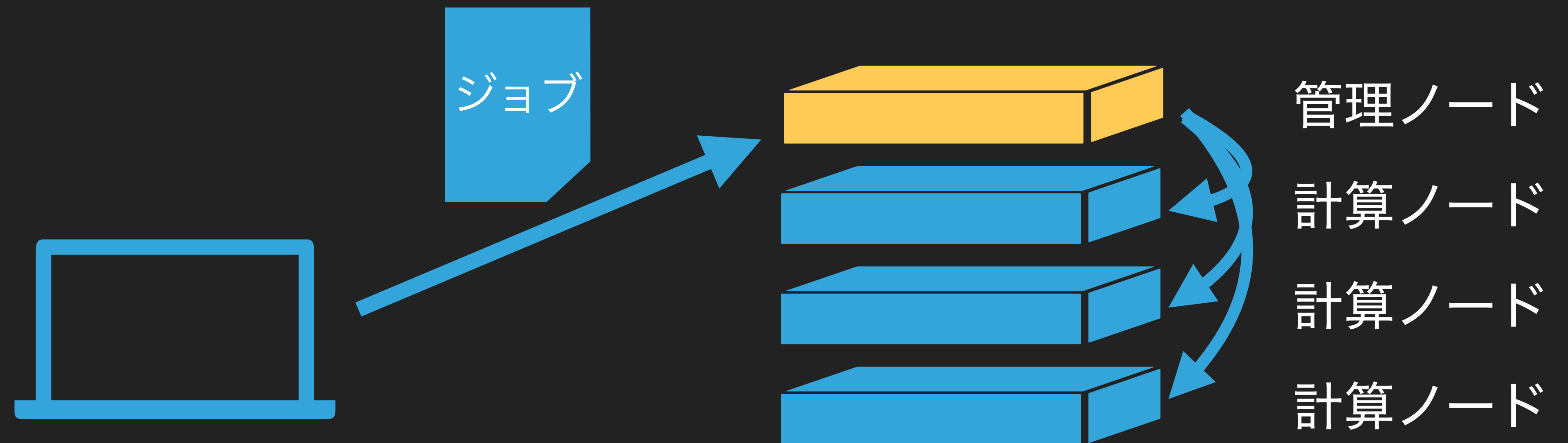
- ▶ A4. KNIME Analyticsをクラスターサーバで動かす
 - ▶ どうやら過去にも取り組んでいた人はいらるみたい

やってみることにした



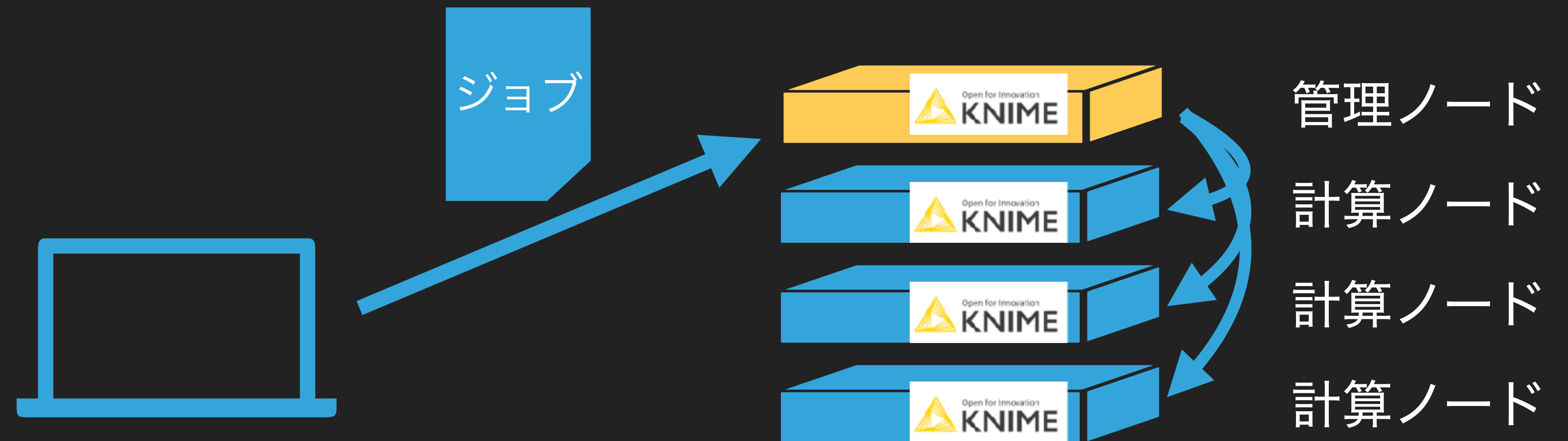
ところで、クラスタサーバって？

- ▶ ジョブを制御する管理ノード(PC)と、計算ノード(PC)からなる
- ▶ 管理ノードに、ジョブリストを渡すと、空いてる計算ノードにジョブを割り振って実行させる



ってことで、

- ▶ 各計算ノードで、KNIMEのBATCHモードを動かすことにした





クラスターサーバの構築

Slurm

どうされてます？すぐに捨ててます？

あるある??

クラスタサーバの構築

- ▶ まだ使えるPC(ガラクタ)が、手元に結構余ってしまった
 - ▶ 留学する人が残していった自作PCクラスタ
 - ▶ クラウド移行して抜け殻となったサーバ
 - ▶ EOLを迎えてVerUpが必要になったOSのサーバ
- ▶ ガラクタで組んでみる

クラスタサーバのソフトウェア

- ▶ 昔は、Sun(Son) Grid Engine、PBS、Torqueでクラスタサーバを組んでいたけど、最近はSlurmらしい
- ▶ AWSのParallel ClusterもSlurmを採用している

クラウドとオンプレで、同じ作法に揃えられますね



Slurmで構築する

- ▶ Ubuntuのパッケージがあったので、全てaptでインストール

計算ノードの制御	Slurmd	全ノードにインストール
----------	--------	-------------

ジョブの管理	Slurmctld	管理ノードにインストール
--------	-----------	--------------

認証	Munge	全ノードにインストール
----	-------	-------------

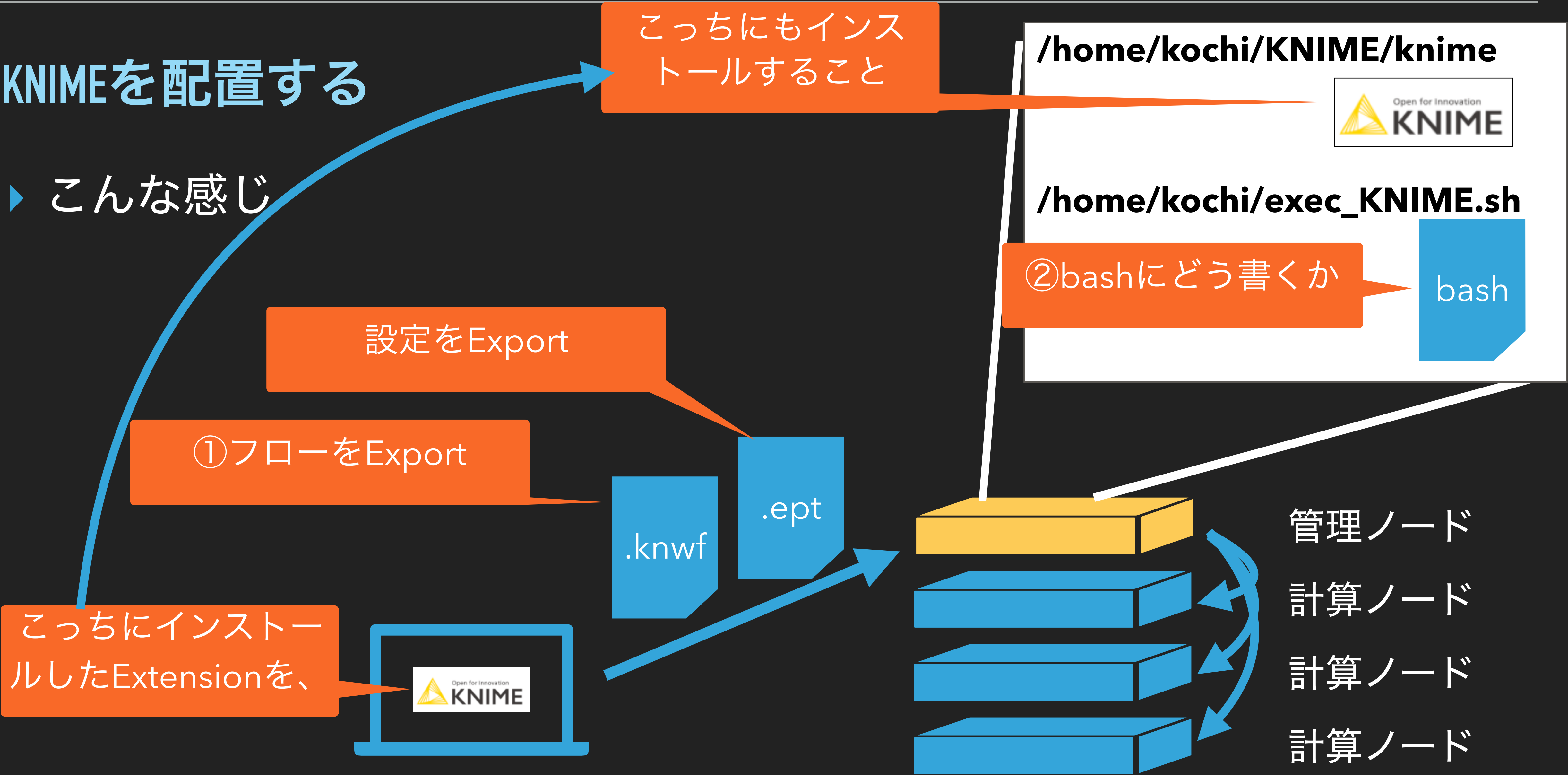
- ▶ slurm.confとmunge.confとサービスの設定
- ▶ 管理ノードの/homeを、計算ノードが/homeとしてマウント
- ▶ pidファイル関連の設定に手こずる
- ▶ あとは、台数増えたらansibleを使う

GridEngineよりも若干
楽な気がする

ってなことで、Slurmで構築できたとさ

KNIMEを配置する

▶ こんな感じ

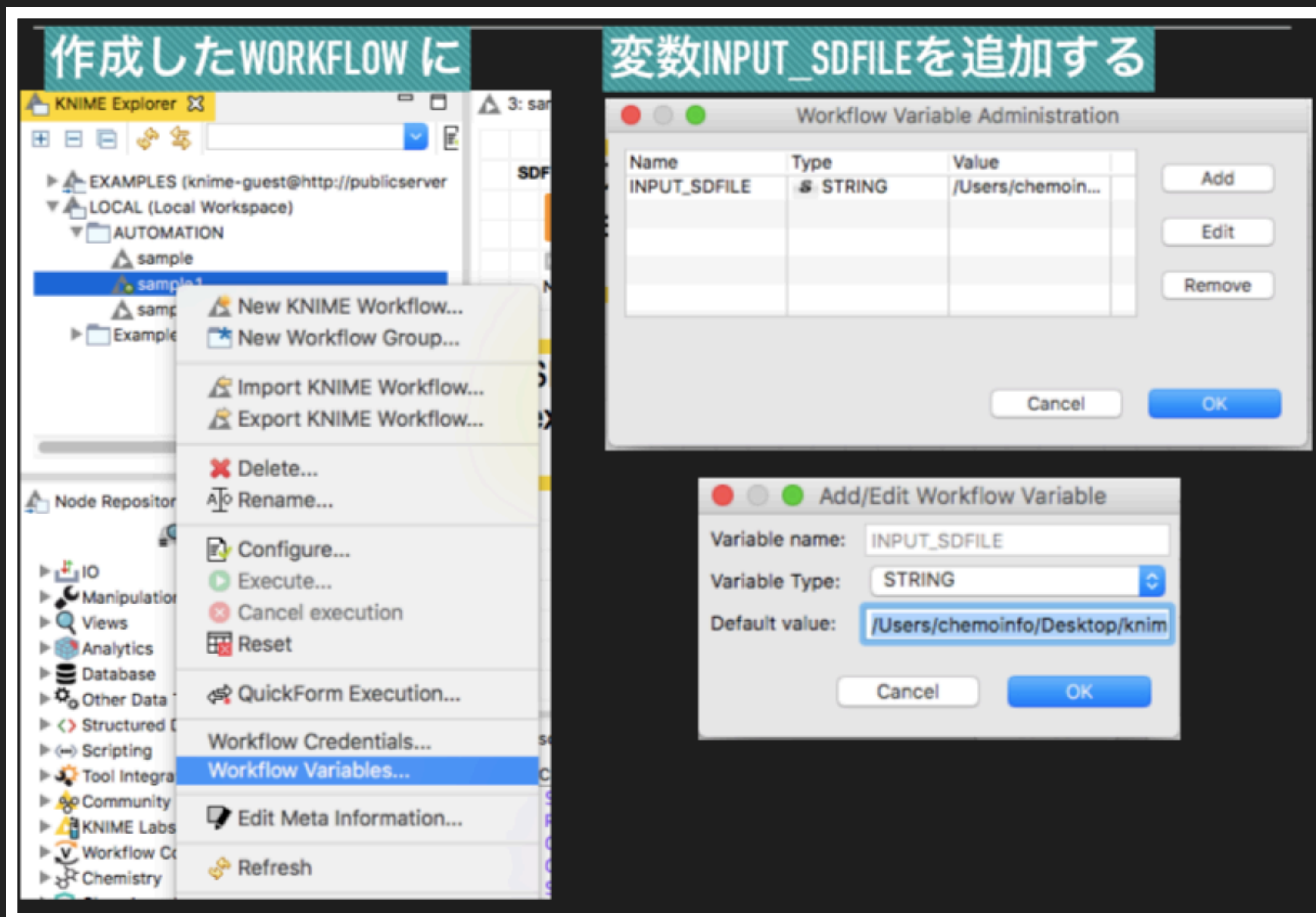




もう少しだ

①フローをEXPORT

- ▶ Workflow Variablesに、**入力と出力の変数**を設定しておく
- ▶ 詳しくは、私が2017年のMishima.syk #10で紹介した資料や、ググればありますね
 - ▶ https://github.com/Mishima-syk/10/blob/master/KNIME_batch_mode_20170709_kochi.pdf



②Bashにどう書くか

```
#!/bin/bash
#SBATCH -p TEST_QUEUE
#SBATCH -o TEST_%j.stdout
#SBATCH -e TEST_%j.stderr
#SBATCH --job-name="TEST_JOB"
```

Slurmから受け取る変数

ここがポイント！
フォルダを指定すると、フォルダに
ロックがかかり、複数同時に起動で
きない

Exportしたフローファイル

WORKFLOWFILE=/home/kochi/Work/sample.knwf

OPTFILE=/home/kochi/Work/pref_20220410.epf

PARAM01=/home/kochi/Work/inp01.csv

PARAM02=/home/kochi/Work/out01.csv

Exportした設定ファイル

実行してるのはこの行だ
け。他の行は変数

```
/home/chemoinfo/KNIME/knime_4.5.2/knime -nosplash -reset -nosave --  
launcher.suppressErrors -application org.knime.product.KNIME_BATCH_APPLICATION -  
preferences=${OPTFILE} -workflowFile=${WORKFLOWFILE} -  
workflow.variable=INP_FILE_PATH,${PARAM01},String -  
workflow.variable=OUT_FILE_PATH,${PARAM02},String
```

②Bashにどう書くか

で、

```
> sbatch exec_ENIME.sh
```

と、実行すれば動作するけど、1ジョブを1ノードで実行するだけなので、

②Bashにどう書くか

```
#!/bin/bash
#SBATCH -p TEST_QUEUE
#SBATCH -o TEST_%j.stdout
#SBATCH -e TEST_%j.stderr
#SBATCH --job-name="TEST_JOB"
```

WORKFLOWFILE=/home/kochi/Work/sample.knwf

OPTFILE=/home/kochi/Work/pref_20220410.epf

PARAM01=\$1

PARAM02=\$2

入出力ファイルを引数で
受け取るようにする

```
/home/chemoinfo/KNIME/knime_4.5.2/knime -nosplash -reset -nosave --
launcher.suppressErrors -application org.knime.product.KNIME_BATCH_APPLICATION -
preferences=${OPTFILE} -workflowFile=${WORKFLOWFILE} -
workflow.variable=INP_FILE_PATH,${PARAM01},String -
workflow.variable=OUT_FILE_PATH,${PARAM02},String
```

②Bashにどう書くか”

と、変更して、

```
> sbatch exec_ENIME.sh /home/kochi/input_1.csv /home/kochi/output_1.csv
```

引数を渡すようにしておいてから、

```
for ia in `seq 0 9`; do  
    sbatch exec_ENIME.sh /home/kochi/input_${ia}.csv /home/kochi/output_${ia}.csv  
done
```

などと、実行することで、10並列でKNIMEを実行できる

まとめ

- ▶ KNIME AnalyticsをCluster Serverで並列実行できる
- ▶ が、bashは必要w
- ▶ なので、大規模で計算したいKNIMEerが次に学ぶべき言語はbash ですね。
- ▶ あと、
 - ▶ Mishima.syk が、気軽にLTできる会になるといいなと思ってます。
 - ▶ 社内に理解してくれる人が少ないようなトピックスを、社外の人と話して、情報交換。
- ▶ Ubuntu 22.04 LTSが来週リリースされる

Ubuntu 22.04 LTS (Jammy Jellyfish) Beta