スーパーコンピュータ@京大
～システム構成と使い方～

OpenMX講習会
2019年12月2日(月)

京都大学　企画・情報部　情報基盤課
スーパーコンピューティング掛

センターのスパコン概要 - 目次

- システム構成と特徴
- システムの使い方(簡略版)
  - ログイン方法
  - バッチ処理システム（PBS）
  - ディスク構成と使い方
- OpenMXの使い方

システム構成と特徴

スーパーコンピュータ @ 京大

Camphor 2 (System A)

CRAY *XC40*

intel Xeon Phi KNL 68cores 1.4GHz x 1 /node

#nodes	= 1,800
#total cores	= 68 cores x 1,800 ➔ 122,400 cores
Peak performance	= 3.05TFops x 1,800 ➔ 5.48 PFlops
Memory capacity	= (96+16 GB) x 1,800 ➔ 196.9 TB
Burst buffer	= 230 TB, 200 GB/sec DataWarp

Storage

DataDirect NETWORKS *ExaScaler (SFA14K)*

Disk capacity	= 24 PB
Bandwidth	= 150 GB/sec
Burst buffer	= 230 TB, 250 GB/sec IME

高速通信網 InfiniBand EDR/FDR

Laurel 2 (System B)

CRAY *CS400 2820XT*

intel Xeon Broadwell 18cores 2.1GHz x 2 /node

#nodes	= 850
#total cores	= 36 cores x 850 ➔ 30,600 cores
Peak performance	= 1.21 TFlops x 850 ➔ 1.03 PFlops
Memory capacity	= 128 GB x 850 ➔ 106.3 TB

Cinnamon 2 (System C)

CRAY *CS400 4840X*

intel Xeon Haswell 18cores 2.3GHz x 4 /node

#nodes	= 16
#total cores	= 72 cores x 16 ➔ 1,152 cores
Peak performance	= 2.65 TFlops x 16 ➔ 42.4 TFlops
Memory capacity	= 3 TB x 16 ➔ 48.0 TB

高速通信網 Omni-Path

スーパーコンピュータ @ 京大

Camphor 2 (System A)

本日の実習で利用

#nodes	= 1,800
#total cores	= 68 cores x 1,800 ➔ 122,400 cores
Peak performance	= 3.05TFops x 1,800 ➔ 5.48 PFlops
Memory capacity	= (96+16 GB) x 1,800 ➔ 196.9 TB
Burst buffer	= 230 TB, 200 GB/sec DataWarp

Storage

DataDirect NETWORKS *ExaScaler (SFA14K)*

Disk capacity	= 24 PB
Bandwidth	= 150 GB/sec
Burst buffer	= 230 TB, 250 GB/sec IME

高速通信網 InfiniBand EDR/FDR

Laurel 2 (System B)

CRAY *CS400 2820XT*

intel Xeon Broadwell 18cores 2.1GHz x 2 /node

#nodes	= 850
#total cores	= 36 cores x 850 ➔ 30,600 cores
Peak performance	= 1.21 TFlops x 850 ➔ 1.03 PFlops
Memory capacity	= 128 GB x 850 ➔ 106.3 TB

Cinnamon 2 (System C)

CRAY *CS400 4840X*

intel Xeon Haswell 18cores 2.3GHz x 4 /node

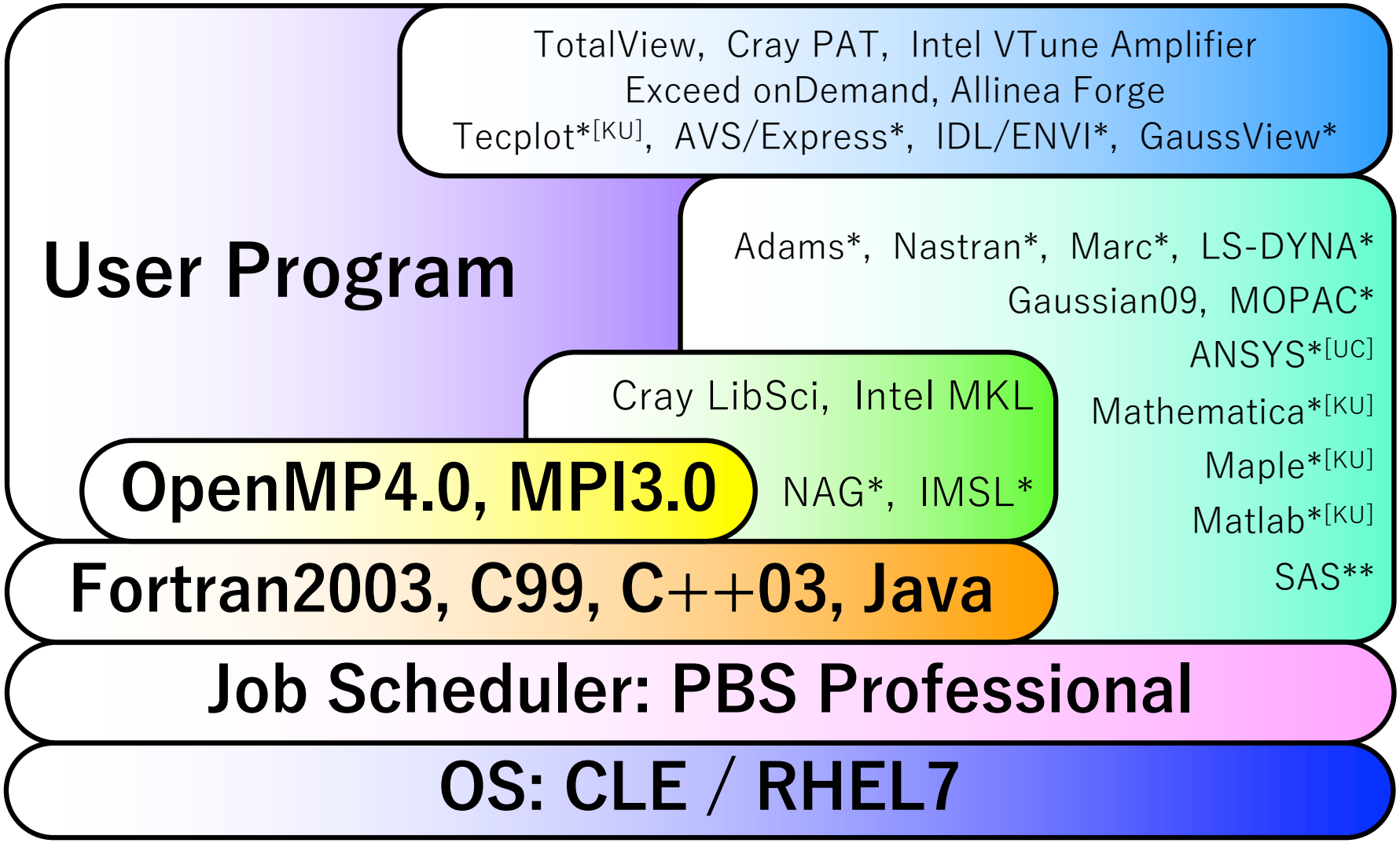
#nodes	= 16
#total cores	= 72 cores x 16 ➔ 1,152 cores
Peak performance	= 2.65 TFlops x 16 ➔ 42.4 TFlops
Memory capacity	= 3 TB x 16 ➔ 48.0 TB

高速通信網 Omni-Path

コンパイラ & ライブラリ

システム	コンパイラ	MPI	数値計算ライブラリ (BLAS, LAPACK)
Camphor 2 (System A)	Cray Compiler (デフォルト) Intel Compiler PGI Compiler GNU Compiler	Cray MPI	Cray LibSci Intel MKL など
Laurel 2 (System B) Cinnamon 2 (System C)	Intel Compiler (デフォルト) Cray Compiler PGI Compiler GNU Compiler	Intel MPI	Intel MKL Cray LibSci IMSL, NAG など

ソフトウエアスタック

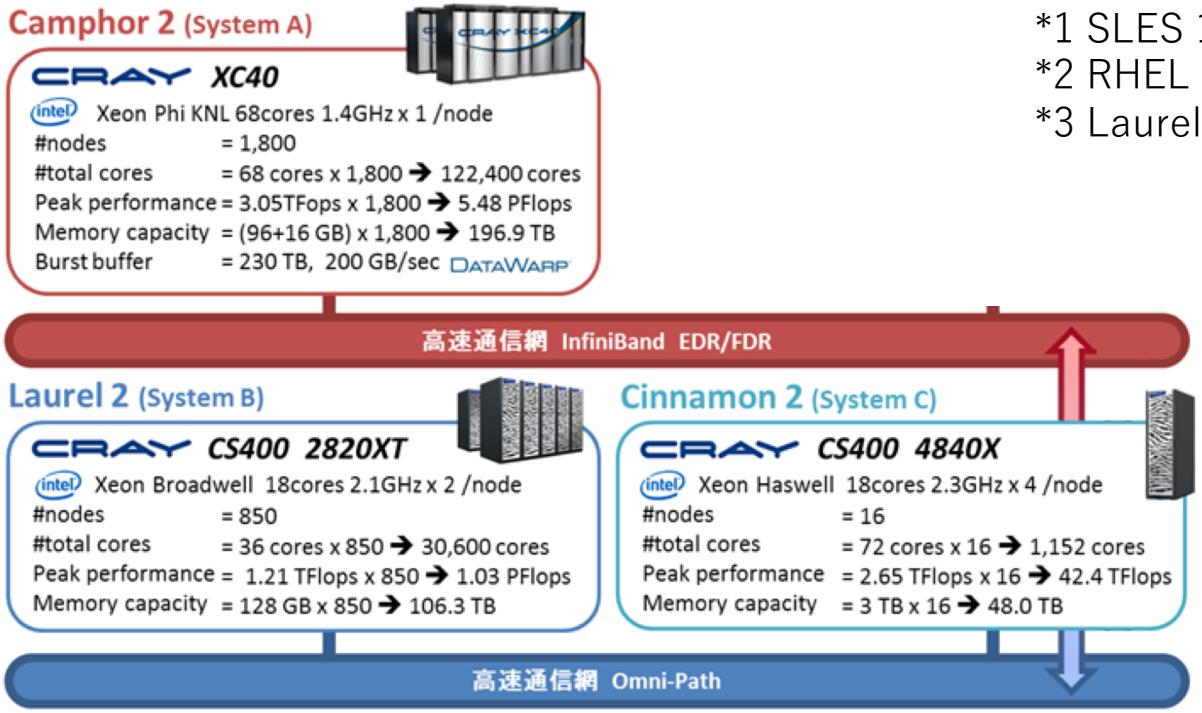


* only for Laurel & Cinnamon
** only for login nodes of Laurel & Cinnamon
[KU] = Available for Kyoto University members only.
[UC] = Available for organizations that has agreed on the application user consortium arrangement.

システムの使い方

ログインノードの構成

システム	ホスト名（FQDN）	OS	台数	備考
Camphor 2 (System A)	camphor.kudpc.kyoto-u.ac.jp	SLES 12(*1)	2台	クスノキ 申請者のみ
Laurel 2 (System B)	laurel.kudpc.kyoto-u.ac.jp	RHEL 7 (*2)	3台(*3)	ゲッケイジュ 全員ログイン可
Cinnamon 2 (System C)	cinnamon.kudpc.kyoto-u.ac.jp	RHEL 7 (*2)	3台(*3)	全員ログイン可

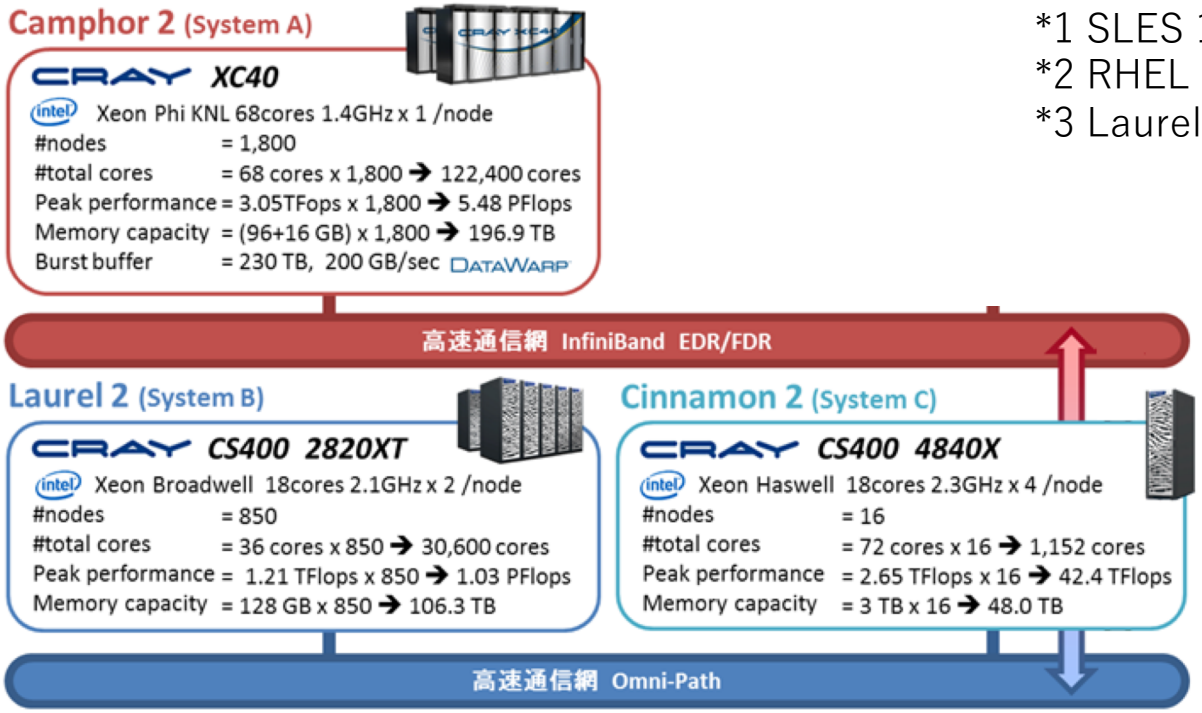


*1 SLES 12: SUSE Linux Enterprise Server 12
*2 RHEL 7 : Red Hat Enterprise Linux 7
*3 Laurel2とCinnamon2で共用

ログインノードの構成

システム	ホスト名（FQDN）	OS	台数	備考
Camphor 2 (System A)	camphor.kudpc.kyoto-u.ac.jp	SL		
Laurel 2 (System B)	laurel.kudpc.kyoto-u.ac.jp	RHEL 7 (*2)	3台(*3)	ゲッケイジュ 全員ログイン可
Cinnamon 2 (System C)	cinnamon.kudpc.kyoto-u.ac.jp	RHEL 7 (*2)	3台(*3)	全員ログイン可

本日の実習で利用

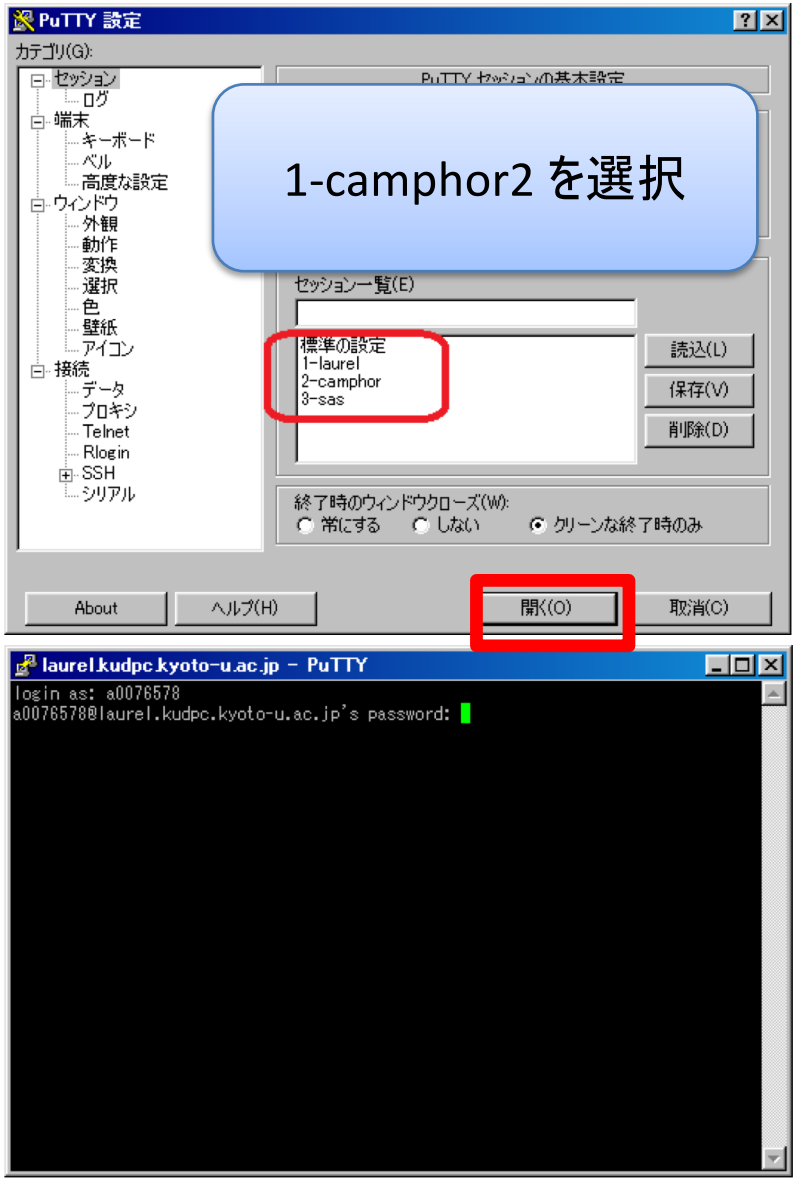


*1 SLES 12: SUSE Linux Enterprise Server 12
*2 RHEL 7 : Red Hat Enterprise Linux 7
*3 Laurel2とCinnamon2で共用

スパコンへのログイン１（PuTTY）

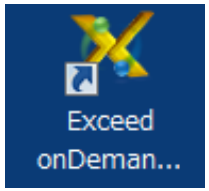
- PuTTYを起動
  - デスクトップのショートカットを
    ダブルクリック

- セッション:
  - **1-camphor2**
    を選択し、**開く** をクリック
- PuTTYセキュリティ警告が表示された場合は **はい** をクリック
- ユーザIDとパスワードは講習会ID通知票のものを使用
  - login as： ユーザIDを入力してEnter
  - Password： パスワードを入力してEnter
    （入力した内容は表示されない）
- ユーザ名@ホスト名 形式の
  プロンプトが表示されれば接続完了

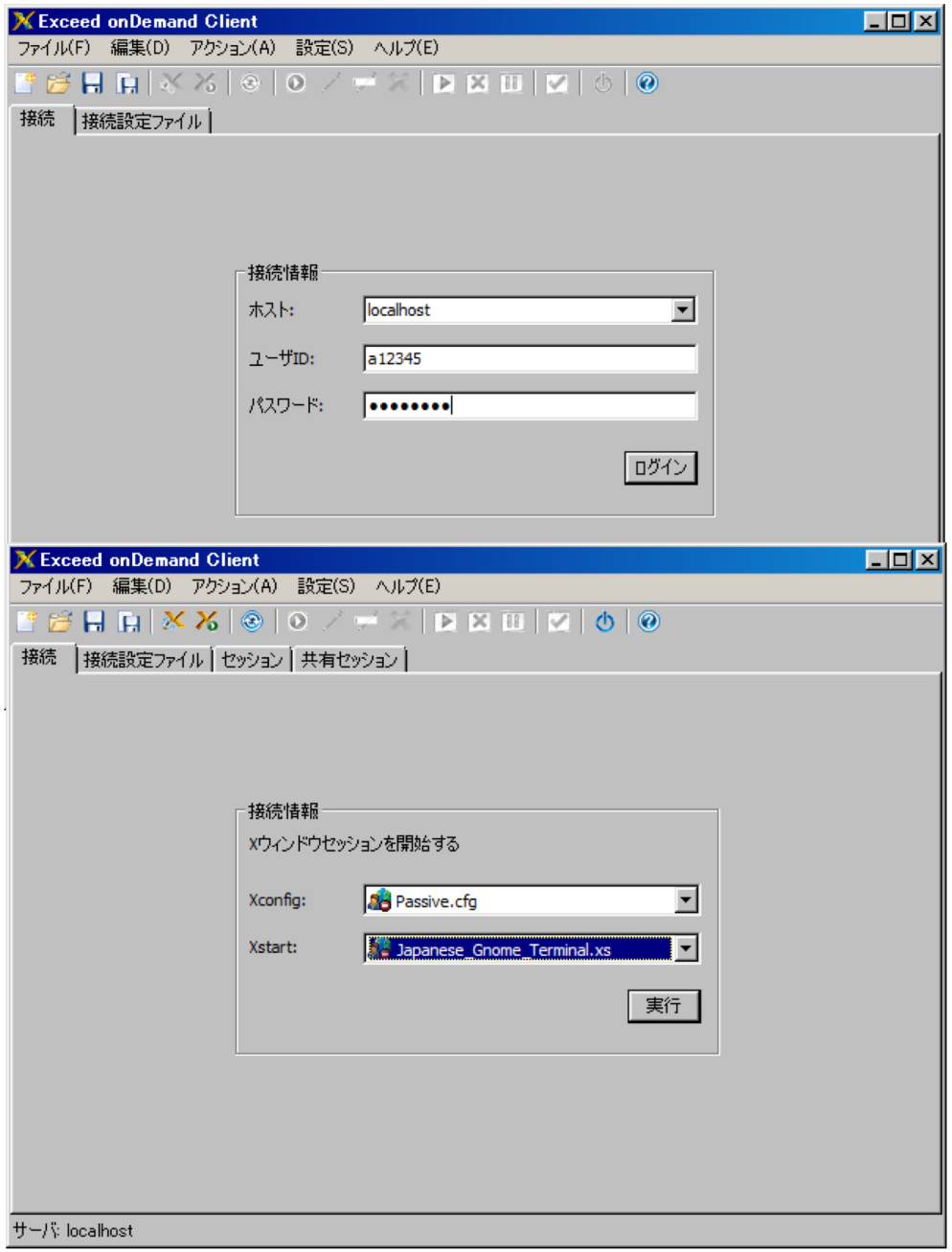


スパコンへのログイン2（EoD）

- Exceed onDemandを起動
  - デスクトップのショートカットをダブルクリック



- ホスト:
  - **localhost**
- ユーザIDとパスワードは講習会ID通知票にあるものを使用
  - 入力後　ログインボタンをクリック
- Xconfig
  - **Passive.cfg**
- Xstart
  - **Japanese_Gnome_Terminal.xs**
- 実行ボタンをクリック



会話型処理での利用

- tssrun：計算ノードで会話形のプログラムを実行するコマンド
  - $ tssrun ./a.out
  - $ tssrun -A p=8 ./a.out（システムA)
  - $ tssrun -A p=8 mpiexec.hydra ./a.out （システムB,C)
- xrun：計算ノードでGUIプログラムを実行するコマンド
  - $ xrun matlab
- プロセスリミット（利用できる資源を制限）
  - tssrun / xrun コマンドでキューを指定しない(会話型キューを利用)場合
    - 計算資源：最大1ノード（システムCは1ソケット）
    - 経過時間：1時間（標準）、24時間（最大） （システムA,B,C)
  - tssrun / xrun コマンドでグループキューを指定する場合
    - ${tssrun | xrun} -q キュー名 -ug グループ名 で実行する
      - $ tssrun -q gr19999b -ug gr19999 ./a.out
      - $ xrun -q gr19999b -ug gr19999 matlab
    - 計算資源：割り当てられた資源数
    - 経過時間：1時間（標準）、336時間（最大）
- 詳しくは以下のWebマニュアルを参照
  - 「kudpc tssrun」で [検索]
  - https://web.kudpc.kyoto-u.ac.jp/manual/ja/run/interactive

バッチでの利用（１）

- ジョブスクリプト
  - 何をするかを ジョブスケジューラ （PBS）に伝える
  - シェルスクリプト形式(オプション指定：先頭に「#QSUB」）

```
#!/bin/bash
#QSUB -q  gr19999a
#QSUB -ug gr19999
#QSUB -A  p=4:t=1:c=1:m=1G
#QSUB -W  6:00

 aprun –n $QSUB_PROCS ¥
 –d $QSUB_THREADS ¥
 -N $QSUB_PPN ./a.out
```

オプション	説明
-q キュー名	ジョブを投入するキューを指定 例） -q gr10001a
-ug グループ名	プロセスのグループを指定 例） -ug gr10001
-A 要求リソース	p: プロセス数 t: プロセスあたりのスレッド数 c: プロセスあたりのコア数 m: プロセスあたりのメモリ容量 例）-A p=4:t=1:c=1:m=1G
-W 経過時間	ジョブの実行に必要な経過時間を指定 例） -W 6:0（時:分）

※ PBSに京大環境の独自のオプションが追加されています

- 詳しくは以下のWebマニュアルを参照
  - 「kudpc　バッチ処理」で [検索]
  - https://web.kudpc.kyoto-u.ac.jp/manual/ja/run/

バッチでの利用（２）

- qsub: ジョブをキューに投入するコマンド
  - $ qsub ジョブスクリプトファイル
  - 適切に投入されると、数字.j{a|b|c} というジョブIDが発行される
- qstat : 投入したジョブの状態を確認するコマンド
  -
```
$ qstat
Job id            Name             User             Time Use S Queue
----------------  ---------------- ---------------  -------- - -----
8622.jb             qsubtest.sh      b59999            00:00:00 R gr19999b
```
- qs : 投入したジョブの詳細情報を確認するコマンド
  -
```
$ qs
 QUEUE     USER     JOBID   STATUS PROC THRD CORE   MEM ELAPSE( limit)
 gr19999b  b59999   5610    RUN       4   32   32   10G  01:12( 20:00)
 gr19999b  b59999   5611    PEND      1   32   32   20G  00:00( 01:00)
```
- qcat : 実行中のジョブの途中経過を確認するコマンド
  - $ qcat -o ジョブID : ジョブの標準出力を表示
  - $ qcat -e ジョブID : ジョブの標準エラー出力を表示

```
$ qcat -o 5610
Tue May  1 00:00:01 JST 2016
Subroutine A step1 finished
Subroutine A step2 finished
Subroutine A step3 finished
```

```
$ qcat -e 5610
Tue May  1 00:00:01 JST 2016
STDERR 1
```

バッチでの利用（３）

- qdel: 投入したジョブをキャンセルするコマンド
  - $ qdel ジョブID
- qgroup : グループコースのキューの利用状況を確認するコマンド
  -

```
$ qgroup
 QUEUE    SYS |   RUN  PEND OTHER | ALLOC ( MIN/ STD/ MAX) | READY
-------------------------------------------------------------------
 gr19999b  B  |     1     0     0 |    72 (  36/  72/ 144) | 0
 gr19999b  B  |     0     0     0 |     0 (  144/ 288/ 576) | 288

 QUEUE    USER     |   RUN(ALLOC)  PEND(REQUEST) OTHER(REQUEST)
-------------------------------------------------------------------
 gr19999b b59999   |     1(   72)     0(     0)     0(     0)
```

- ジョブレポート
  - バッチ処理として投入したジョブが終了すると、qsubコマンドを実行したディレクトリにジョブレポートが書き出される
    - {A|B|C}mmddhh.o{ジョブID} : 標準出力、ジョブ情報
    - {A|B|C}mmddhh.e{ジョブID} : 標準エラー出力

- 詳しくは以下のWebマニュアルを参照
  - 「kudpc　バッチ処理」 で [検索]
  - https://web.kudpc.kyoto-u.ac.jp/manual/ja/run/