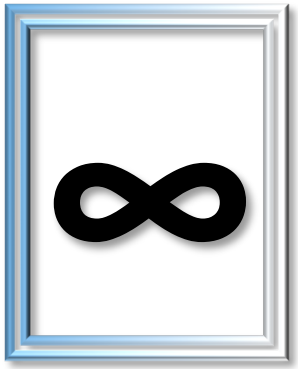
TACC
TEXAS ADVANCED COMPUTING CENTER
WWW.TACC.UTEXAS.EDU

TEXAS
The University of Texas at Austin

ISC High Performance
The HPC Event.

amask: A Tool for Evaluating Affinity Masks in Many-core Processors



PRESENTED BY:

Kent Milfeld

Outline

- Motivation
- Affinity -- what is it
- Ways to show process masks
- Extracting masks with AMASK
- Examples

Motivation

Need to easily view where application processes will execute?

- process count < hardware thread count
- L2/L3 cache sharing (tiles/sockets)
- Minimal Memory Distance
- Messaging/IO process near PCI-e bus

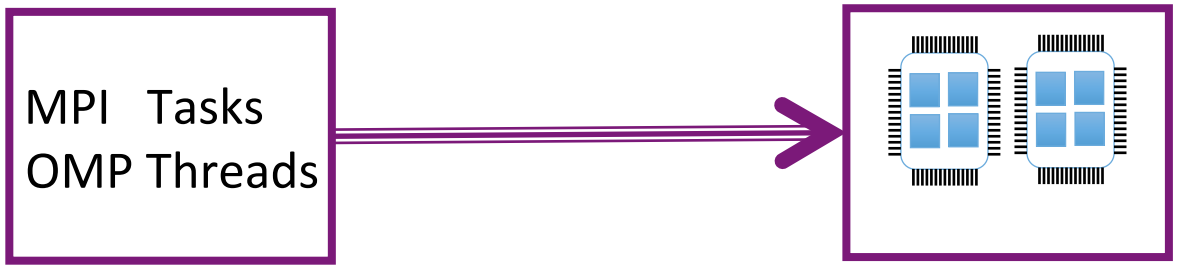
- Xeon-Phi (KNL) Node: 72 Cores 288 hardware threads
- Xeon (SKL-X) Node: 2x28=56 Cores 112 hardware threads

Outline

- Motivation
- Affinity -- what is it
- Ways to show process masks
- Extracting masks with AMASK
- Examples

CPU Affinity -- the mask

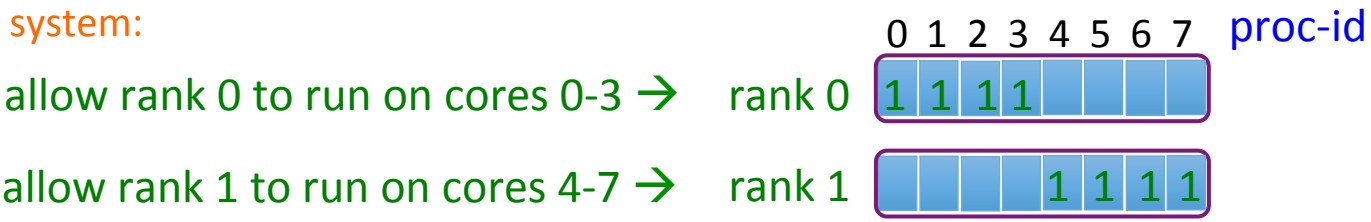
processes map onto processors



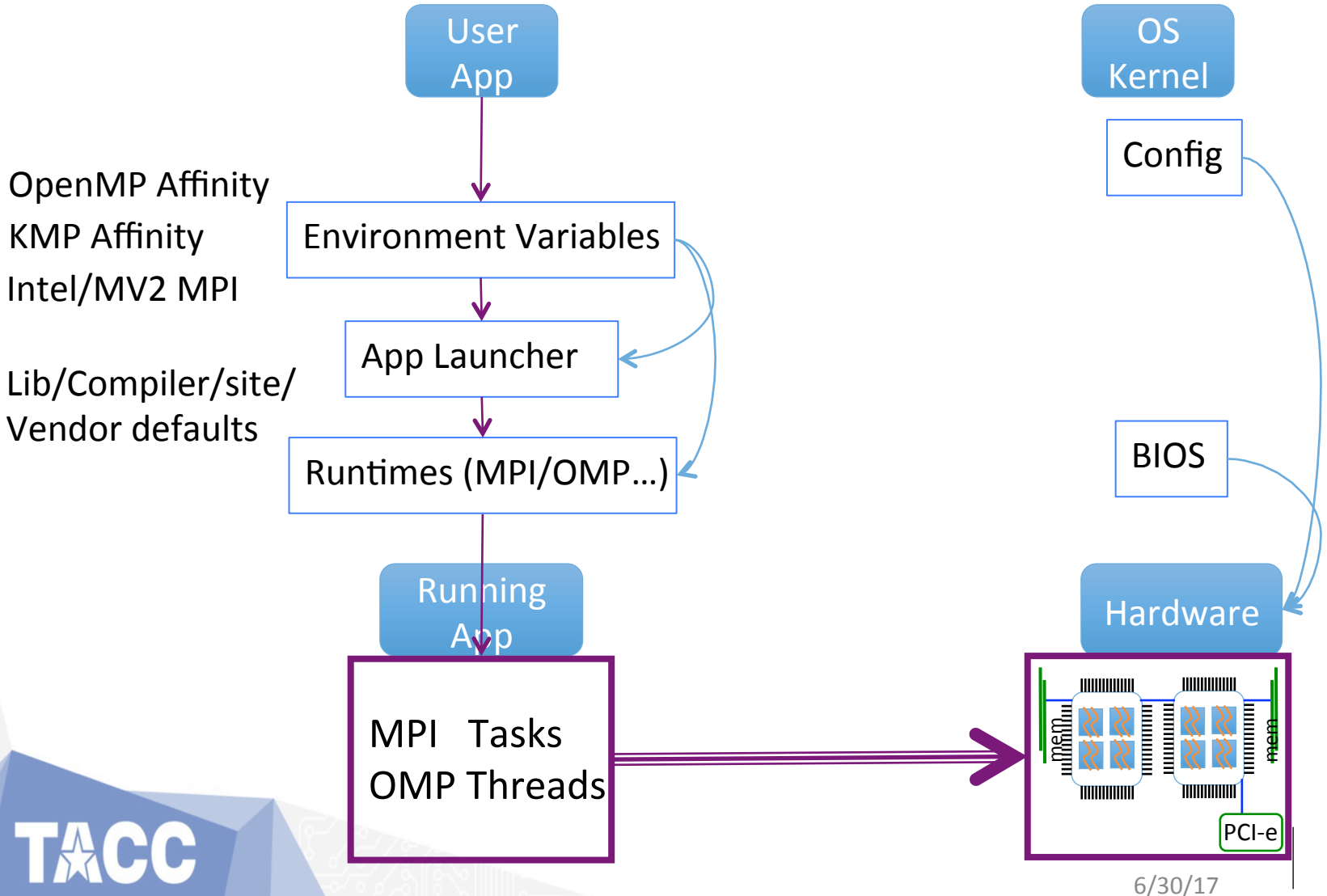
Rank or thread-id proc-id

Each process has a bit mask: a set bit→ process can run on proc-id.

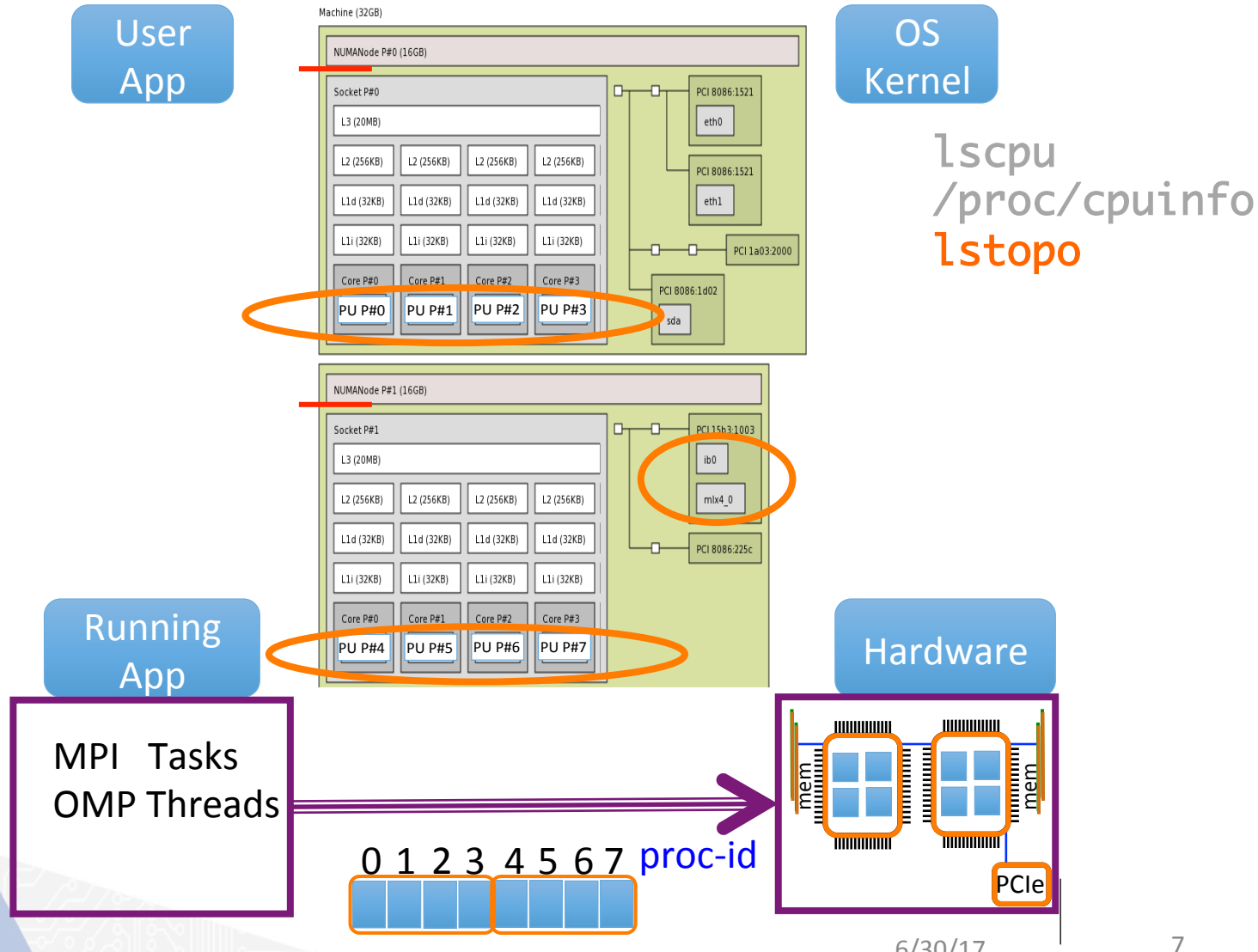
e.g. 8-core system:



Computing Environment Hardware/OS Setup



Unix proc-id info



Unix Process View

User App

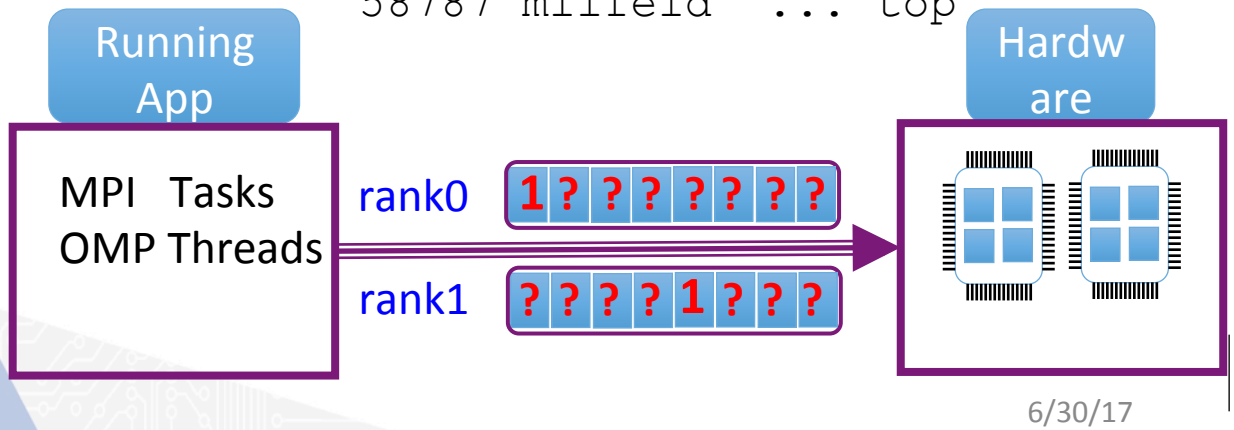
```
mpirun –np 2 …
top -- hit 1 key
```

OS Kernel

```
top -load average:…
Tasks: …

Cpu0  : 99.7%us,  0.3%sy,...   0.0%id, ...
Cpu1  :  0.0%us,  0.0%sy,... 100.0%id, ...
Cpu2  :  0.0%us,  0.0%sy,... 100.0%id, ...
Cpu3  :  0.3%us,  0.3%sy,...  99.3%id, ...
Cpu4  :100.0%us,  0.0%sy,...   0.0%id, ...
Cpu5  :  0.0%us,  0.0%sy,... 100.0%id, ...
Cpu6  :  0.0%us,  0.0%sy,... 100.0%id, ...
Cpu7  :  0.0%us,  0.0%sy,... 100.0%id, ...

  PID USER      ... COMMAND
60673 milfeld   ... mpi_affinity
60672 milfeld   ... mpi_affinity
58787 milfeld   ... top
```



Outline

- Motivation
- Affinity -- what is it
- Ways to show process masks
- Extracting masks with AMASK
- Examples

Viewing Affinity: Intel I_MPI_DEBUG=4

```
KNL  IMPI   export I_MPI_DEBUG=4
mpirun -np 16 my_favorite_app
```

```
Rank Node                                    Pin cpu
 0   c401 {0,1,2,3,4,68,69,70,71,136,137,138,139,204,205,206,207}
 1   c401 {5,6,7,8,72,73,74,75,76,140,141,142,143,208,209,210,211}
 2   c401 {9,10,11,12,77,78,79,80,144,145,146,147,148,212,213,214,215}
 3   c401 {13,14,15,16,81,82,83,84,149,150,151,152,216,217,218,219,220}
 4   c401 {17,18,19,20,21,85,86,87,88,153,154,155,156,221,222,223,224}
 5   c401 {22,23,24,25,89,90,91,92,93,157,158,159,160,225,226,227,228}
 6   c401 {26,27,28,29,94,95,96,97,161,162,163,164,165,229,230,231,232}
 7   c401 {30,31,32,33,98,99,100,101,166,167,168,169,233,234,235,236,237}
 8   c401 {34,35,36,37,38,102,103,104,105,170,171,172,173,238,239,240,241}
 9   c401 {39,40,41,42,106,107,108,109,110,174,175,176,177,242,243,244,245}
10   c401 {43,44,45,46,111,112,113,114,178,179,180,181,182,246,247,248,249}
11   c401 {47,48,49,50,115,116,117,118,183,184,185,186,250,251,252,253,254}
12   c401 {51,52,53,54,55,119,120,121,122,187,188,189,190,255,256,257,258}
13   c401 {56,57,58,59,123,124,125,126,127,191,192,193,194,259,260,261,262}
14   c401 {60,61,62,63,128,129,130,131,195,196,197,198,199,263,264,265,266}
15   c401 {64,65,66,67,132,133,134,135,200,201,202,203,267,268,269,270,271}
```

Viewing Affinity: Cray acheck-cray

24 core Hyper-Threaded

aprun -n 8 -N 8 -d 1 -j 1 -cc cpu ./acheck-cray -v

Better! Easier comparison of process masks

But…Needs numbers

```
host    rank pinning mask
nid00013   0   1.......................1.......................
           1   .1.......................1......................
           2   ..1.......................1.....................
           3   ...1.......................1....................
           4   ....1.......................1...................
           5   .....1.......................1..................
           6   ......1.......................1.................
           7   .......1.......................1................
```

Source: BindingExamples_ECMWF.pdf 24 cores/Hyper-Threading

Viewing Affinity: amask…almost

Lonestar5 24 core Hyper-Threading

mpirun -np 8 mpi_affinity -ls

Each row of matrix is an Affinity mask.
A set mask bit = 1

```
rank |    0    |   10    |   20    |   30    |   40    |
0000 1------------------------1-----------------------
0001 -1------------------------1----------------------
0002 --1------------------------1---------------------
0003 ---1------------------------1--------------------
0004 ----1------------------------1-------------------
0005 -----1------------------------1------------------
0006 ------1------------------------1-----------------
0007 -------1------------------------1----------------
```

Viewing Affinity: amask

Lonestar5 24 core Hyper-Threading

mpirun -np 8 mpi_affinity -ls

Each row of matrix is an Affinity mask.
proc-id of set bit = matrix digit + column # in |...|

```
rank |    0    |   10    |   20    |   30    |   40    |
0000 0-----------------------4-----------------------
0001 -1----------------------5----------------------
0002 --2---------------------6---------------------
0003 ---3--------------------7--------------------
0004 ----4-------------------8-------------------
0005 -----5------------------9------------------
0006 ------6-----------------0-----------------
0007 -------7----------------1----------------
```

Outline

- Motivation
- Affinity -- what is it
- Ways to show process masks
- Extracting masks with AMASK
- Examples

How to get masks with amask executables

Pure OpenMP

```
$    export OMP_NUM_THREADS=68
$    amask_omp                    #amask exec
$    my_omp_app
```

Pure MPI

```
$    mpirun -np 4  amask_mpi     #amask exec
$    mpirun -np 4  my_mpi_app
```

Hybrid

```
$    export OMP_NUM_THREADS=16
$    mpirun –np 17 amask_hybrid  #amask exec
$    mpirun –np 17 my_mpiomp_app
```

How to get masks with API

Pure OpenMP

```
$ export OMP_NUM_THREADS=68
$ my_omp_app
```

```
...
#pragma omp parallel
  amask_omp();
```

Pure MPI

```
$ mpirun -np 4   my_mpi_app
```

```
MPI_Init(NULL,NULL);
   amask_mpi();
      ...
MPI_Finalize();
```

Hybrid

```
$ export OMP_NUM_THREADS=16
$ mpirun –np 17 my_mpiomp_app
```

```
MPI_Init(NULL,NULL);
...
 #pragma omp parallel
   amask_hybrid();
...
MPI_Finalize();
```

Outline

- Motivation
- Affinity -- what is it
- Ways to show process masks
- Extracting masks with AMASK
- Examples

The need to know your affinity mask…

```
# MVAPICH2 MPI

mpirun –np 2 amask_mpi

rank |    0    |   10
0000 0---------------
0001 -1--------------
```

```
# MVAPICH2 MPI

mpirun …tacc_affinity amask_mpi

rank |    0    |   10
0000 01234567--------
0001 --------89012345
```

```
# Intel MPI

mpirun –np 2 amask_mpi

rank |    0    |   10
0000 --------89012345
0001 01234567--------
```

What about … hyper-threading?

- There are many reasons to assign processes to proc-ids which have specific hardware capabilities.

  NUMA nodes
  Sockets
  Tiles
  PCI Interfaces
  …
  Hyperthreads

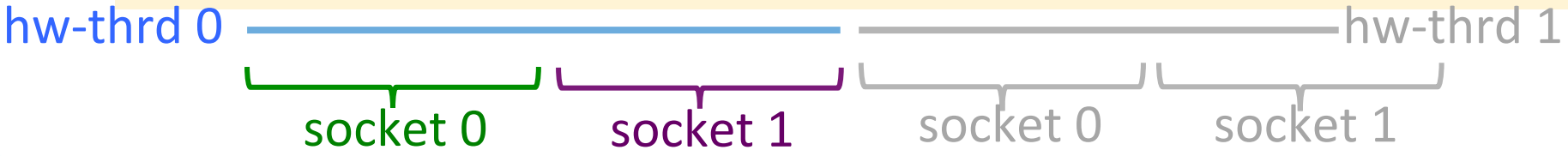
What about hyper-threading…

Hyper-Threaded systems 2 x 12 cores → 48 hardware threads

```
$ mpirun -np 4 amask_mpi  -ls  # list as SMT mask

      Each row of matrix is an Affinity mask.
      A set mask bit = matrix digit + column group # in |...|

      rank |    0    |   10    |   20    |   30    |   40    |
      0000 012345------------------456789-----------------
      0001 ------678901------------------012345-----------
      0002 ------------234567------------------678901------
      0003 ------------------890123------------------234567
```



What about hyper-threading…

Hyper-Threaded systems 2 x 12 cores → 48 hardware threads

Core-Centric MAP

```
$ mpirun -np 4 amask_mpi
```

Each row of matrix is a CORE mask for a HW-thread.
core id = matrix digit + column group # in |...|
proc-id = core id + add 24 to each additional row

Better! But not that Helpful for 2 SMTs/core

```
rank |    0    |   10    |   20    |
0000 012345==================          HW-thread 0
     012345------------------          HW-thread 1
0001 ======678901============
     ------678901------------
0002 ============234567======
     ------------234567------
0003 ==================890123
     ------------------890123
```

What about hyper-threading…

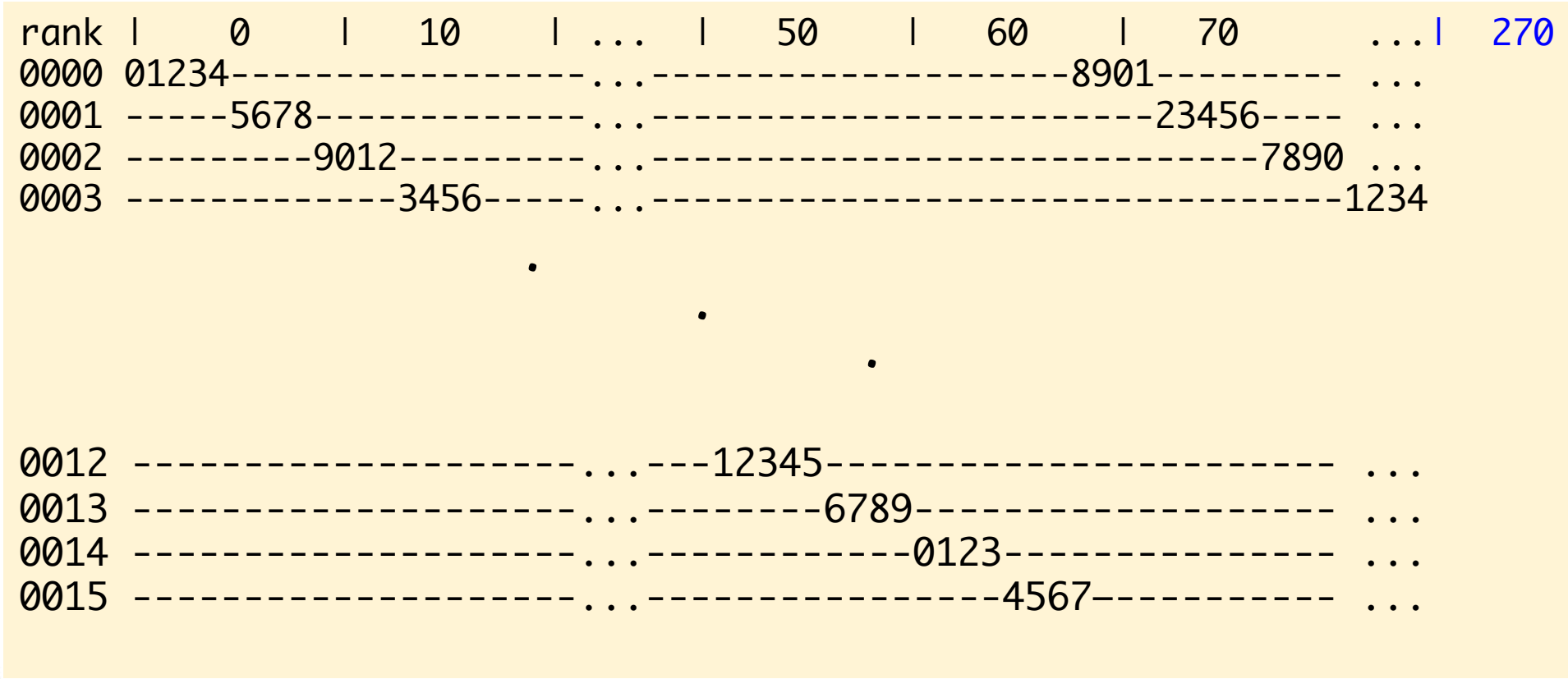
TACC KNL: 4 SMTs x 68 cores = 272 hardware threads

```
$ mpirun –np 16 amask_mpi -ls  #for hybrid, export OMP_NUM_THREADS=17
```

```
 Each row of matrix is an Affinity mask. A set mask bit = matrix digit + column # in |...|
rank |      | 10  | 20  | 30  | 40  | 50  | 60  | 70  | 80  | 90  | 100 | 110 | 120 | 130 | 140 | 150 | 160 | 170 | 180 | 190 | 200 | 210 | 220 | 230 | 240 | 250 | 260 |
0000 01234-------------------------------------------8901-------------------------------------------6789-------------------------------------------4567-----------------------------------------------
0001 -----5678----------------------------------------23456-----------------------------------------0123-----------------------------------------8901-----------------------------------------------
0002 ---------9012-------------------------------------7890-------------------------------------45678-------------------------------------2345-----------------------------------------------
0003 ------------3456----------------------------------1234-----------------------------------9012-----------------------------------67890----------------------------------------------
0004 ---------------78901-------------------------------5678-------------------------------3456-------------------------------1234-----------------------------------------------
0005 ------------------2345----------------------------90123----------------------------7890----------------------------5678-----------------------------------------------
0006 --------------------6789----------------------------4567----------------------------12345----------------------------9012-----------------------------------------------
0007 ----------------------0123----------------------------8901----------------------------6789----------------------------34567----------------------------------------------
0008 ------------------------45678-------------------------2345-------------------------0123-------------------------8901-----------------------------------------------
0009 --------------------------9012-------------------------67890-------------------------4567-------------------------2345-----------------------------------------------
0010 ----------------------------3456----------------------1234----------------------89012----------------------6789-----------------------------------------------
0011 ------------------------------7890-------------------5678-------------------3456-------------------01234----------------------------------------------
0012 --------------------------------12345----------------9012----------------7890----------------5678-----------------------------------------------
0013 ----------------------------------6789----------------34567----------------1234----------------9012-----------------------------------------------
0014 ------------------------------------0123----------------8901----------------56789----------------3456-----------------------------------------------
0015 --------------------------------------4567----------------2345----------------0123----------------78901----------------------------------------------
```

Not a monospace font.

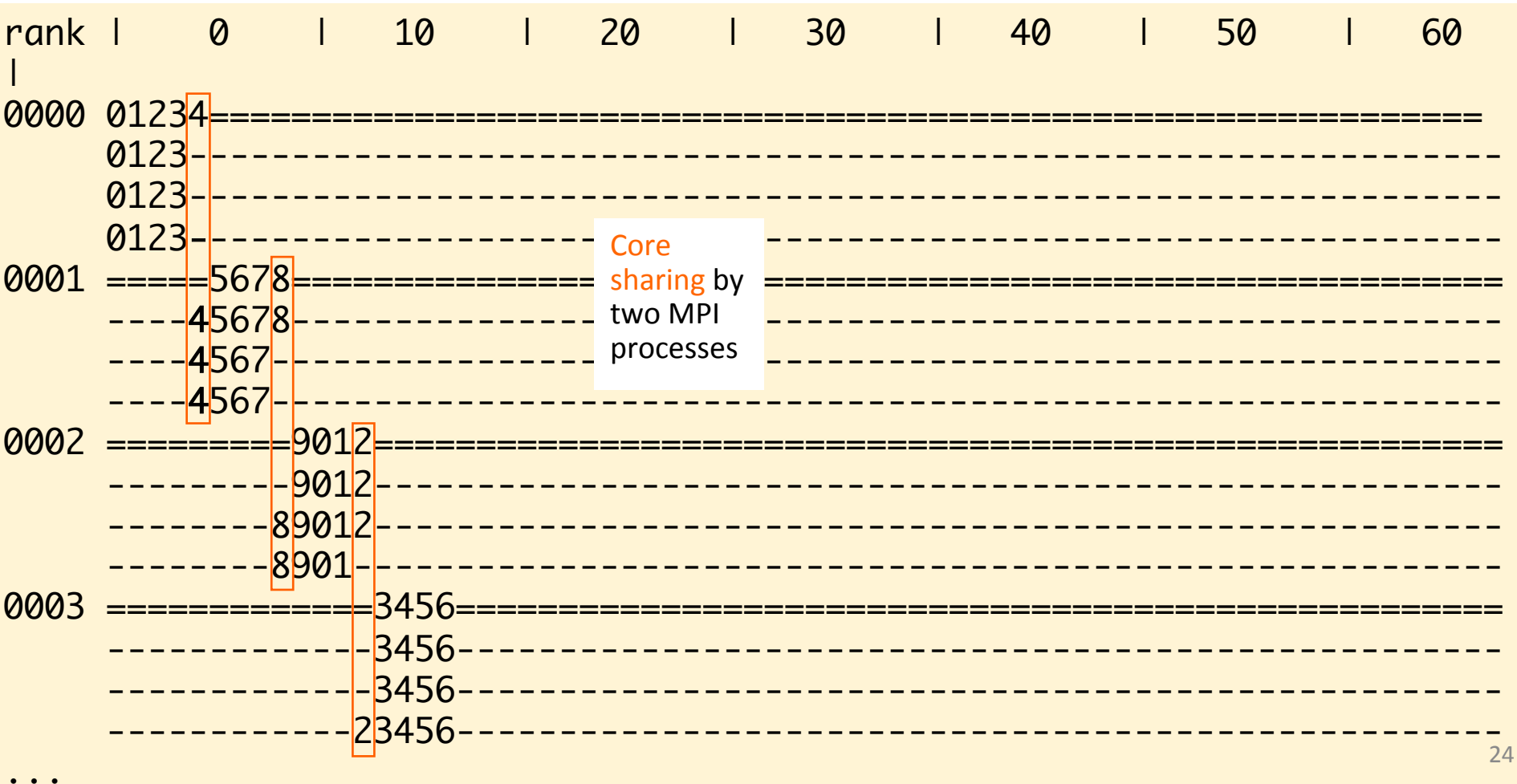
A closer look:



Maybe a core listing would be insightful?

A better view: Cores

```
$ mpirun –np 16 amask_mpi     #for hybrid, export OMP_NUM_THREADS=17
```



Potato (16x17) or Potahto (17x16)

```
$ mpirun -np 17 amask_mpi  # for hybrid, export OMP_NUM_THREADS=16

rank |    0    |   10    |   20    |   30    |   40    |   50    |   60
0000 0123=============================================================
     0123-------------------------------------------------------------
     0123-------------------------------------------------------------
     0123-------------------------------------------------------------
0001 ====4567=========================================================
     ----4567-------------------------------------------------------
     ----4567-------------------------------------------------------
     ----4567-------------------------------------------------------
0002 ========8901=====================================================
     --------8901-----------------------------------------------------
     --------8901-----------------------------------------------------
     --------8901-----------------------------------------------------
...
0016 ================================================================4567
     ----------------------------------------------------------------4567
     ----------------------------------------------------------------4567
     ----------------------------------------------------------------4567
```

NO Core sharing by any MPI processes

Affinity Mask

- The Process Mask
  - Shows the possibilities for a process execution
  - Application developers must be aware of Hardware → processor-id (proc-id) mapping to derive benefits of affinity settings
  - Often defaults are good(?)!

Why AMASK?

AMASK: the Affinity MASK tool.

git clone https://github.com/tacc/amask

Because How Often:

are you not aware of what the default affinity settings are?

do you set affinity and not know for sure if you are getting what you ask for?

Thanks!
Questions?