Intel® Xeon Phi™ "Knights Landing" (KNL) System Software

Clark Snyder, Peter Hill, John Sygulla

CUG 2017 CAFFEINATED COMPUTING

Redmond, Washington May 7-11, 2017

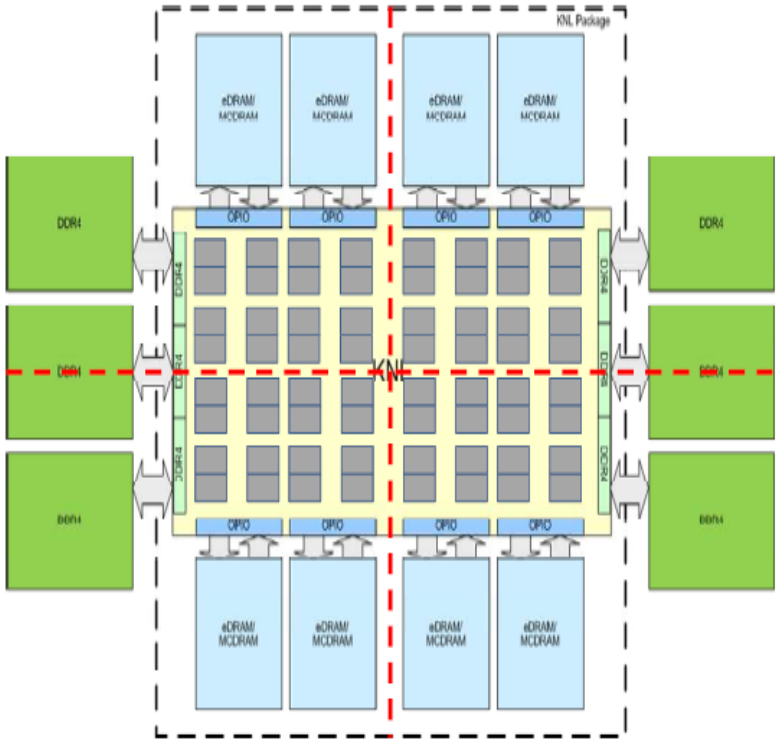
Motivation

- **The Intel® Xeon Phi™ "Knights Landing" (KNL) has 20 different configurations**
  - 5 NUMA modes X 4 memory modes = 20 configurations
- **How do I, as a user or system administrator, manage these options on my Cray® XC™ System?**

Agenda

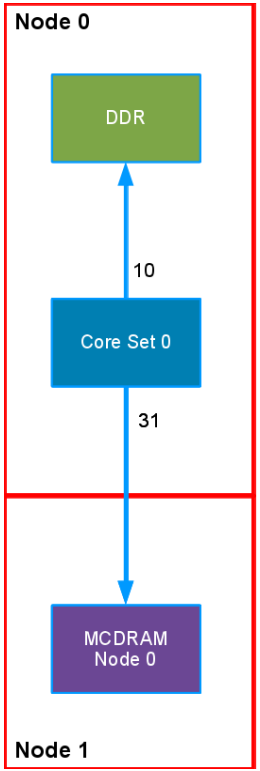
- **As a user, how do I…**
  - Choose the best configuration?
  - Configure the KNLs?
  - Figure out how the KNLs are configured?
  - Use zonesort and what is it?
- **As a system administrator, how do I…**
  - Configure the KNLs?
  - Figure out how the KNLs are configured?
- **Are there costs to reconfiguration?**
- **Summary**
- **Q&A**

KNL Architecture Overview



- **Processing elements**
  - Each tile has 2 cores
  - Each core has 4 threads
- **MCDRAM configuration**
  - Allocates MCDRAM between cache and addressable (flat) memory (4 options)
- **NUMA**
  - Splits tiles, DDR, and flat MCDRAM 1, 2, or 4 ways (5 options)
  - Addressable MCDRAM is always in separate NUMA node(s) from DDR and CPUs

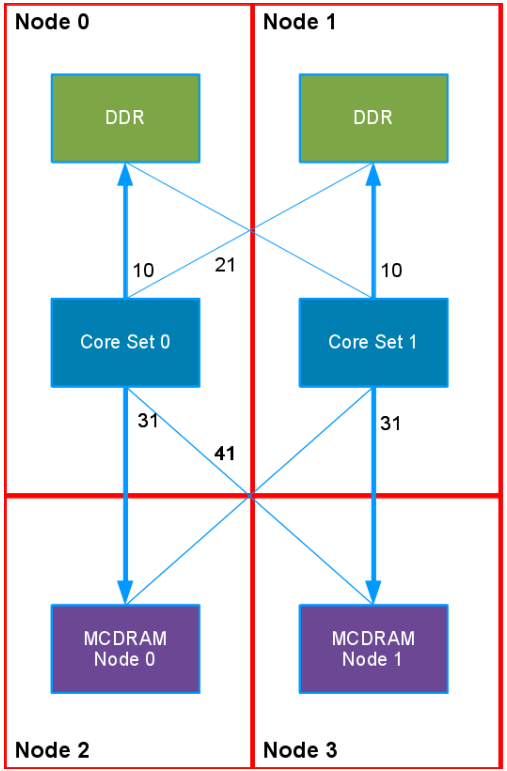
KNL Architecture Overview – a2a, hemi, quad



- **NUMA: all-to-all (a2a), hemisphere (hemi), quadrant (quad)**
  - Change internal data flows
  - Only externally visible difference is performance
- **MCDRAM:**
  - If 100% cache, NUMA node 1 disappears

Note: Relative weights on lines indicate kernel allocation preference where lower numbers are preferred

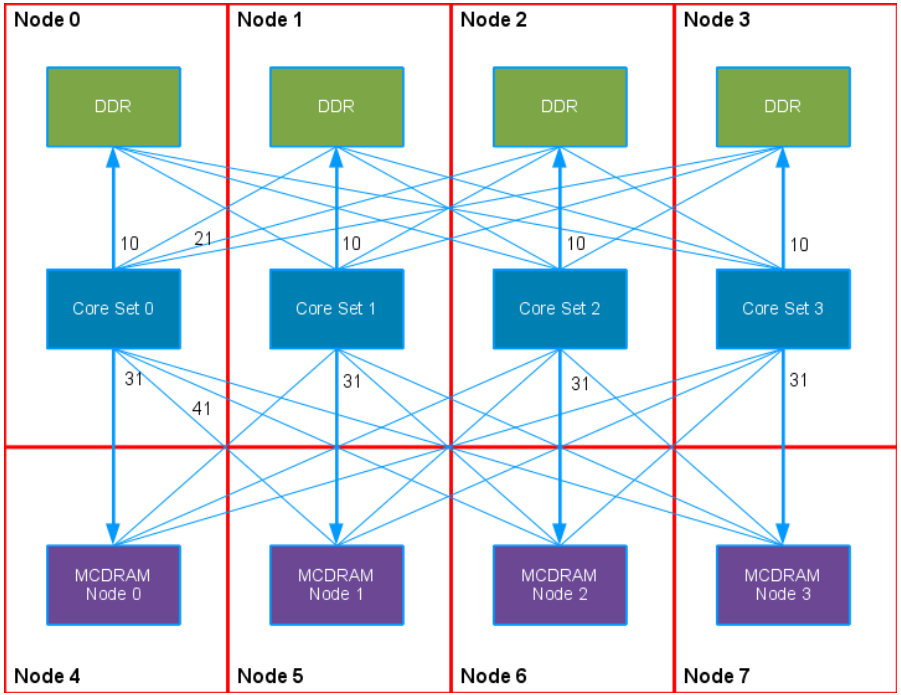
KNL Architecture Overview – snc2



- **NUMA: sub-NUMA cluster 2 (snc2)**
  - Divides DDR and tiles into 2 NUMA nodes
  - Divides flat MCDRAM into 2 NUMA nodes
- **MCDRAM:**
  - If 100% cache, NUMA nodes 2 & 3 disappear

Note: Relative weights on lines indicate kernel allocation preference where lower numbers are preferred

KNL Architecture Overview – snc4



- **NUMA: sub-NUMA cluster 4 (snc4)**
  - Divides DDR and tiles into 4 NUMA nodes
  - Divides flat MCDRAM into 4 NUMA nodes
- **MCDRAM:**
  - If 100% cache, NUMA nodes 4-7 disappear

Note: Relative weights on lines indicate kernel allocation preference where lower numbers are preferred

Which configuration should I use?

- **Job placement is harder in SNC modes**
  - Flat MCDRAM compounds the difficulty
  - SNC4 on 7250 results in unequal tile/core counts per NUMA node

SNC4 Tiles per NUMA Node (NN)

KNL SKU	Total Tiles	NN 0 Tiles	NN 1 Tiles	NN 2 Tiles	NN 3 Tiles
7210	32	8	8	8	8
7230	32	8	8	8	8
7250	34	9	9	8	8

- **Easiest configuration to use is quad/cache**
  - It performs well for most codes with the least fussing
  - No issue with uneven numbers of cores per NUMA node
  - No need to force memory allocations into flat MCDRAM

For more performance information, refer to the CUG tutorial "Getting the Most Out of Knights Landing" by John Levesque

How can a user configure a KNL?

- **Use the workload manager (WLM) to request a configuration for your job**
  - The WLM will match your request to pre-configured nodes; and/or
  - Reconfigure nodes to meet your request
- **Examples:**

Moab	**$ msub -l os=CLE_quad_flat run_script**
PBS	**$ qsub -l aoe=quad_0 run_script**
Slurm	**$ sbatch -C quad,flat run_script**

- **Upcoming CLE6.0UP04 feature: report node reconfiguration state as "rebootq", rather than "down", in xtnodestat, xtprocadmin and apstat**

How is the KNL currently configured?

- ALPS ‘apstat –M’ (from login node)

```
$ apstat -M
NID  Memory(MB)  HBM(MB)  Cache(MB)  NumaCfg
 24      114688    16384          0     quad
 25      106496    16384       8192     quad
```

- Slurm ‘sinfo’ (from login node)

```
$ sinfo -o "%N %f" NODELIST AVAIL_FEATURES nid00[008-047,052-063,072-115,120-127,140-191]
flat,split,equal,cache,a2a,snc2,snc4,hemi,quad
$ sinfo -o "%N %b" NODELIST ACTIVE_FEATURES nid00[008-047,052-063,072-115,120-127,140-166]
quad,cache nid00[188-191] cache,quad nid00[167-187] quad,flat
```

- cnselect (from login node)

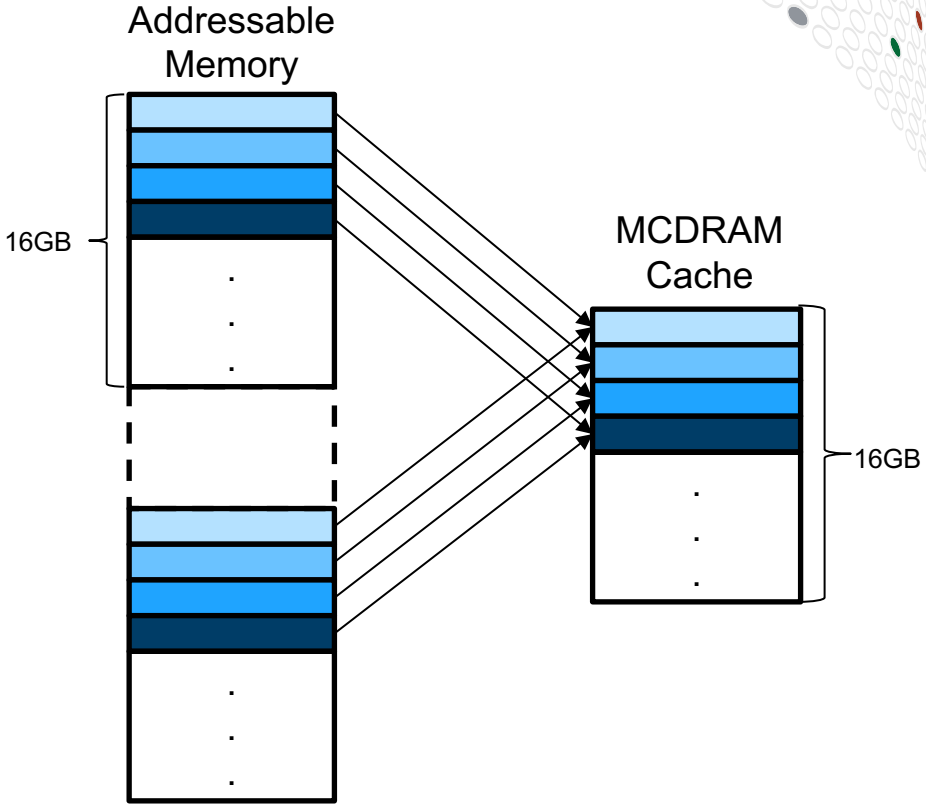
```
$ cnselect hbmcachepct.eq.100.and.numa_cfg.eq.quad
24-27,56-59,68-75,92-94,144-147,160-179
```

- hwloc: lstopo (from compute node)

```
$ aprun -qL 58 lstopo-no-graphics
Machine (94GB total) + NUMANode L#0 (P#0 94GB) + Package L#0 + L3(MemorySideCache) L#0 (16GB)
…
```

What is zonesort and how do I use it?

- **Issue**
  - MCDRAM cache is a physically addressed, direct-mapped
    - (Physical address) modulo (cache size) = (cache address)
    - (RAM size) module (cache size) = (number of conflicting addresses)
  - As memory is allocated and freed, the actual physical memory that is free changes as does the order in which this memory is placed on the free list
  - If two hot memory addresses vie for the same cache line:
    - Cache evictions go up and performance goes down
  - Performance may vary significantly from run to run
- **How zonesort helps**
  - Sorts memory on the free list by physical address
    - Improves run-time consistency by putting free memory in a consistent order
  - Invoked automatically by ALPS and Slurm
  - Supported by SchedMD in Slurm 17.02 release
  - Upcoming CLE 6.0UP04 ALPS feature to periodically invoke zonesort during an application run



How can an administrator configure a KNL?

- **capmc set_mcdram_cfg | set_numa_cfg**
  - But, the configuration won't take effect until the next reboot
  - The capmc node_reinit command will bounce and boot nodes

```
crayadm@smw:~> capmc set_numa_cfg -n 59 -m quad -p
crayadm@smw:~> capmc set_mcdram_cfg -n 59 -m flat -p
crayadm@smw:~> capmc node_reinit -n 59
{
    "e":0,
    "err_msg":"Success"
}
```

How will the KNL be configured next?

- **capmc get_mcdram_cfg | get_numa_cfg (SMW | login)**
  - But remember capmc shows settings to use during next boot, which *may or may not* match the current configuration

```
crayadm@smw:~> capmc get_mcdram_cfg -pn 24,128
NID            | MCDRAM Mode  | DRAM Size  | MCDRAM Size
===============|==============|============|============
24             | cache/100    | 96GB       | 16384MB
128            | flat/0       | 96GB       | 16384MB
Success
crayadm@smw:~> capmc get_numa_cfg -pn 24,128
NID            | NUMA Mode
===============|===========
24             | quad
128            | a2a
Success
```

How is the KNL currently configured?

- xthwinv

```
crayadm@smw:~> xthwinv -x c0-0c0s14
...
        <mcdram_memory size="16384" units="MB">
            <count>8</count>
            <max_speed>7.2</max_speed>
            <cfg_total>16384</cfg_total>
            <cfg_cache>16384</cfg_cache>
            <cfg_flat>0</cfg_flat>
            <mcdram_cfg>cache</mcdram_cfg>
        </mcdram_memory>
        <numa_cfg>quad</numa_cfg>
...
```

How was the KNL configured?

- **Console log on the SMW (/var/opt/cray/log/p0-current/console-<date>)**
  - BIOS messages show configuration information at each boot

```
2016-04-08T07:03:07.486640-05:00 c0-0c0s14n0  BUS_STATUS: DDR4 memSpeed          = 0x0960
2016-04-08T07:03:07.486659-05:00 c0-0c0s14n0              MCDRAM Active count   = 0xFF
2016-04-08T07:03:07.486676-05:00 c0-0c0s14n0              MCDRAM speed          = 0x48
2016-04-08T07:03:07.486694-05:00 c0-0c0s14n0              MCDRAM totalMem       = 0x00000100
2016-04-08T07:03:07.486712-05:00 c0-0c0s14n0              MCDRAM totalCache     = 0x00000100
2016-04-08T07:03:07.486731-05:00 c0-0c0s14n0              MCDRAM totalFlat      = 0x00000000
2016-04-08T07:03:07.486748-05:00 c0-0c0s14n0              MCDRAM memoryModel    = 0x04
2016-04-08T07:03:07.486770-05:00 c0-0c0s14n0              MCDRAM memoryMode     = 0x00
2016-04-08T07:03:07.486781-05:00 c0-0c0s14n0              MCDRAM totalClusters  = 0x04
2016-04-08T07:03:07.486793-05:00 c0-0c0s14n0              MCDRAM cacheRatio     = 0x04
```

How was the KNL configured? (BIOS decoder)

MCDRAM memoryModel	capmc NUMA config
0x00	a2a
0x01	snc2
0x02	snc4
0x03	hemi
0x04	quad

MCDRAM memoryMode	MCDRAM cacheRatio	capmc MCDRAM config	cache % of MCDRAM
0x00	0x04	cache/100	100%
0x01	0x00	flat/0	0%
0x02	0x01	split/25	25%
0x02	0x02	equal/50	50%

How was the KNL configured?

- **SMW commands log (/var/opt/cray/log/commands/log.<date>)**
  - Tracks the subcommands issued, e.g. xtbounce, xtcli boot
- **xtremoted log (/var/opt/cray/log/xtremoted-<date>)**
  - Captures the capmc operations

```
<157>1 2017-04-18T08:28:58.633589-05:00 smw xtremoted 48520 - [hss_xtremoted@34] auth_cb:
Remote IP (172.30.49.161) URI(/capmc/set_mcdram_cfg) request is  authorized
<157>1 2017-04-18T08:28:59.007533-05:00 smw xtremoted_dbutil 55110 - [hss_xtremoted@34]
do_key_value_edit: Setting `mcdram_cfg=cache` for nids
[128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,160,161,16
2,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,56,57,58,59,92,93,94,
95,24,25,26,27,68,69,70,71,72,73,74,75]
```

Are there downsides to reconfiguration?

- **More choices…**
  - For users and administrators
- **Reconfiguration requires rebooting the compute node(s), which…**
  - Is not 100% reliable
  - Takes time

See also the CUG presentation "CLE 6 Boot Performance and Reliability", by Joel Landsteiner, which is part of the tutorial "Migrating, Managing, and Booting Cray XC and CMC/eLogin…"

Boot and reconfiguration times at scale

Argonne Theta • 20-cabinet Cray XC40 system • 3,624 Xeon Phi 7230 compute nodes • CLE 6.0/8.0.UP03 • 27 March 2017

System Boot, time in seconds [1]

Archive[2]	*Bounce*	*Boot and SDB*	*Fanout Service*	*Wait Service*	*Fanout Compute*	*Wait Compute*	*Other[2]*	*Total*	
44	738	408	52	346	126	729	65	2508	
								41'48"	

KNL Mode Reconfiguration, time in seconds [3]

Shutdown	*Bounce*	*Fanout Compute*	*Wait Compute*	*Total*	
26	200	110	745	1081	
25	166	109	769	1069	
26	165	108	728	1027	
26	*177*	*109*	*747*	*1059*	*Average*
				17'39"	

[1] Includes bounce (hardware initialization, with Aries "linktune") and boot of 3,740 service and compute nodes
[2] *Archive* processing, and *Other* xtbootsys overhead, do not include 487 seconds spent waiting for human input
[3] Includes shutdown, bounce (hardware initialization, excluding Aries) and boot of 3,624 KNL compute nodes

Summary

- **KNL configurability brings**
  - Choice of modes to use
  - New commands and options for configuring KNLs and monitoring KNL configurations
  - Trade-offs for configuration time vs. execution time

Legal Disclaimer

Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.

Cray Inc. may make changes to specifications and product descriptions at any time, without notice.

All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Cray hardware and software products may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Cray uses codenames internally to identify products that are in development and not yet publically announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.

Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, and URIKA. The following are trademarks of Cray Inc.: APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHLEX, LIBSCI, NODEKARE, REVEAL, THREADSTORM. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.

Clark Snyder
csnyder@cray.com

CUG.2017.CAFFEINATED COMPUTING

Redmond, Washington May 7-11, 2017