Cray XC50 CLE Port to ARM
CUG 2018

Jeff Schutkoske, Cray Inc.

Agenda

- Purpose
- Functionality
- Lessons Learned
- Performance
- Summary
- Q&A

Purpose

The port of the Cray Linux Environment (CLE) to the XC50 ARM system:

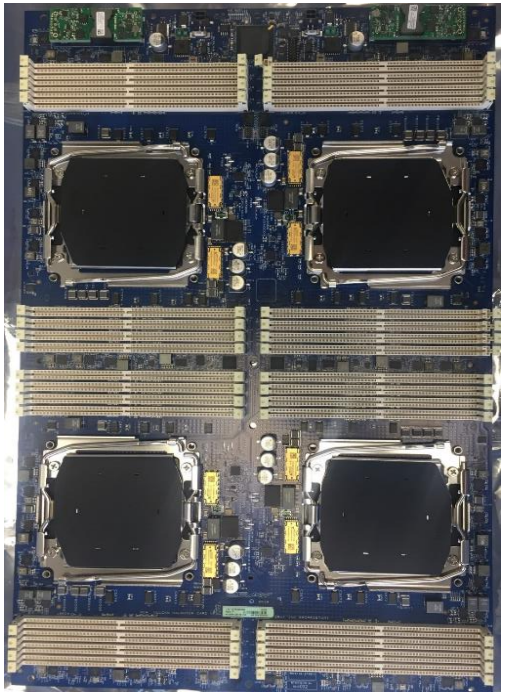
1. **Describe the base functionality**
2. **Share lessons learned**
3. **Provide initial performance**

Functionality

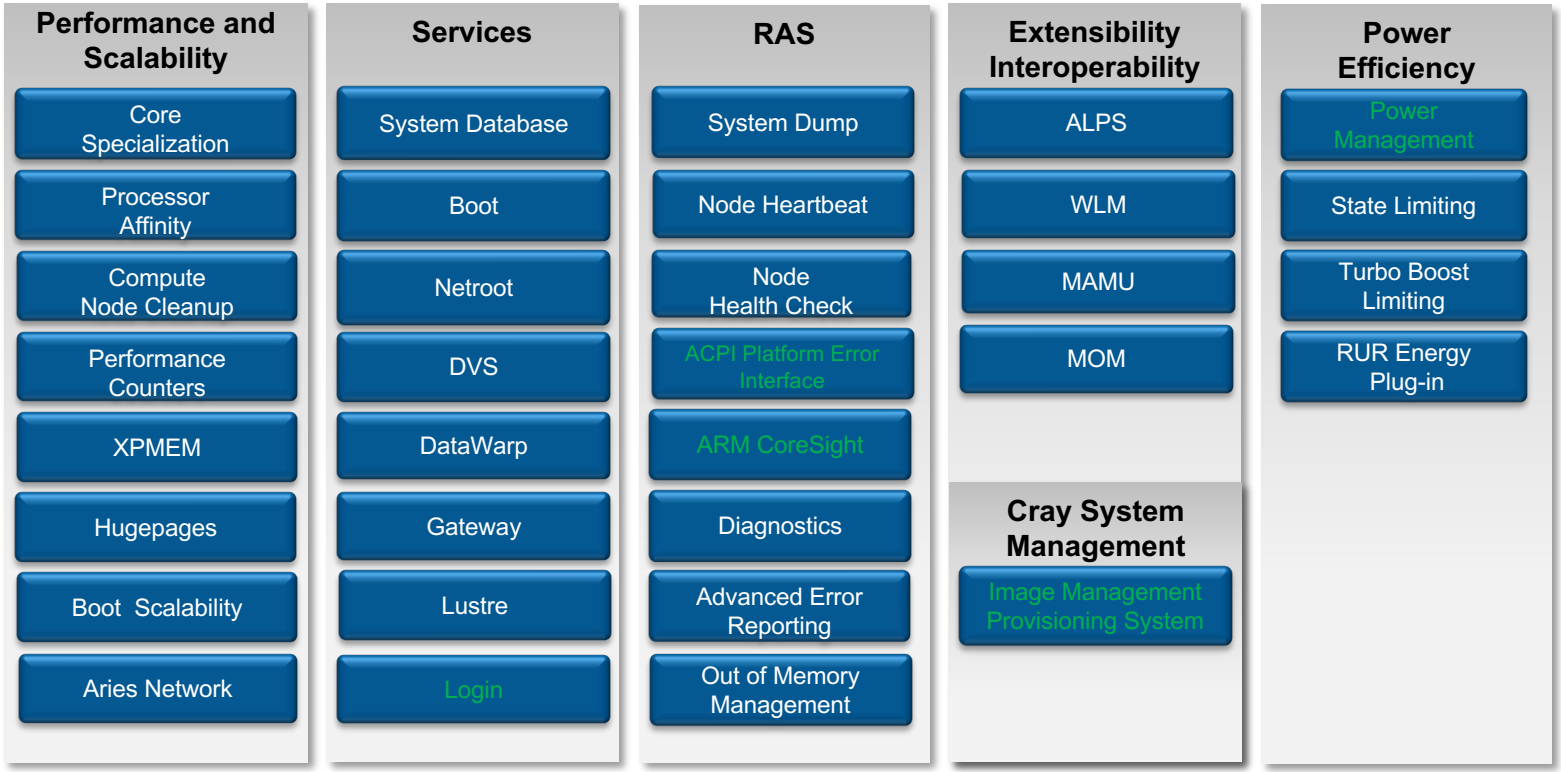
Cray XC50 ARM System Is Here

Cray XC50 ARM System Basics

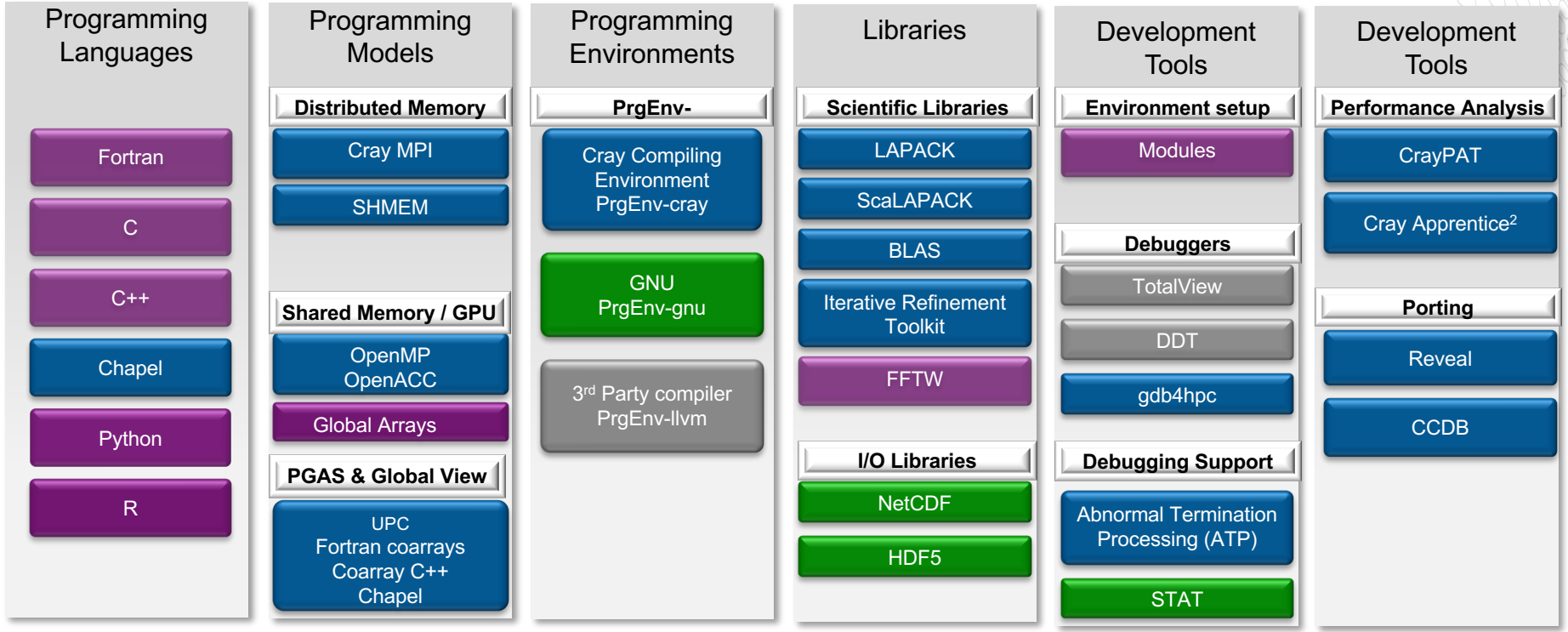
- Dual socket node
- 32 cores per socket
- 8 DIMMs per socket
- 2.1 GHz base frequency + boost
- Inter-Chip Interconnect 75GB/s/dir
- CLE 6.0 UP07 GA
- SLES 12 SP3



Cray Linux Environment (CLE)



Cray Programming Environment (Cray PE)



Cray Developed
Cray added value to 3rd party
3rd party packaging
Licensed ISV SW

COMPUTE | STORE | ANALYZE

Lessons Learned

ARM vs x86 Data Layout - Fortran

CCE Fortran	Kind(1)	Kind(2)	Kind(4) default	Kind(8)	Kind(16)	x86 ↔Arm Compatible ?
logical	8 bits	16 bits	32 bits	64 bits	-	✓ Yes
integer	8 bits	16 bits	32 bits	64 bits	-	✓ Yes
real	-	-	32 bits	64 bits	128 bits	✓ Yes
complex	-	-	64 bits	128 bits	256 bits	✓ Yes

ARM vs x86 Data Layout – C and C++

C and C++	64-bit x86 ABI	CCE 64-bit x86	64-bit Arm (Cray and ABI)	Compatible?
bool	8 bits	8 bits	8 bits	✓ Yes
char	8 bits	8 bits	8 bits	✓ *Partial*
short	16 bits	16 bits	16 bits	✓ Yes
int	32 bits	32 bits	32 bits	✓ Yes
long	64 bits	64 bits	64 bits	✓ Yes
long long	64 bits	64 bits	64 bits	✓ Yes
float	32 bits IEEE	32 bits IEEE	32 bits IEEE	✓ Yes
double	64 bits IEEE	64 bits IEEE	64 bits IEEE	✓ Yes
long double	80 bits non-IEEE	64 bits IEEE	128 bits IEEE	× No

ARM versus x86 – C and C++ *char* Defaults

- **The C and C++ *char* type has different defaults**
  - x86: *char* defaults to *signed char*
  - ARM: *char* defaults to *unsigned char*

- **Though the difference seems minor, it has proven to be a significant issue when porting from x86 to ARM**
  - The command line option "***–h signedchars***" can be used to force ARM to the x86 behavior

ARMv8 Processor Quirks

- ARMv8 processor is less forgiving on alignment even with checking disabled
- Some applications used Intel's timing instruction directly through an *ASM()* intrinsic
- ARMv8 is not sequentially (or processor) consistent while x86 is

ARMv8 Port

- Applications written to be portable require very little to no changes
- Recompile and run in many cases

Performance

Cray Compiling Environment (CCE)

- **Compared 158 codes from relevant benchmarks**
  - 76 C codes
  - 11 C++ codes
  - 71 Fortran codes
- **Cray CCE compared against latest LLVM and gcc**
  - Over 67% of the benchmarks executed faster using CCE when compared to LLVM (10% were on par)
  - Over 60% of the benchmarks executed faster when compared to gcc (13% were on par)

Early Results

- Dual socket, 32 cores per socket, 8 DIMMs per socket
- Increased memory bandwidth
- Increased core counts (64 per node)
- High processor frequency (2.1 GHz)

Native ARM Login Nodes

```
jjs@desktop:~> ssh crayadm@login-aarch
Last login: Mon Apr 23 10:35:28 2018 from 172.30.55.82

    Running 67MB Suse 12.3 image slurm_login-large_17.11.4_cle_6.0up07_sles_12sp3
    CLE release 6.0.UP07, build 6.0.7011(201803112300)
    0 vcores, boot_freemem: 256155mb

crayadm@login-aarch:~> uname -a
Linux login-aarch 4.4.103-6.38_4.0.103-cray_ari_s #1 SMP Mon Mar 5 22:40:16 UTC 2018 (4.0.103) aarch64
aarch64 aarch64

crayadm@login-aarch:~> cat /proc/cpuinfo |tail -n 10

processor       : 255
BogoMIPS        : 400.00
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics
CPU implementer : 0x43
CPU architecture: 8
CPU variant     : 0x1
CPU part        : 0x0af
CPU revision    : 0
```

Hello World SHMEM and UPC

```
crayadm@login-aarch:~> cc shmem-hello.c -o shmem-hello
crayadm@login-aarch:~> srun -N 4 ./shmem-hello
Hello from thread 1 on CPU 0 on host nid00039
Hello from thread 2 on CPU 0 on host nid00040
Hello from thread 3 on CPU 0 on host nid00041
Hello from thread 0 on CPU 0 on host nid00038
```

```
crayadm@login-aarch:~> cc -h upc hello.upc -o hello-upc
crayadm@login-aarch:~> srun -N 4 ./hello-upc
Hello from thread 1 on host nid00039
Hello from thread 3 on host nid00041
Hello from thread 2 on host nid00040
Hello from thread 0 on host nid00038
```

Netstress SHMEM

```
crayadm@login-aarch:~/netstress-2.4.0> make shmem
cc -DCRAY -O3 -D_SHMEM -DNETSTRESS_VERSION=\"2.4.0\"  -o ./netstress-shmem src/*.c -lm
src/netstress_utils.c:
src/rand64.c:
src/timer.c:

crayadm@login-aarch:~/netstress-2.4.0> export XT_SYMMETRIC_HEAP_SIZE=1024M
crayadm@login-aarch:~/netstress-2.4.0> srun -N 4 ./netstress-shmem –m 8K –M 64K
<snipped table for brevity>

 Group        Size        Max Time    Min B/W     Avg B/W     Max B/W
                          (sec)       (MB/s)      (MB/s)      (MB/s)
  all:        8 KB        0.0027      690.09      693.73      696.51
  all:        16 KB       0.0013      2811.20     2843.08     2895.55
  all:        32 KB       0.0023      3224.40     3305.90     3446.62
  all:        64 KB       0.0044      3399.13     3499.82     3648.70
# Total run time: 0.01s
```

Netstress UPC

```
crayadm@login-aarch:~/netstress-2.4.0>  make upc
cc -D_UPC -DNETSTRESS_VERSION=\"2.4.0\"  -o ./netstress-upc src/*.c -h upc -h omp -DCRAY -O3 -lm
src/netstress.c:
src/netstress_utils.c:
src/rand64.c:
src/timer.c:

crayadm@login-aarch:~/netstress-2.4.0> export XT_SYMMETRIC_HEAP_SIZE=1024M
crayadm@login-aarch:~/netstress-2.4.0> srun -N 4 ./netstress-upc -m 8K -M 64K
<snipped table for brevity>

 Group          Size          Max Time      Min B/W       Avg B/W       Max B/W
                              (sec)         (MB/s)        (MB/s)        (MB/s)
  all:          8 KB          0.0009        1998.56       2016.62       2029.50
  all:          16 KB         0.0014        2763.29       2769.00       2785.80
  all:          32 KB         0.0024        3187.48       3228.21       3272.03
  all:          64 KB         0.0043        3466.56       3524.43       3611.01
# Total run time: 0.02s
```

Summary

- **Cray XC50 systems with ARMv8 available**
  - Dual socket, 32 cores per socket, 8 DIMMs per socket
- **CLE available**
  - Equivalent system functionality, performance, and scalability
- **Cray PE available**
  - Compilers, programming environments, and developer tools

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 publicly 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, URIKA and YARCDATA. The following are trademarks of Cray Inc.: CHAPEL, CLUSTER CONNECT, CLUSTERSTOR, CRAYDOC, CRAYPAT, CRAYPORT, DATAWARP, ECOPHLEX, LIBSCI, NODEKARE, REVEAL. 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 on this website are the property of their respective owners.


Q&A
Jeffrey Schutkoske
jjs@cray.com