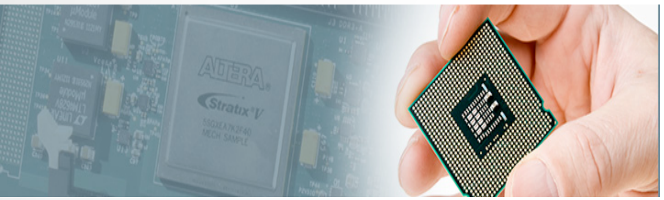
OpenMP for Heterogeneous Multicore Embedded Systems using MCA API standard interface

Sunita Chandrasekaran (sunita@cs.uh.edu)
Peng Sun, Suyang zhu, Barbara Chapman
HPCTools Group, University of Houston, USA

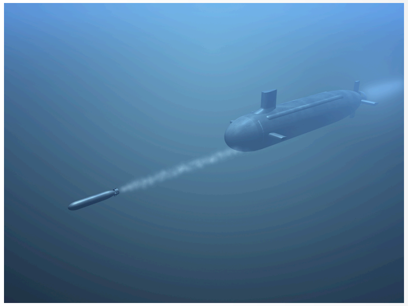
in collaboration with freescale semiconductor (FSL) and Semiconductor Research Corporation (SRC)

OpenMP Booth @ SC14

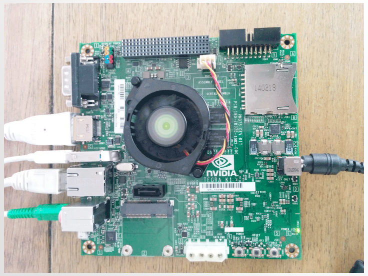
Real-world applications using embedded systems



FPGAS used to receive a number of acoustic pings form an image



DSP, a device to interact with a user through the embedded processor within



A self-piloted car powered by the NVIDIA Jetson TK1

Embedded programmers' requirements

- Write once, reuse anywhere
  - Avoid rewriting from scratch
  - Provide incremental migration path essential for application codes
  - Exploit multiple levels of parallelism
- …with familiar and/or commodity programming models
  - None are perfect, but industry adoption is critical

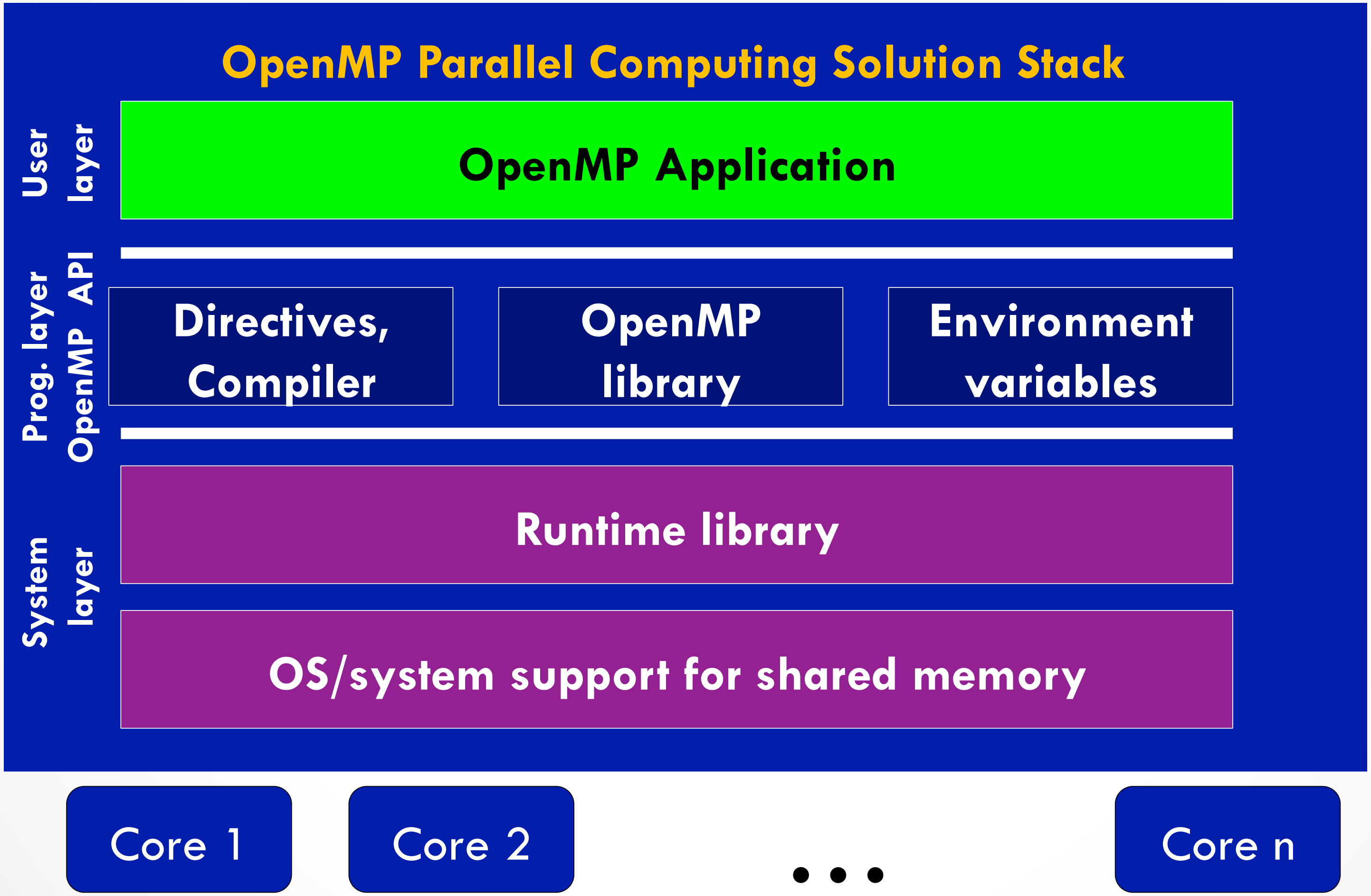
OpenMP – widely adopted standard (www.openmp.org)

- Industry standard for shared memory parallel programming
  - Supported by large # of vendors (TI, AMD, IBM, Intel, Cray, NVIDIA, HP…..)
  - OpenMP 4.0 – provides support for compute devices such as Xeon Phis, GPUs and others
- High level directive-based solution
  - Compiler directives, library routines and envt. variable
  - Programming Model - Fork-Join parallelism
  - Threads communicate by sharing variables
  - Synchronizations to control race conditions
  - Structured programming to reduce likelihood of bugs

```
void main()
{
  double Res[1000];
  #pragma omp parallel f
  for(int i=0;i<1000;i++) {
    do_huge_comp(Res[i]
  }
}
```

Programming made easy!

OpenMP Solution Stack

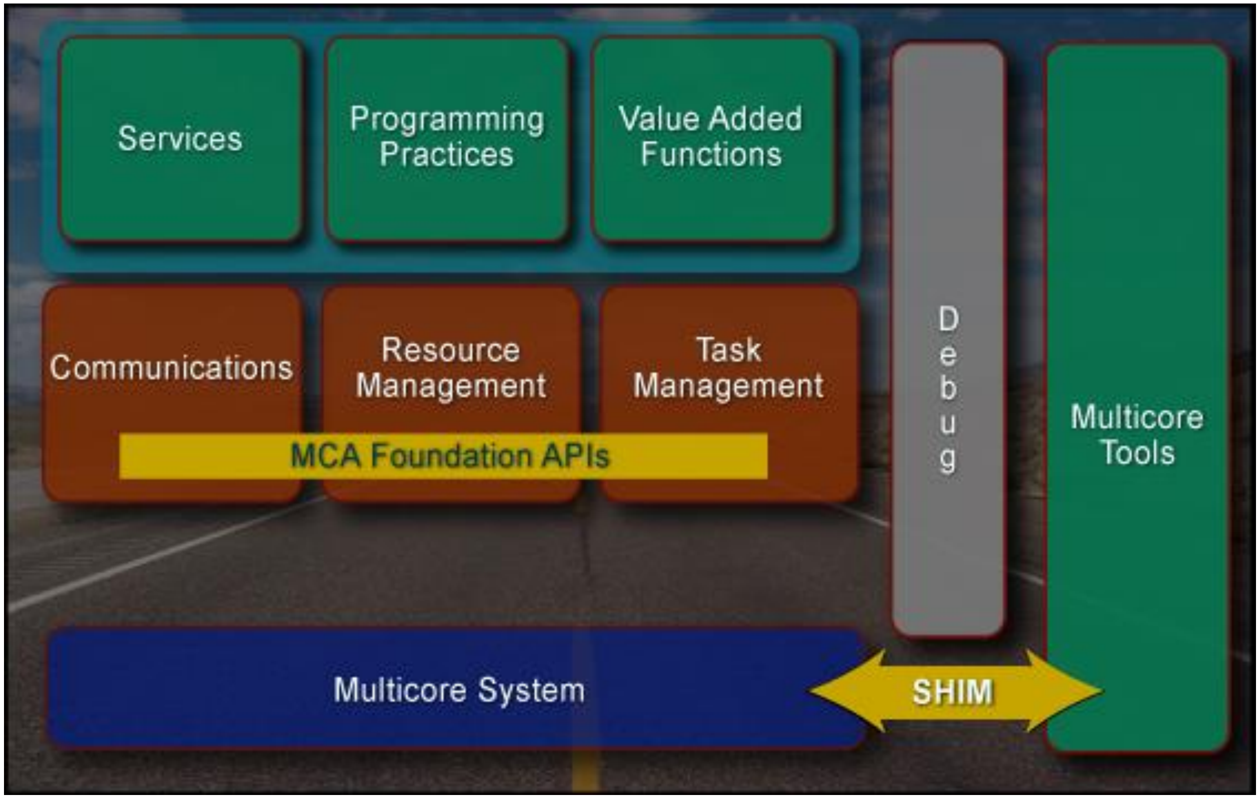


OpenMP for Embedded Systems

- Programming embedded systems a challenge
- Need high-level standards such as OpenMP
  - Runtime relies on lower level components
    - OS, threading/hardware libraries, memory allocation, synchronization e.g. Linux, Pthreads
      - But embedded systems typically lack some of these features
  - OpenMP has shared-memory cache-coherent memory model
    - But embedded platforms feature distributed, non-uniform memory, with no cache-coherency
- Vocabulary for heterogeneity is required in the embedded space

OpenMP 4.0 is there!!

Multicore Association Industry standard API (MCA)



MCA Foundation APIs

Communications (MCAPI)
- Lightweight messaging

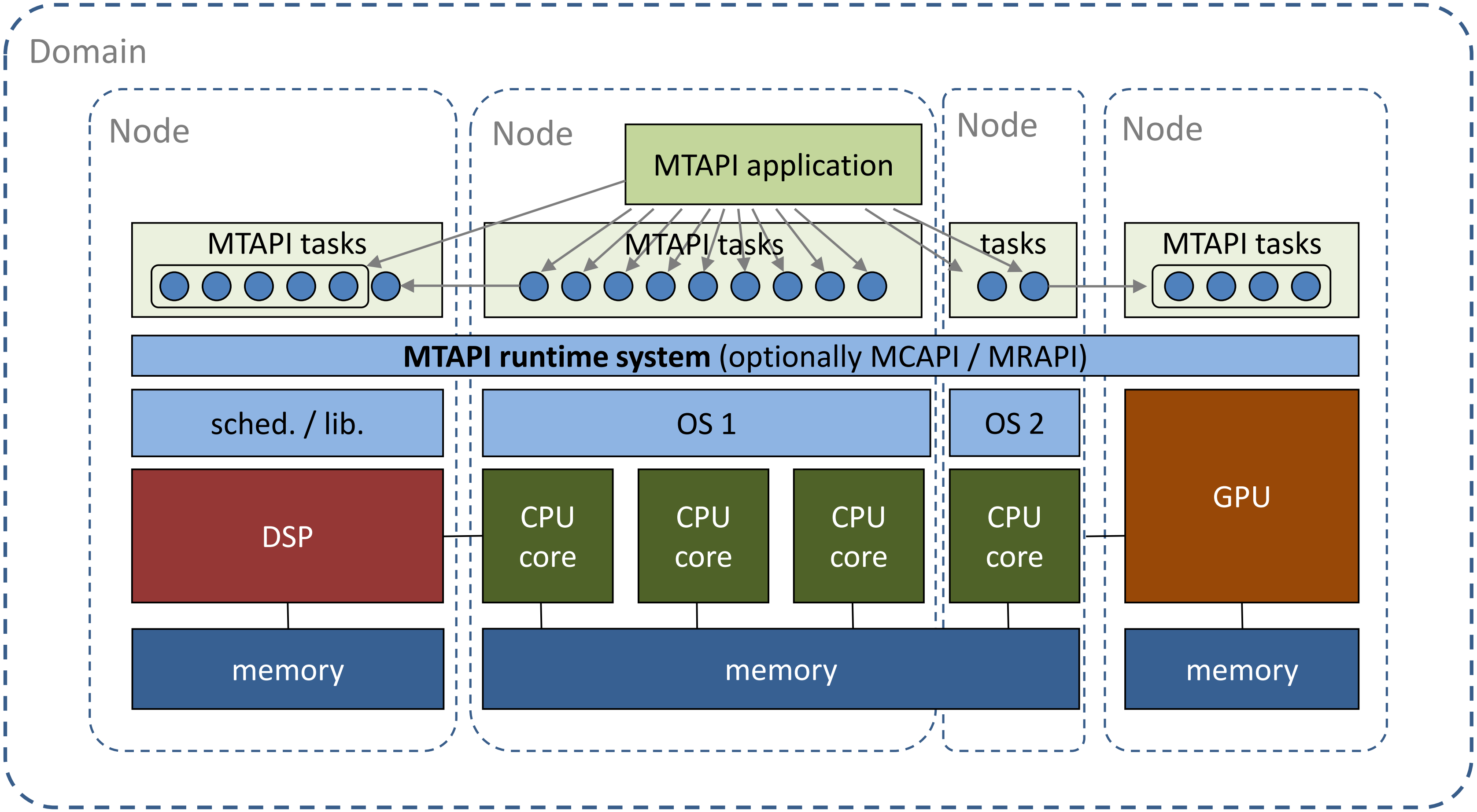
Resource Management (MRAPI)
- Basic synchronization
- Shared/Distributed Memory
- System Metadata

Task Management (MTAPI)
- Task lifecycle
- Task placement
- Task priority

SW/HW Interface for Multicore/Manycore (SHIM)
-XML HW description from SW perspective

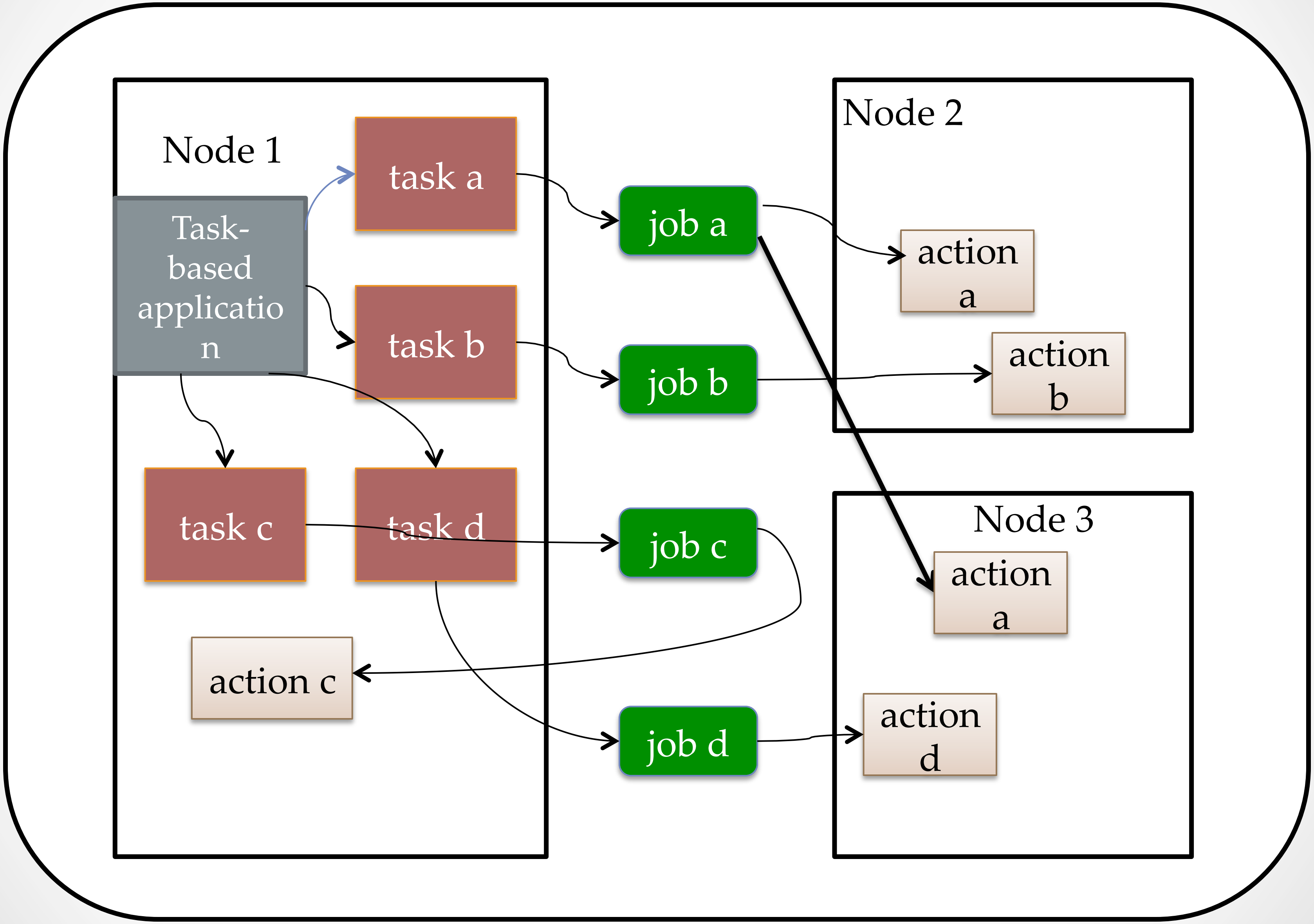
THE Multicore ASSOCIATION ™

Tasks in Heterogeneous Systems



- Tasks execute a job, implemented by an action function, on a local or remote node
- Task can be started individually or via queues (to influence the scheduling behavior)
- Arguments are copied to the remote node
- Results are copied back to the node that started the task

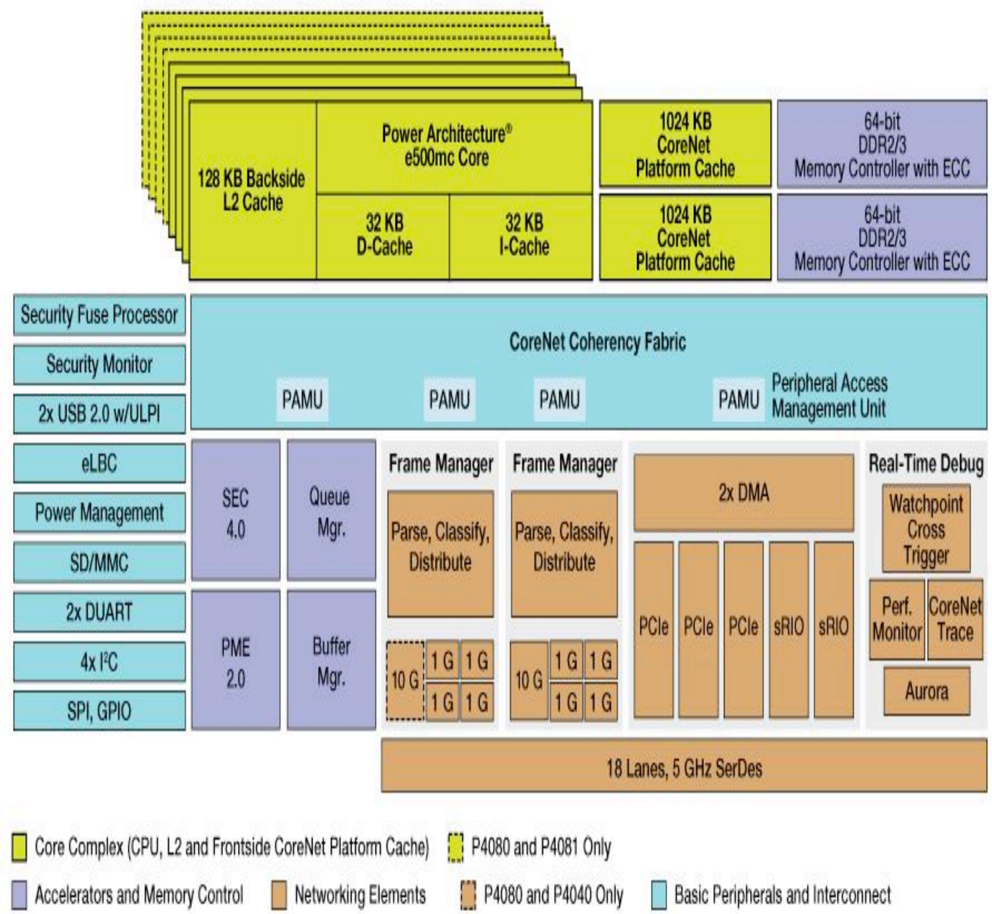
Tasks, Jobs, Actions



Runtime Design and Optimizations

- Optimized Thread Creation, Waiting and Awakening
  - All threads in a team cannot be identical
  - Uses MRAPI meta data primitives
  - Avoid over-subscription
  - Distributed spin-waiting
- Synchronization Construct
- Memory Model
  - Uses MRAPI shared/remote memory primitives

Freescale's Communication processor with data path



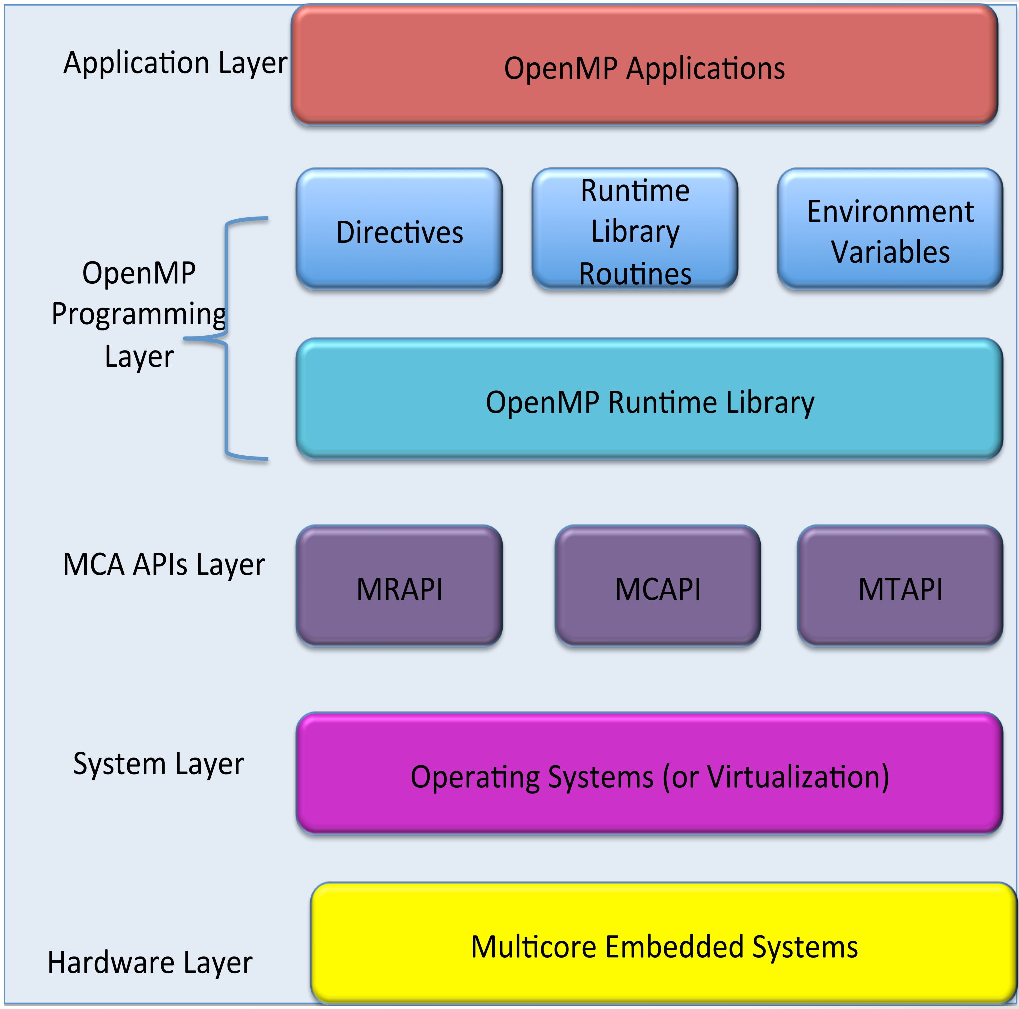
QorIQ P4080 processor

- 4-8 Power architecture e500mc cores
- Accelerators
  - Encryption (SEC)
  - Pattern Matching Engine (PME)
- Target applications:
  - Aerospace and Defense
  - Ethernet Switch, Router
  - Pre-crash detection
  - Forward Collision Warning

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=P4080&tid=redP4040

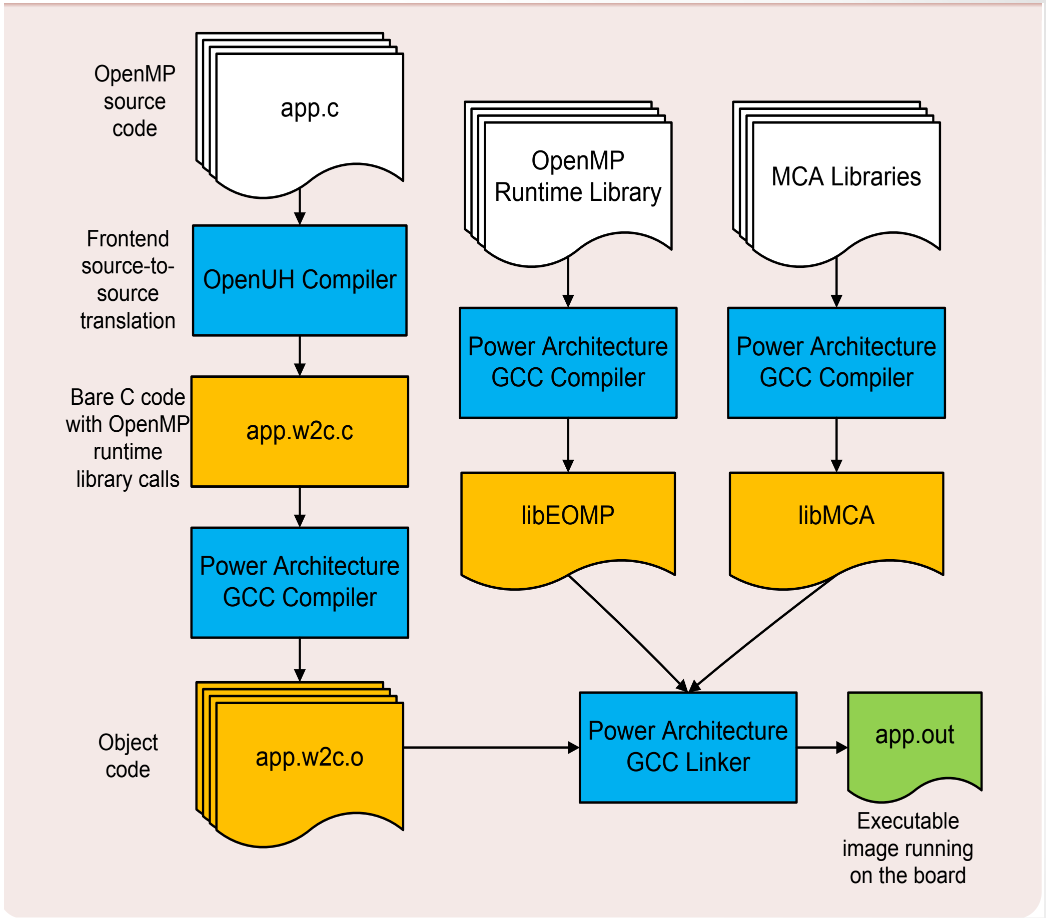
Portable OpenMP Implementation

- Translated OpenMP for MPSoCs
- Used Multicore Association (MCA) APIs as target for our OpenMP translation
- Developed MCA-based runtime:
  - Portable across MPSoCs
  - Light-weight
  - Supports non-cache-coherent systems
  - Performance comparable to customized vendor-specific implementations



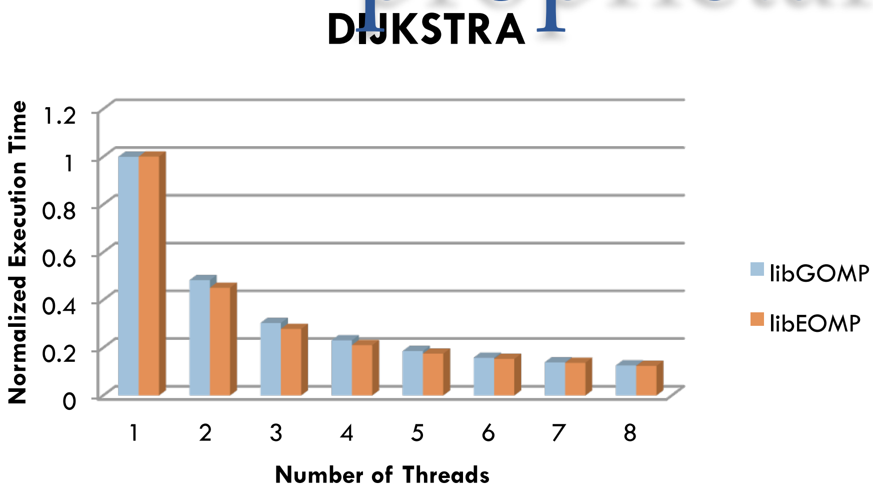
Compilation Process

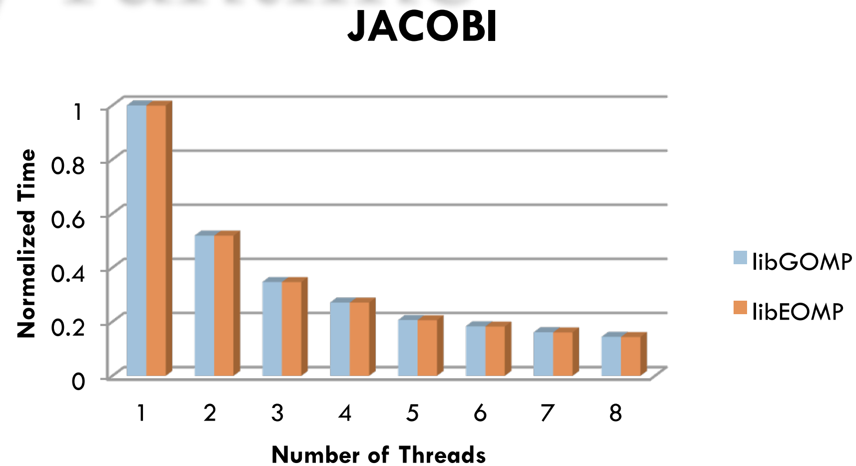
- OpenUH as our frontend open source, optimizing compiler suite for C, C++ and Fortran, based on Open64
  - Translates C+OpenMP source into C with OpenMP runtime function calls
- PowerPC-GCC as our backend to generate the object file and libraries
- Final executable file is generated by linking the object file, our OpenMP runtime library and the MCA runtime library.

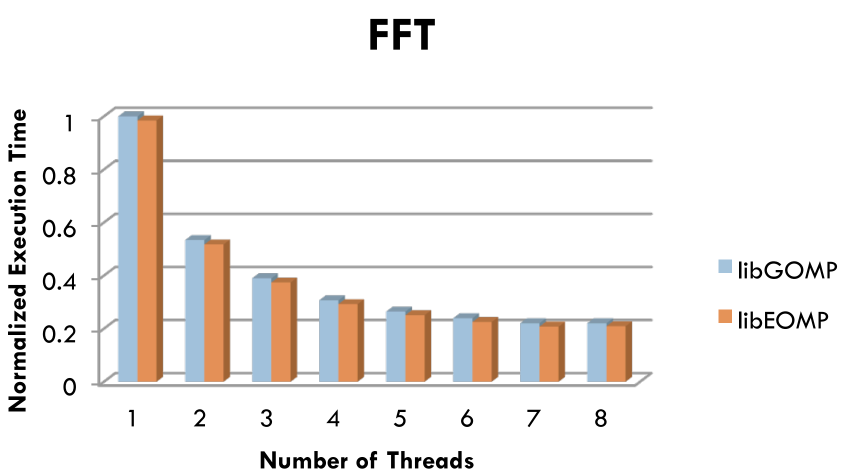


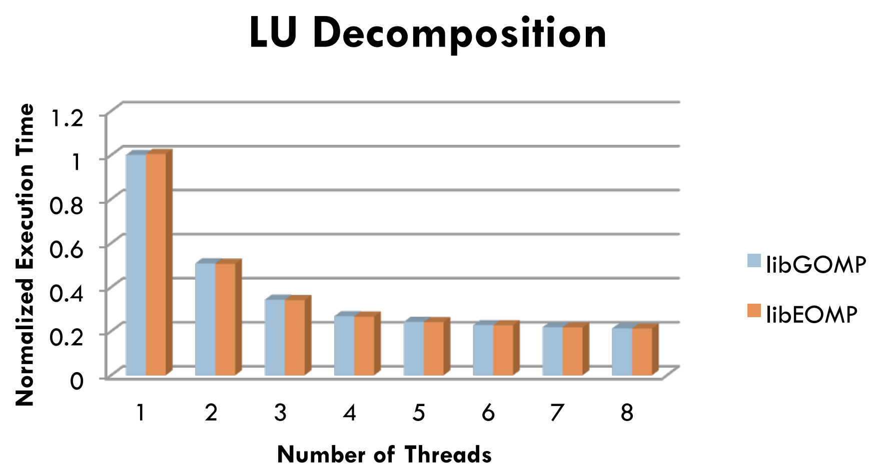
Dual-core power processor from Freescale Semiconductor

Enhanced OpenMP runtime Vs proprietary runtime

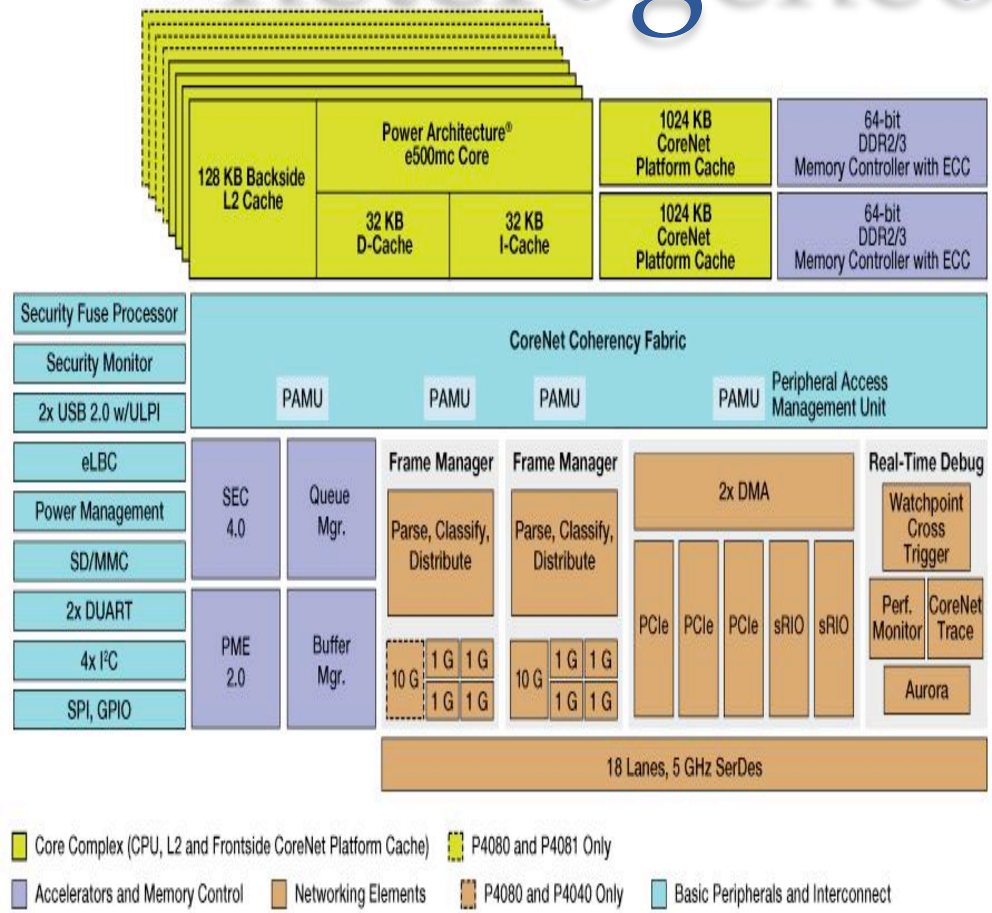








OpenMP + MCA for heterogeneous systems



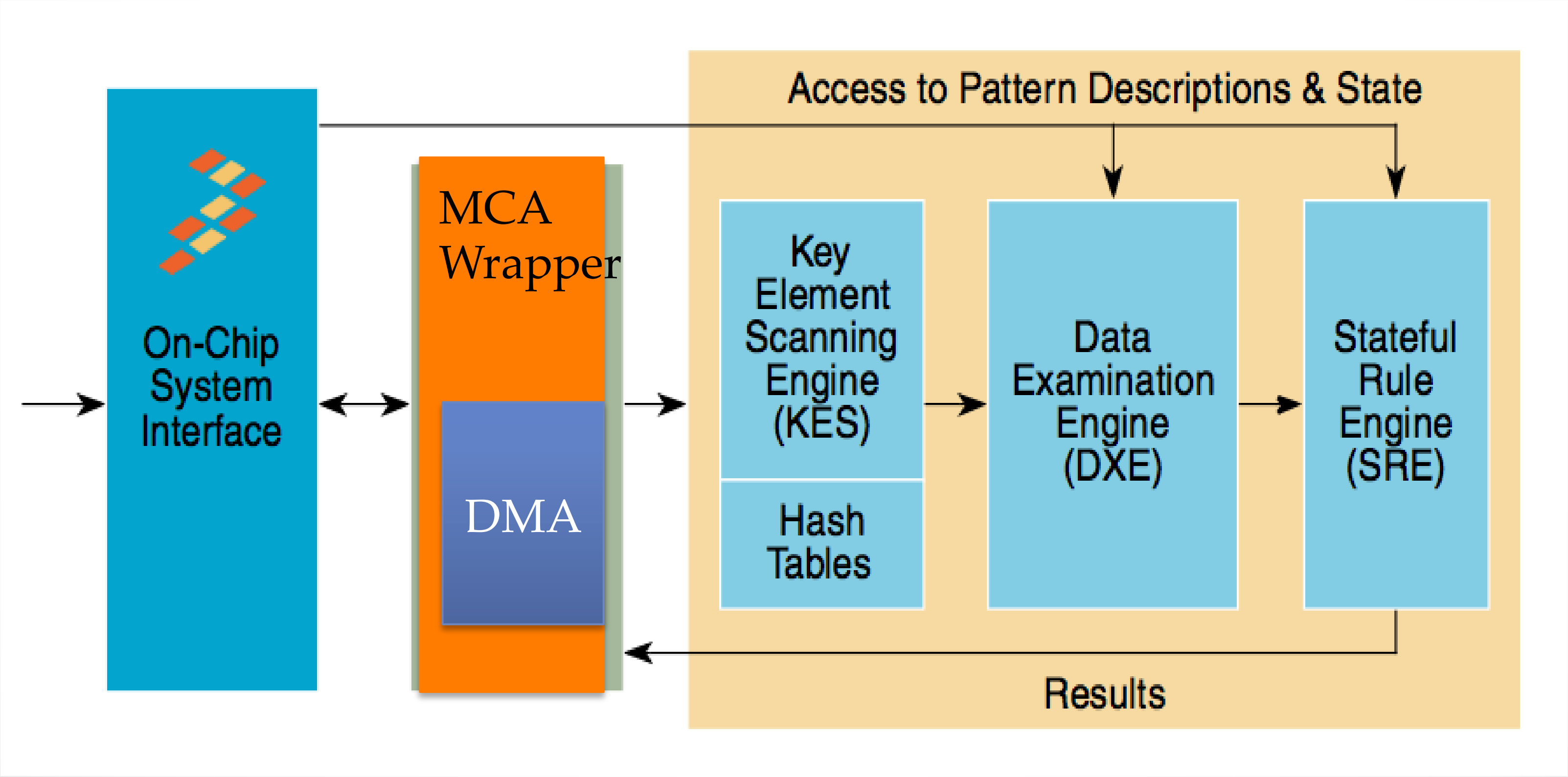
QorIQ P4080 processor

- 4-8 Power architecture e500mc cores
- Accelerators
  - Encryption (SEC)
  - Pattern Matching Engine (PME)
- Target applications:
  - Aerospace and Defense
  - Ethernet Switch, Router
  - Pre-crash detection
  - Forward Collision Warning

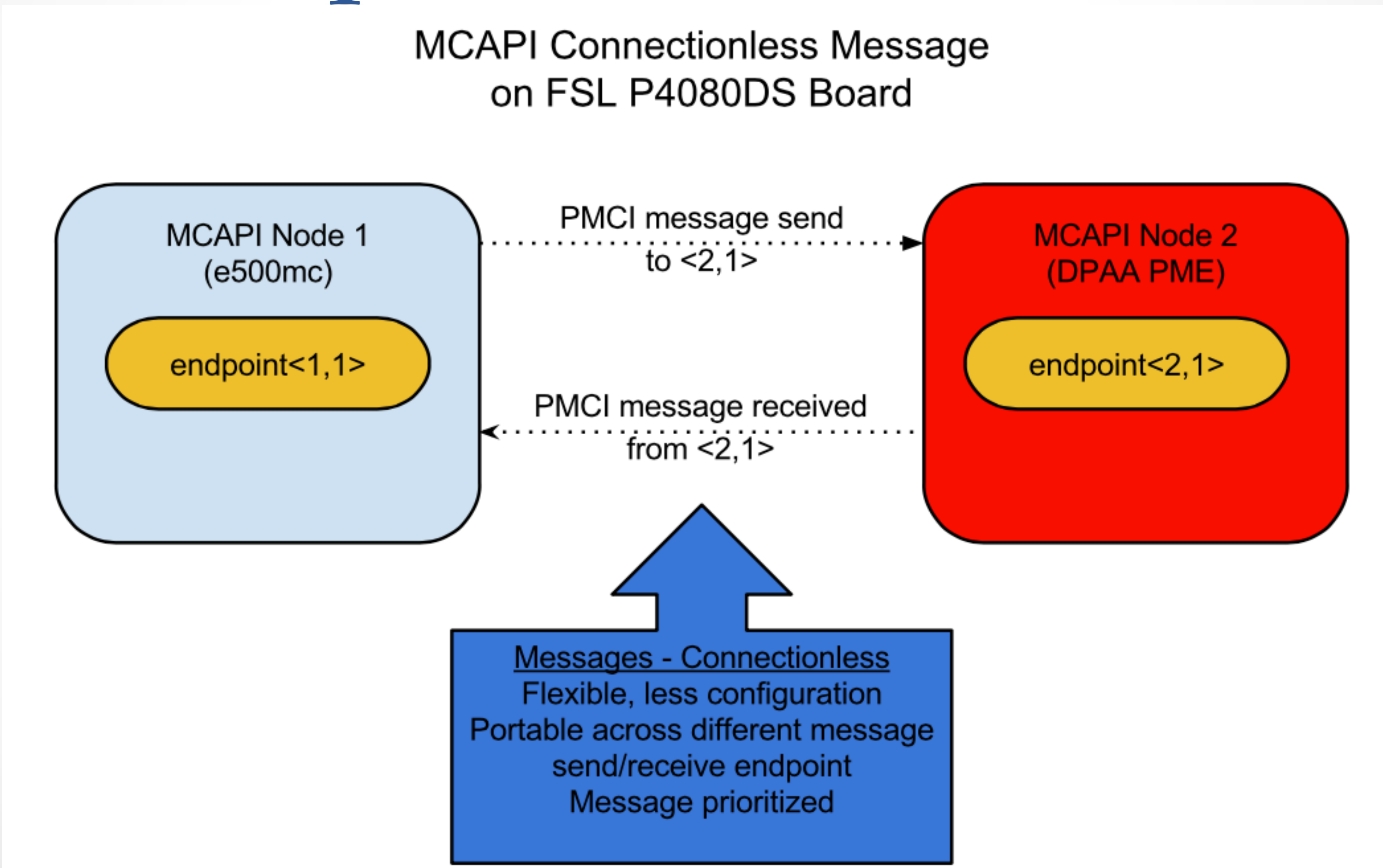
http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=P4080&tid=redP4040

Creating an MCA wrapper

For communication between the power processor and PME



MCAPI Connectionless technique



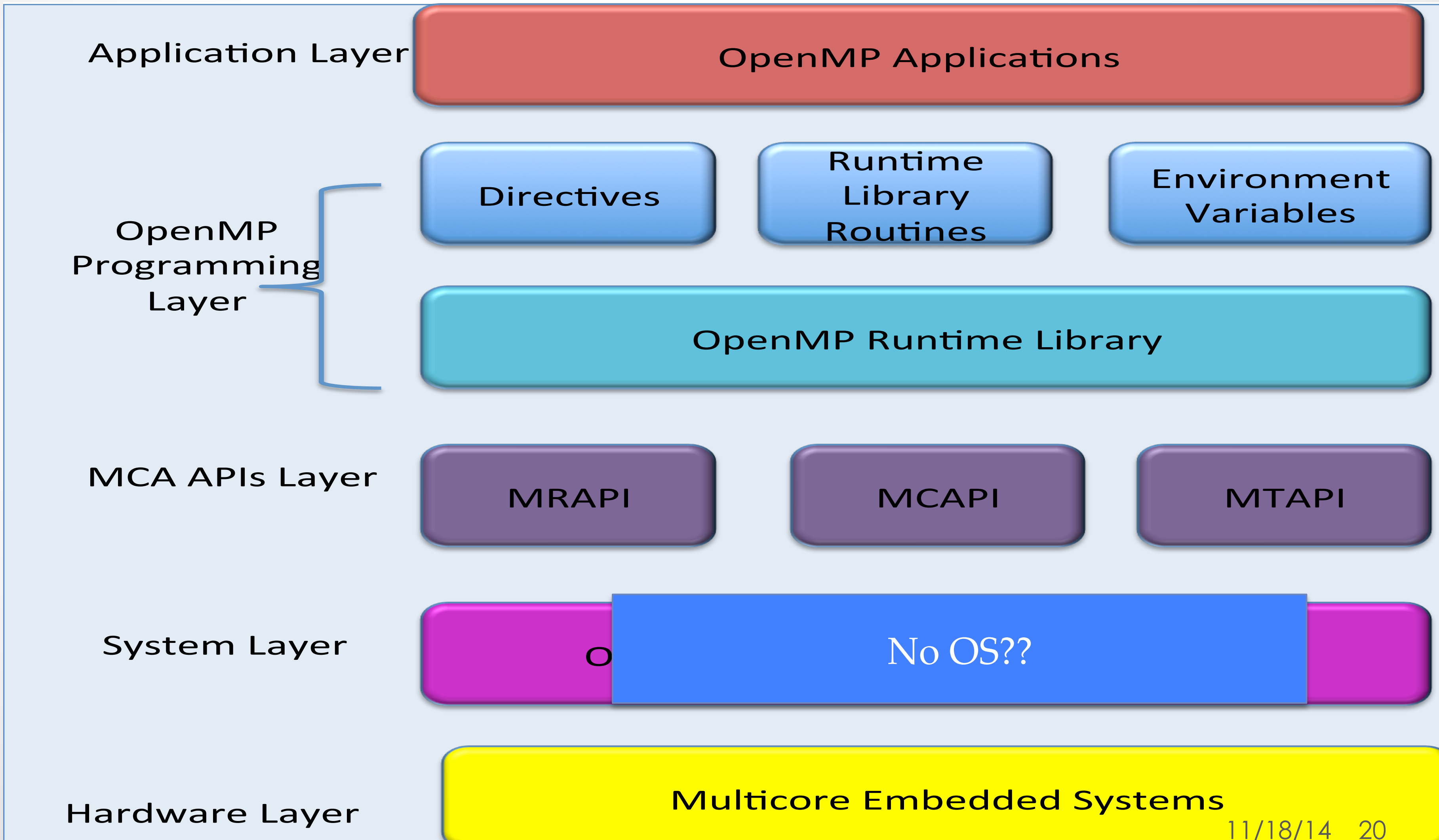
MxAPI for accelerators – lessons learned

- PME does not share memory with the main processor
- Data movement via a DMA channel
- Required thorough knowledge of low-level API very tightly tied to the hardware
    - Time consuming
    - Requires constant support from the vendor to understand the low-level API
- Created an MCA wrapper to abstract all low-level details
    - However the wrapper can be used for devices that relies on that same low-level API.

MTAPI Design and Implementation- Current Status

- On-going work: Implementing MTAPI features
- Writing small test cases to validate the MTAPI implementation
- Need to evaluate the MTAPI implementation on a heterogeneous multicore platform
- Preliminary results demonstrated overhead while communicating with a remote node through MCAPI

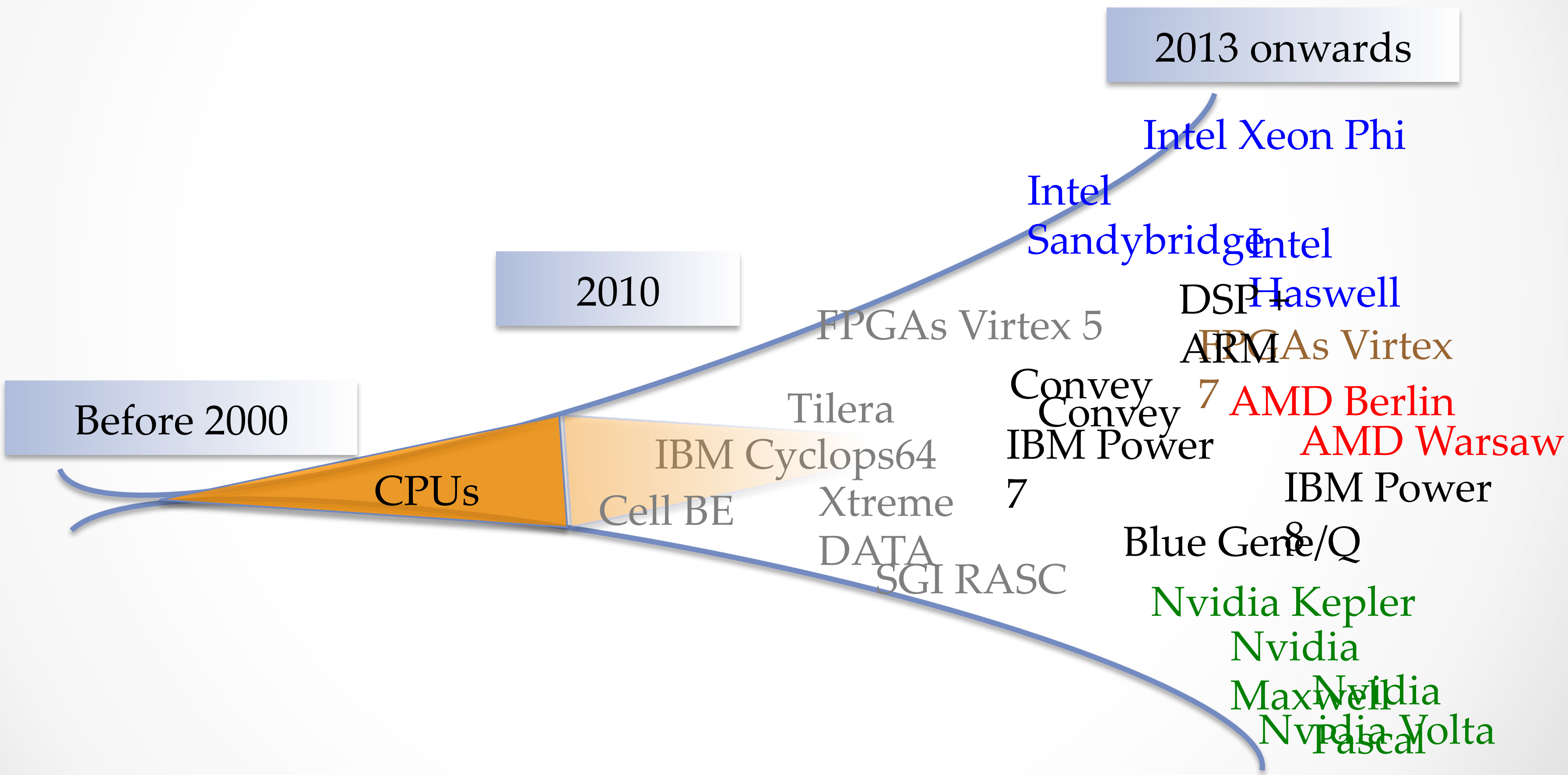
OpenMP and MCA API



Summary

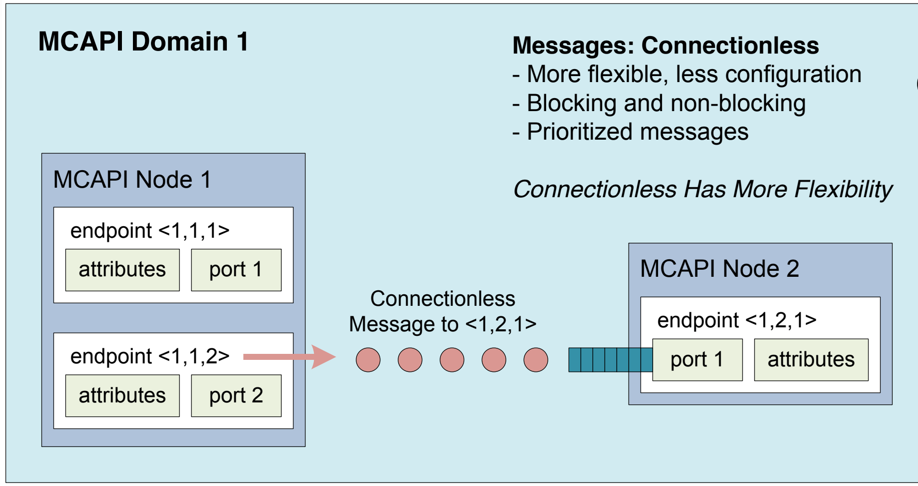
- Extend OpenMP runtime library with MxAPI as the translation layer to target heterogeneous multicore SoCs
  - MTAPI prototype implementation – on-going @ UH
    - SIEMENS : http://www.techdesignforums.com/blog/2014/10/31/siemens-produces-open-source-code-multicore-acceleration/
  - Targeted a specialized accelerators

Accelerators are more than just GPUs

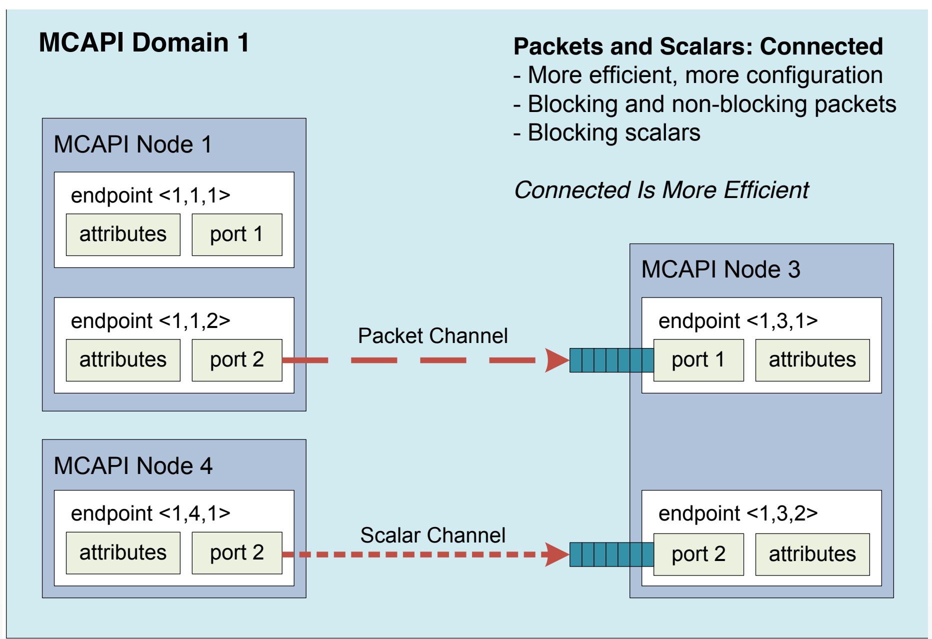


MCAPI – Communication API

Connectionless Message



Connection-oriented Channels



MRAPI Memory Concept