Multicore System Architecture (from the Software Perspective)

Paul Tingey

System Architect (Alliances)

(paul.tingey@windriver.com)

© 2009 Wind River Systems, Inc.

WIND RIVER

Agenda

- Why Migrate to Multicore ?
- Architecture options
- Multicore Runtime Components
- Multicore Development Tools
- Summary

© 2009 Wind River Systems, Inc.

Why migrate to multicore ?

© 2009 Wind River Systems, Inc.

Multicore Device Evolution

Different devices will adopt multicore technologies for different reasons in different ways and at different times.

© 2009 Wind River Systems, Inc.

WIND RIVER

Multicore Use in Devices

Why

- Better products
  - Higher performance
  - More functionality
  - Reliability, safety
  - Lower Cost
  - Greater security
  - Protection of IP

How

- More computing power
  - Absolute
  - Per W, per $, per mm^3
- More cost efficiency
- Stronger partitioning
- Heterogeneous OS support
  - Software reuse
- Redundancy

Cost / Risk

- Increased development cost, time and risk
  - Algorithms, partitioning
  - HW & SW design
  - SW development challenges
  - Benefits are not guaranteed

© 2009 Wind River Systems, Inc.

Multicore Paradox

Benefits

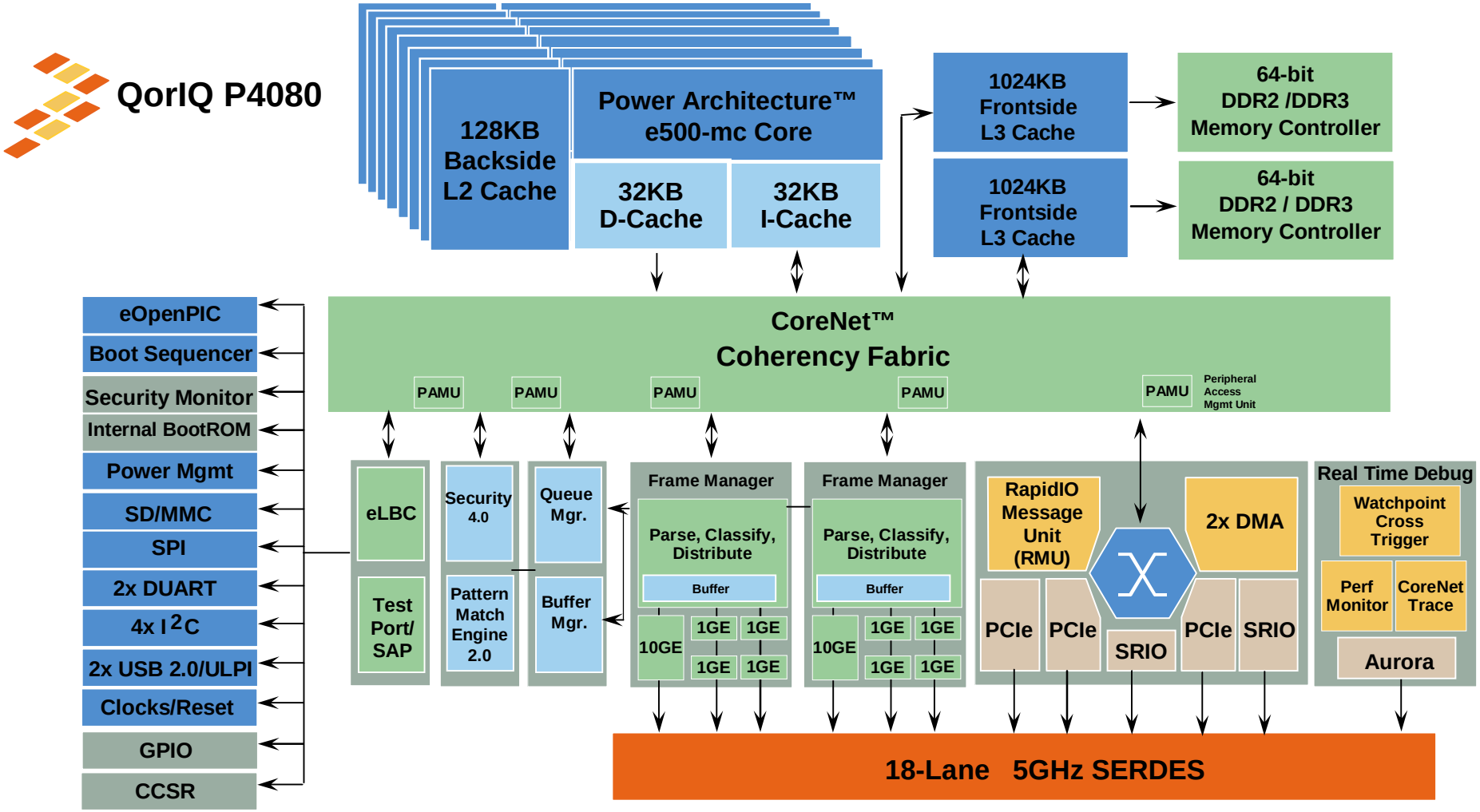
- Performance
- Consolidation
- Reduced cost
- Reduced power

Concerns

- Increased complexity (costs, time-to-market)
- Quality/reliability
- Performance

© 2009 Wind River Systems, Inc.

Freescale QorIQ P4080 Block Diagram



© 2009 Wind River Systems, Inc.

WIND RIVER

Multicore Runtime Components

Multicore Awareness

- Configuration
- Load
- Boot
- Communications

Multiple OS options

- Linux
- RTOS (VxWorks)
- Lightweight OS/Executive
- Bare Metal

Flexible Config Options

- Symmetric Multiprocessing
- Asymmetric Multiprocessing
- Supervised
- Hypervised

Ecosystem

- Hardware
  - e.g. Freescale
- Software
  - e.g. Virtutech

© 2009 Wind River Systems, Inc.

Architecture Options

© 2009 Wind River Systems, Inc.

Operating System Options

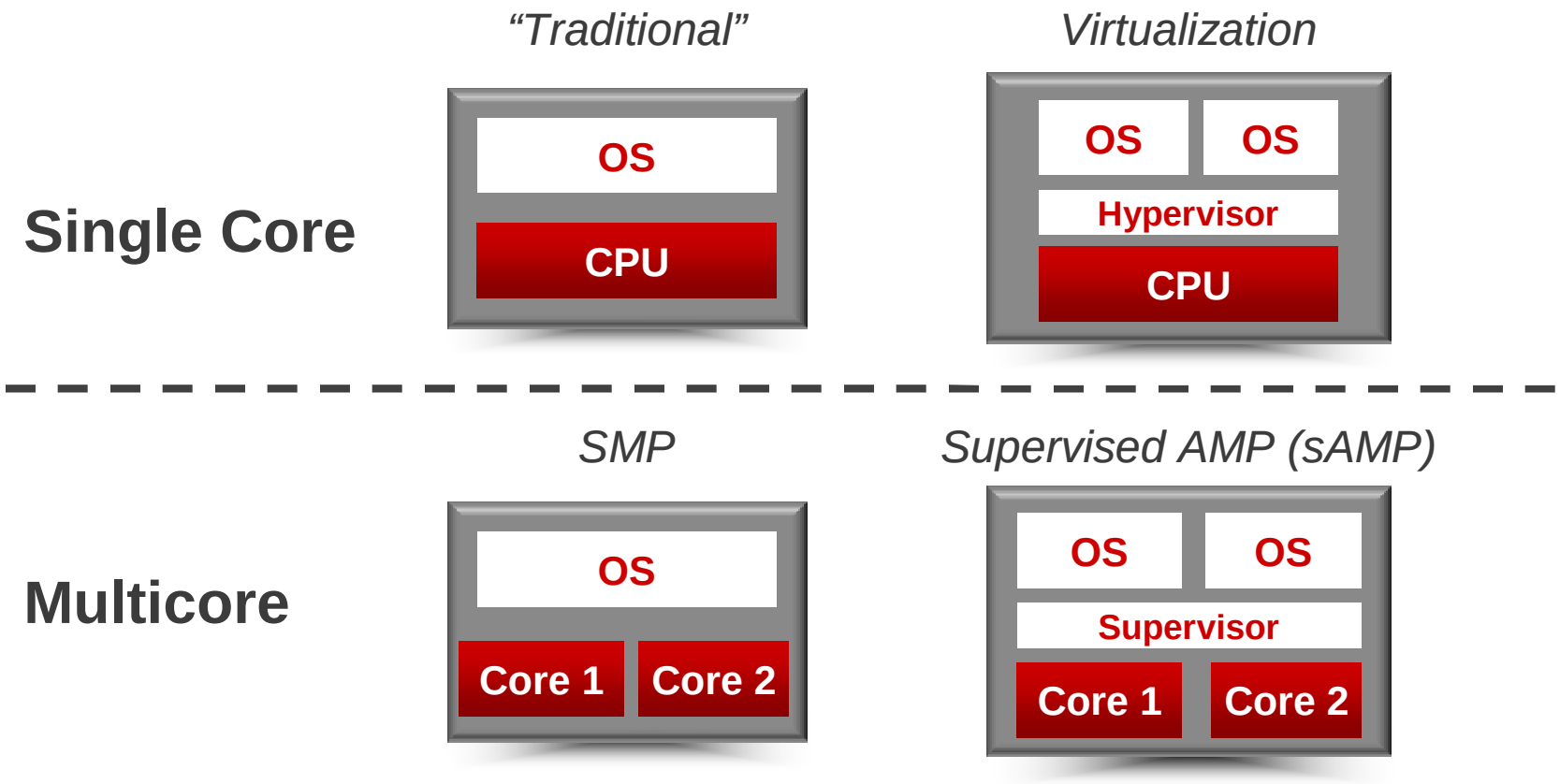
Linux / Win	RTOS	Executive	“Bare Metal”
• Tends to more heavyweight • Often used for GUI or data management	• Medium weight real-time OS • Often used for amalgamating or processing data	• Lightweight kernel (LWE) • A cut-down RTOS • Often used for low-level data processing	• Custom app written directly “onto” the silicon • Still needs some enablement functionality

More complexity, more generic

Less complexity, more application specific

© 2009 Wind River Systems, Inc.

Primary Multicore Software Configurations

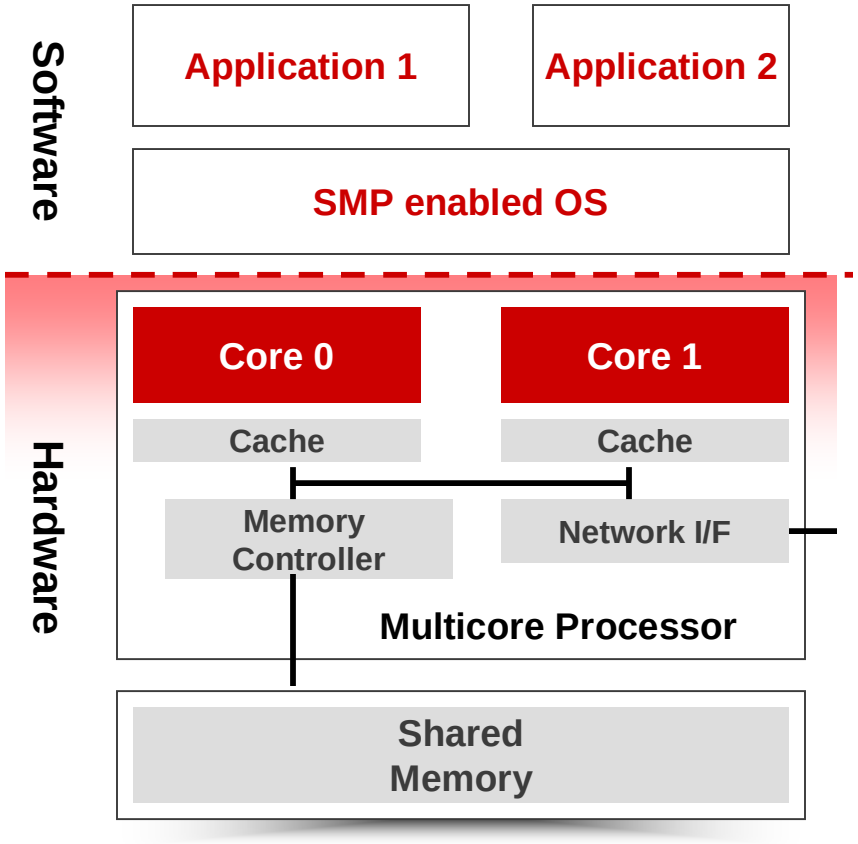


Arbitrary combinations of these primary configurations can be used to create more advanced configurations.

© 2009 Wind River Systems, Inc.

WIND RIVER

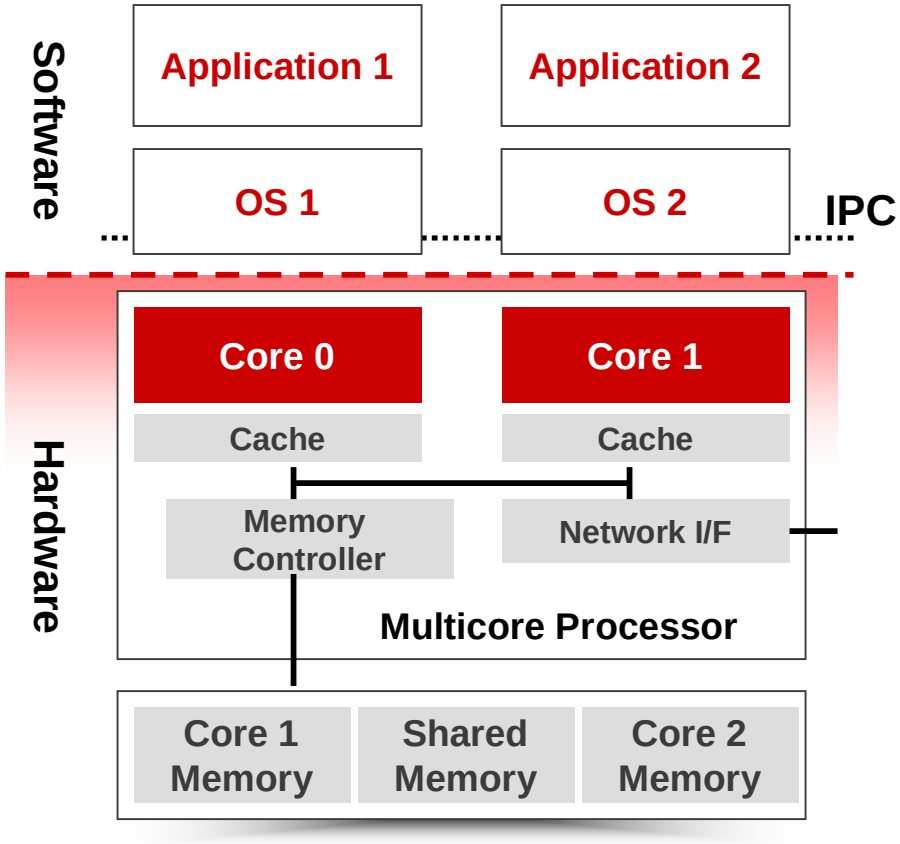
Symmetric Multiprocessing



- An SMP system uses
  - SMP operating system
  - SMP optimized middleware
    - Takes advantage of multiple cores to gain performance
  - or SMP-ready middleware
    - Runs safely in SMP mode
- SMP optimised applications require explicit protection of shared data
  - Memory barriers, spinlocks, atomic operators, etc

© 2009 Wind River Systems, Inc.
WIND RIVER

Asymmetric Multiprocessing

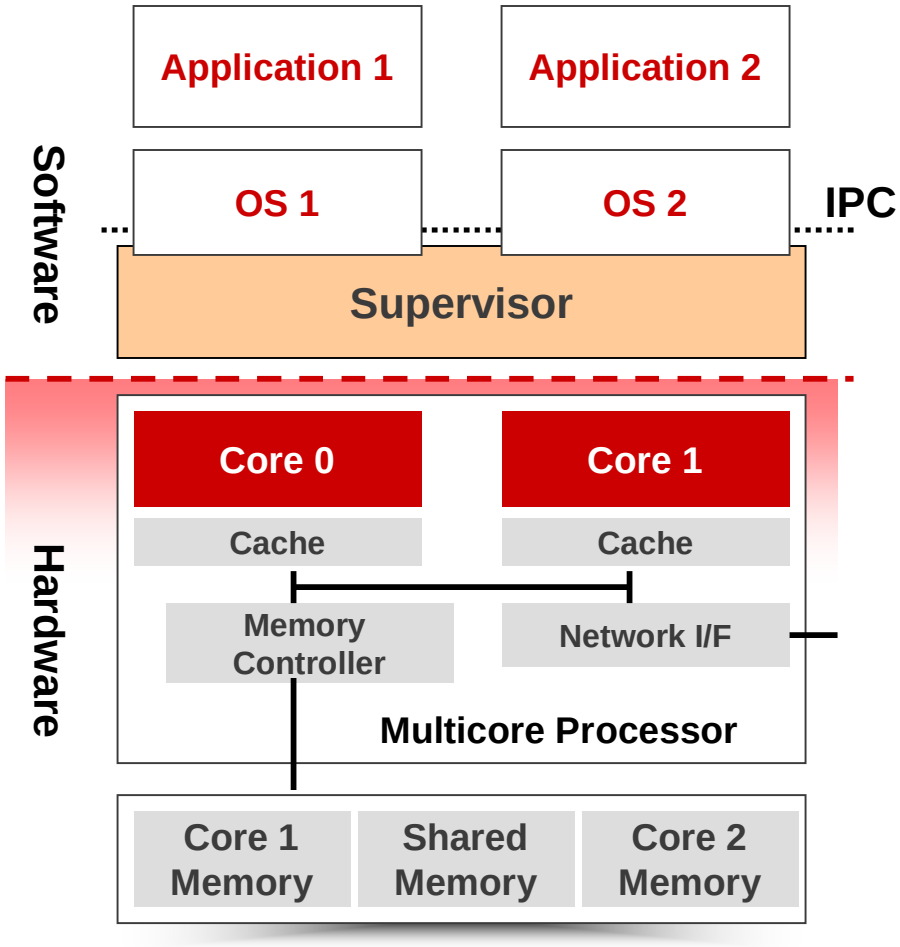


- An AMP system uses
  - A mix of standard operating systems
  - Standard middleware
  - Fast interprocessor communications
- AMP offers the opportunity for consolidation of dissimilar functions into a single device
  - Effective communication between cores is critical
- Hardware/device access can be an issue

© 2009 Wind River Systems, Inc.

WIND RIVER

Supervised Asymmetric Multiprocessing

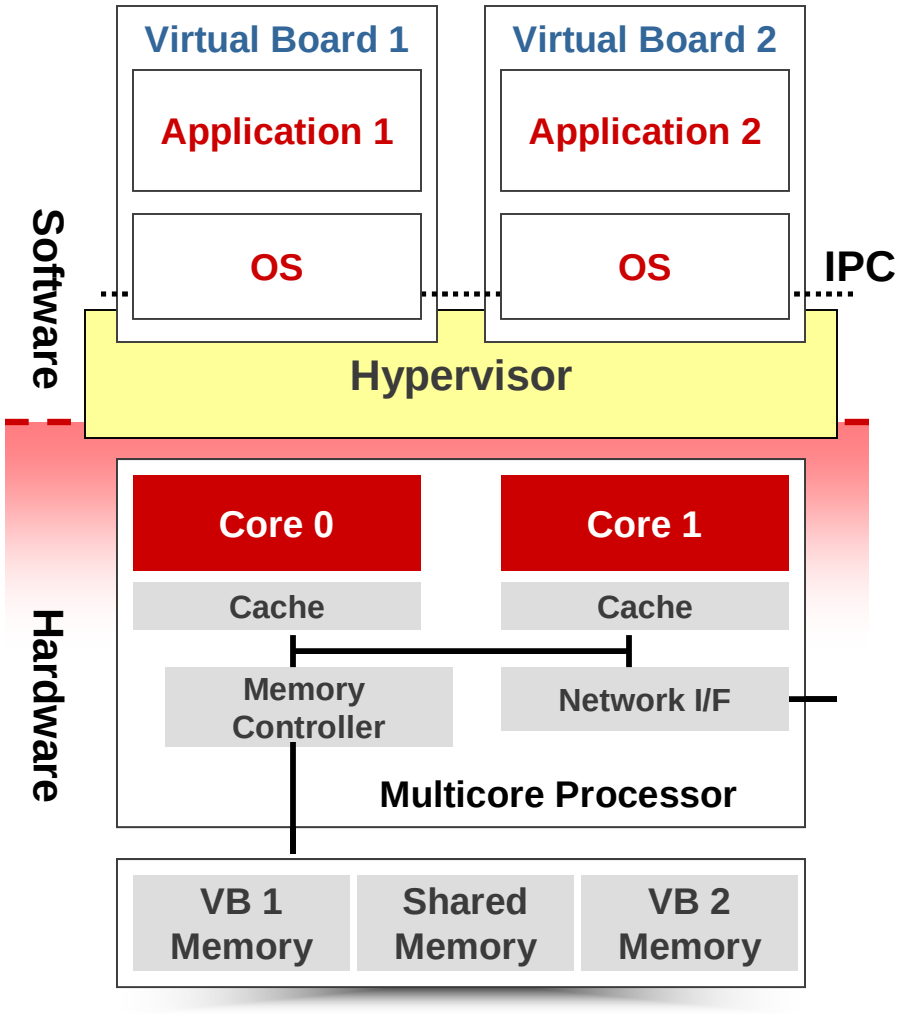


- A Supervised AMP system uses
  - A mix of standard operating systems
  - Standard middleware
  - Fast interprocessor communications
- The supervisor manages access to shared hardware
  - Provides a measure of separation between operating environments
  - May enable controlled reboot of single cores

© 2009 Wind River Systems, Inc.

WIND RIVER

Virtualised Multicore Processing

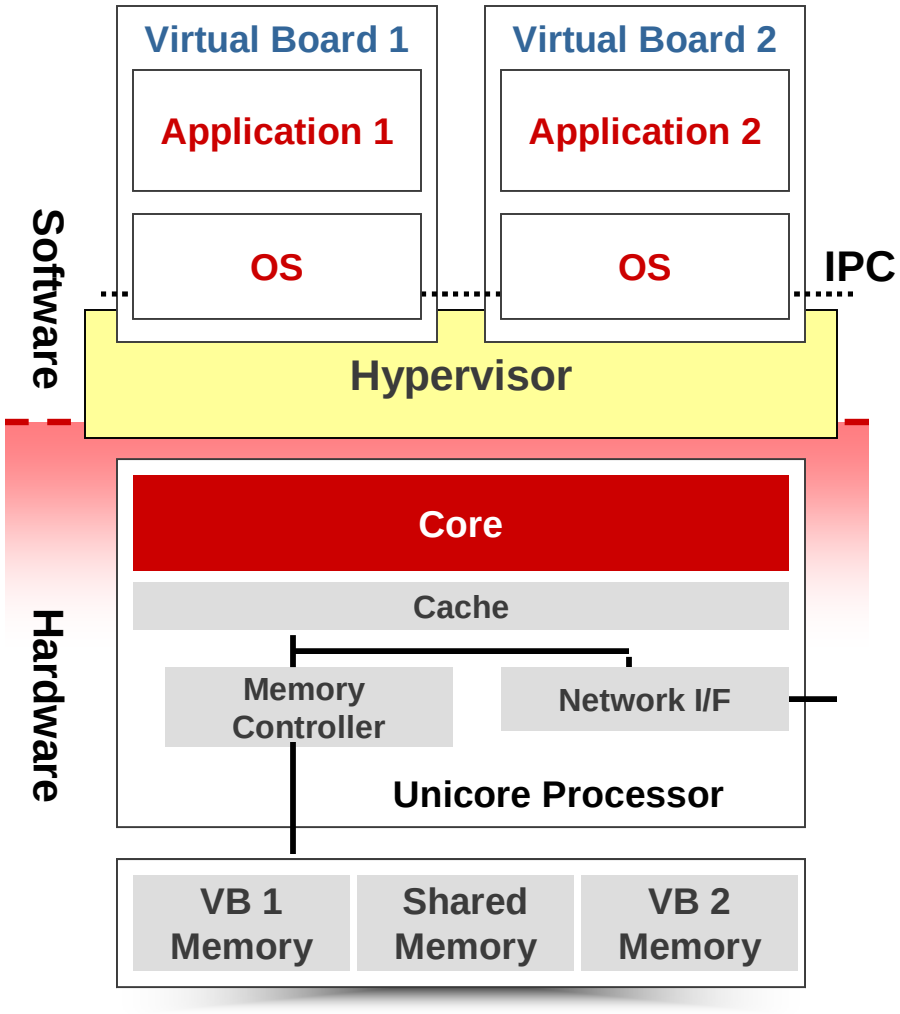


- A Hypervised system uses
  - A mix of standard operating systems
  - Standard middleware
  - Fast interprocessor communications
- The hypervisor manages the load/boot and establishes memory protection for each OS
  - Provides separation between operating environments
  - Enables reboot from failure without complete system reset
- Could form the basis for a certified (safe/secure) system
- Sometimes used for "asset bridges"

WIND RIVER

© 2009 Wind River Systems, Inc.

Virtualised Multicore Processing

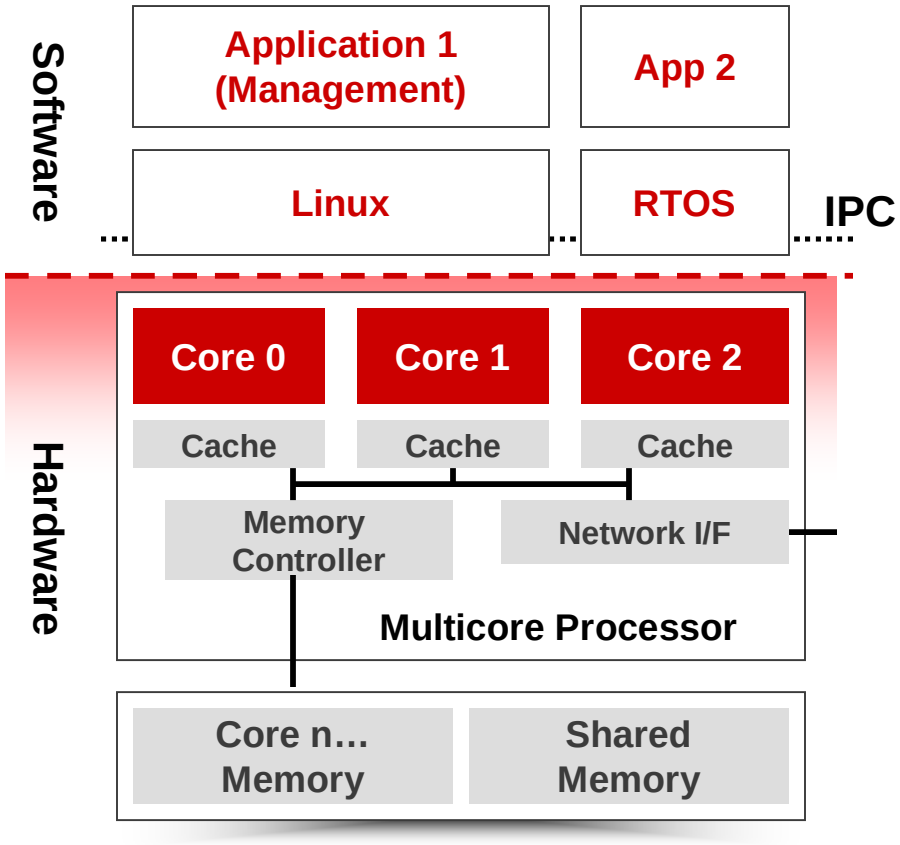


- A single-core Hypervised system uses
  - A mix of standard operating systems
  - Standard middleware
  - Fast interprocessor communications
- Could form the basis for a certified (safe/secure) system

© 2009 Wind River Systems, Inc.

WIND RIVER

Hybrid Multiprocessing

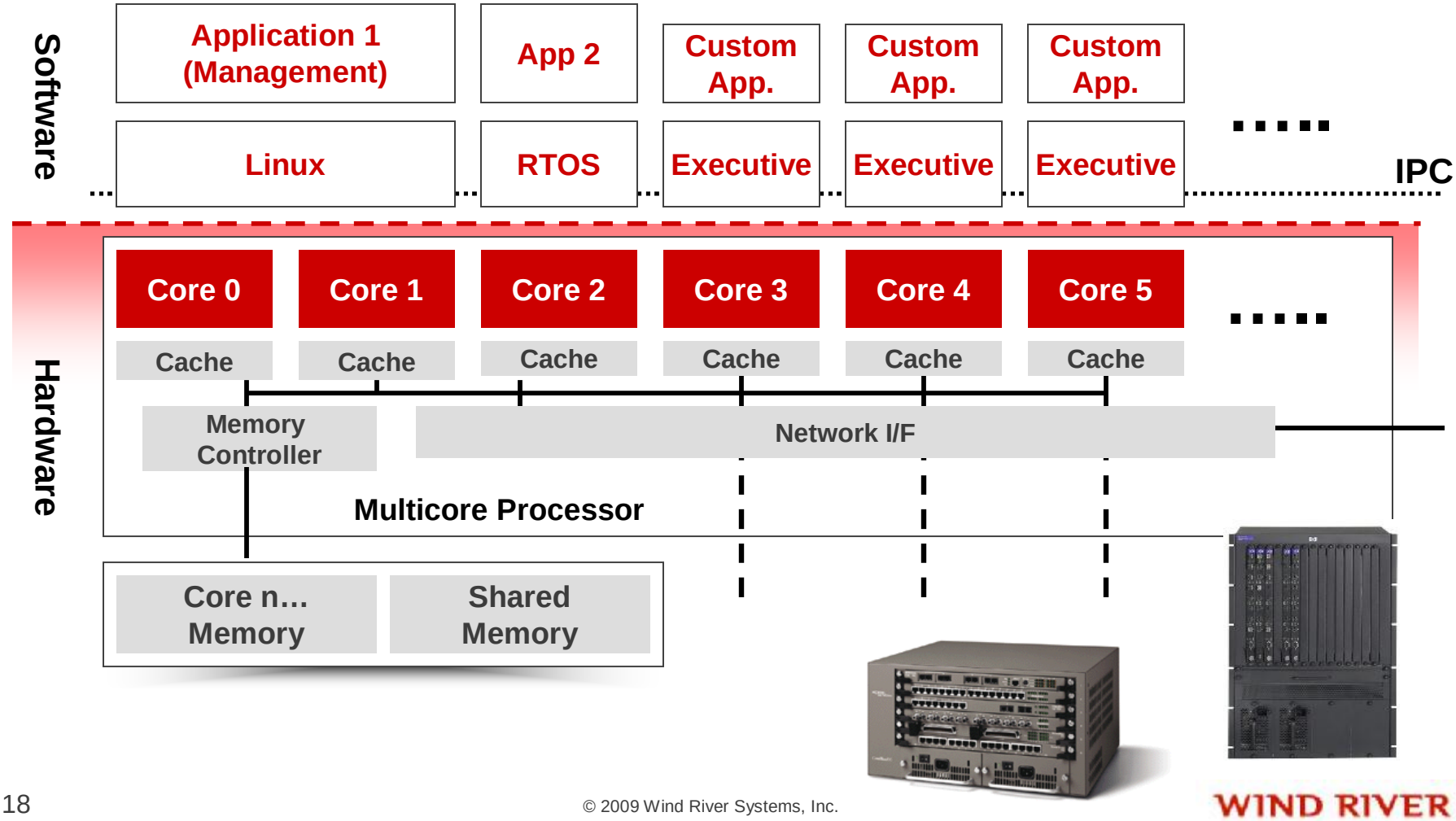


- A hybrid system uses
  - A mix of standard and SMP operating systems
    - Often Linux for management or GUI processing
    - An RTOS for data processing
  - Fast interprocessor communications
- A hybrid architecture may suit cases where legacy applications and data plane processing are consolidated into one device
- Many different configurations are possible !

WIND RIVER

© 2009 Wind River Systems, Inc.

Hybrid Multiprocessing (with offload)



© 2009 Wind River Systems, Inc.

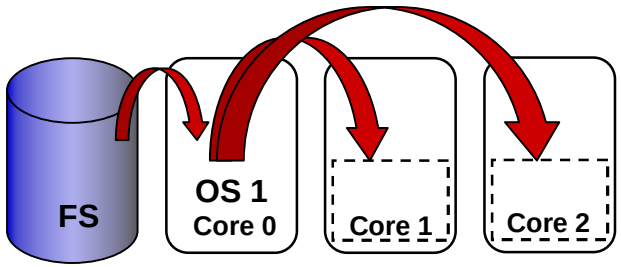
WIND RIVER

Multicore Runtime components

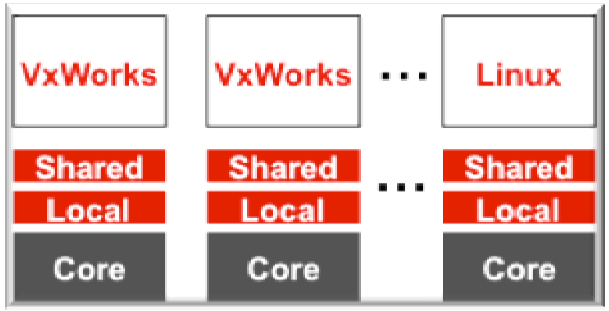
© 2009 Wind River Systems, Inc.

Multicore Run-Time Support

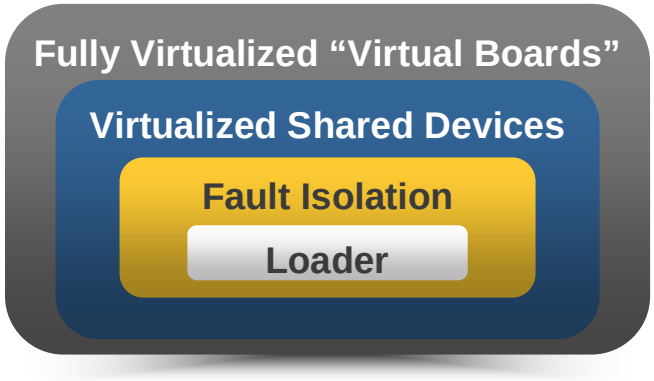
Multi-OS Boot



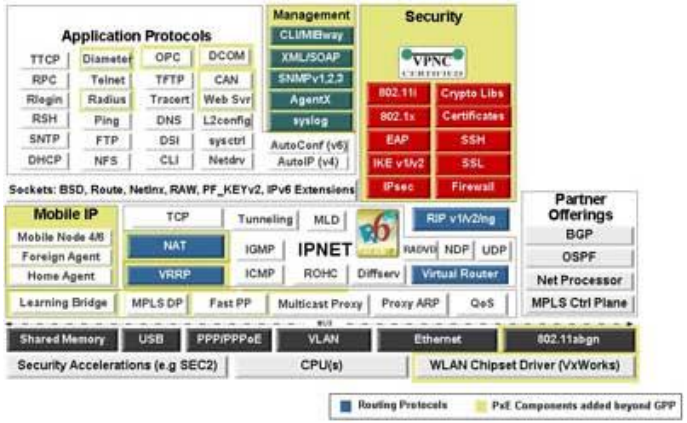
Intercore Communication



"Virtual Board" Support



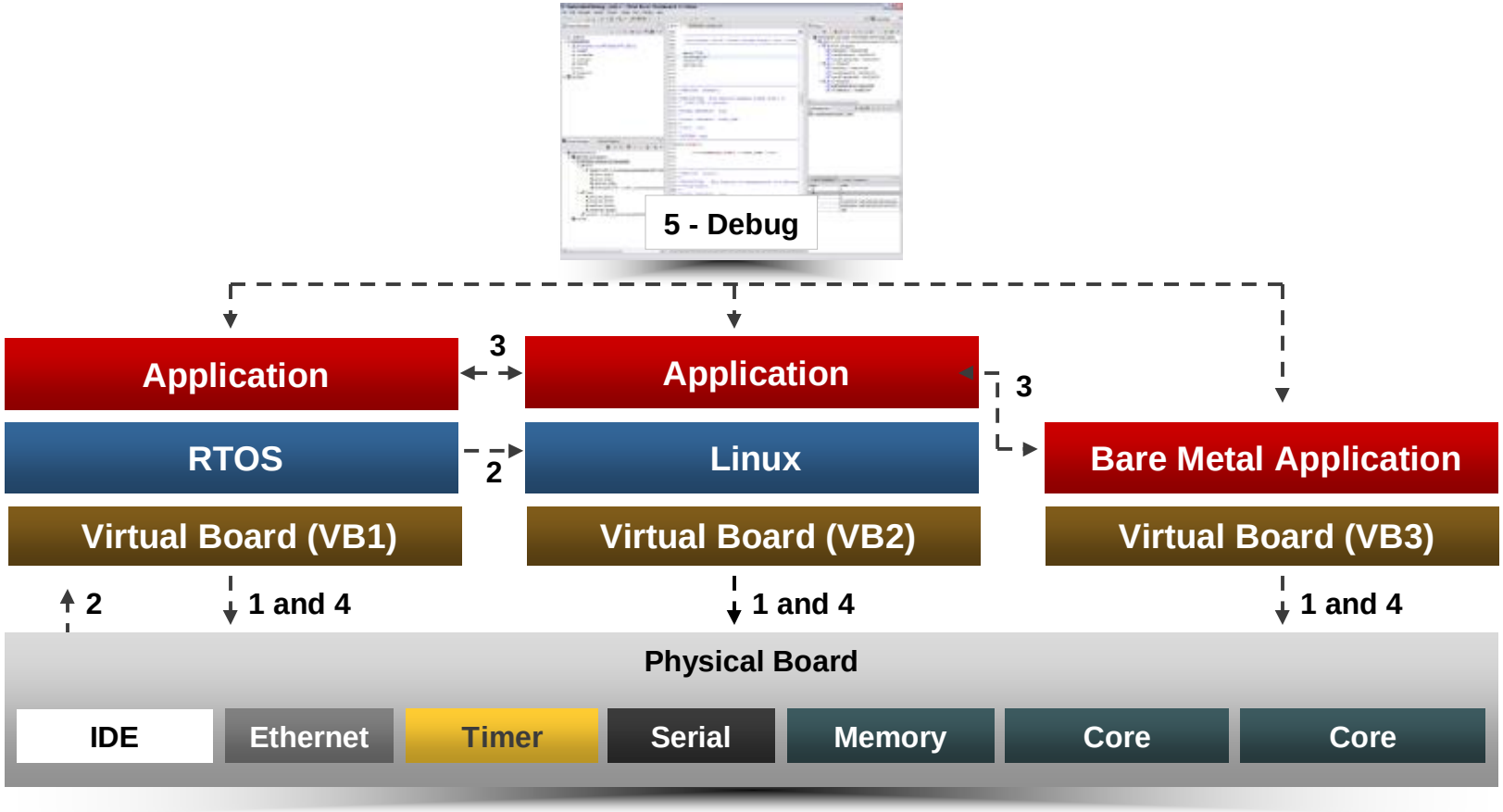
Multicore Enabled Networking



© 2009 Wind River Systems, Inc.

WIND RIVER

Multiple Operating Systems: Challenges



1. Hardware resource partitioning (Memory, Cores, Devices, etc.)
2. Multi-OS booting
3. High-speed IPC
4. Resource access/sharing (consoles, network interfaces, file IO)
5. Debug connectivity

© 2009 Wind River Systems, Inc.
WIND RIVER

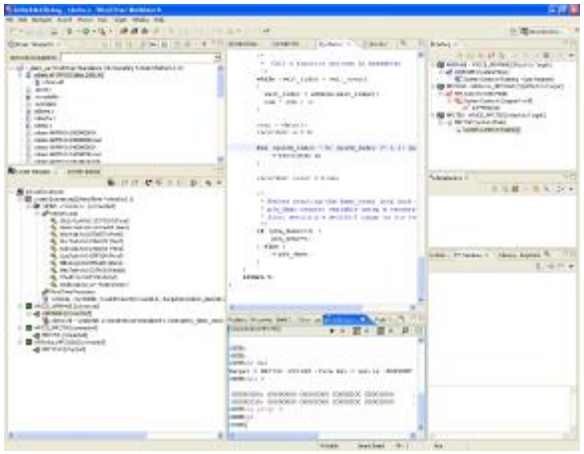
Multicore Development Tools

© 2009 Wind River Systems, Inc.

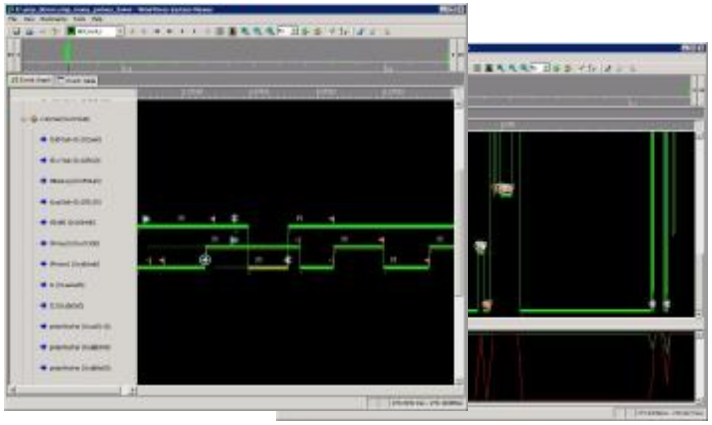
WIND RIVER

Multicore Debug & System Analysis Tools

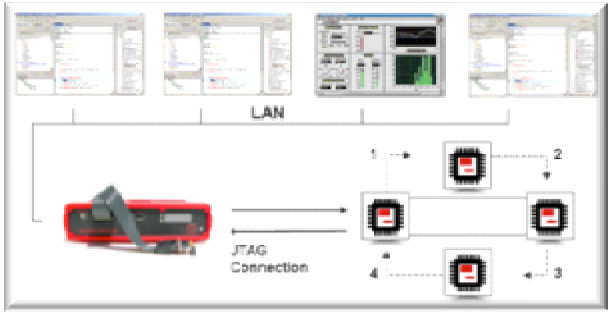
Debugging Multiple Contexts



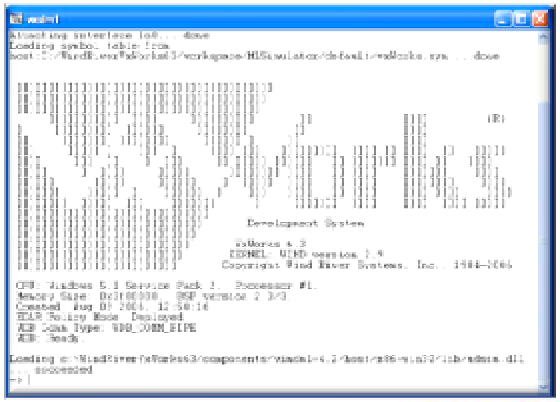
Viewing Asynchronous Events



Multicore On-Chip Debugging



Prototyping and Simulation



© 2009 Wind River Systems, Inc.

WIND RIVER

Multicore On-Chip Debugging

Color-Coded Processor Contexts

Selected Context Registers

© 2009 Wind River Systems, Inc.

WIND RIVER

Summary

- Multicore and virtualization technologies are creating an inflection point in the device industry
- Software design for multicore systems is inherently complex
  - and requires close collaboration between hardware and software engineers
- There is no single solution for all situations
  - solutions are evolving to fit particular application areas
- Multicore runtime and development tools should be chosen carefully to ensure a efficient and effective system design and development

© 2009 Wind River Systems, Inc.

WIND RIVER