Software Development on Multicore Power® Architecture Platforms
Dr. Xin-Xin Yang
Senior Manager, Software R&D
Networking & Multimedia Solutions Group, Freescale Inc.

CELEBRATING 21 YEARS OF POWER ARCHITECTURE ANNIVERSARY

ASIA
POWER
ARCHITECTURE
CONFERENCE

SHANGHAI, CHINA  OCTOBER 25, 2012
THINK

POWER

CHOOSE

POWER

INNOVATION

COLLABORATION

GROWTH

CELEBRATING 21 YEARS OF POWER ARCHITECTURE ANNIVERSARY
Agenda

• Development and Evolution of Multicore Power® Architecture Platforms
• System Characteristics in Multicore Platforms and Challenges on Software Design
• Multicore Software Solution Model
• Key Technologies in Software on Power® Architecture
  – Virtualization
  – User Space I/O
  – Performance Optimization
  – Power Management
  – Distribution and Linux SDK
• Q&A
Development and Evolution of Multicore Power® Architecture Platforms
Wide Application of Multicore Power Architecture

Service Provider
- Core routers
- RNC
- Edge routers
- Central office / broadband access

Enterprise/Data Center
- Multiservice routers
- Cloud Computing
- WLAN access points
- Ethernet switches
- Security / UTM appliances

Industrial & Aerospace
- Factory & building automation
- Automotive
- Machine to machine communications
- Power protection
- Medical imaging & networks
- Aerospace communications, radar, sonar

SOHO / Consumer
- Home media distribution
- Residential gateways
- Small form-factor control systems
An example: Application in Networking

**Drivers**
Performance / Watt, system cost & power, Increased bandwidth, increased number of subscribers, increased processing per packet, and software complexity

**Application Examples**
Cloud networking, Routing & switching; control plane + data plane consolidation, multimedia content processing, 4G wireless processing. Servers: Multiple users access similar service simultaneously
Multicore Power: Leadership in High-Performance Processors

**Leading innovation in networking architectures**
- QorIQ: Broadest scalable family of processors in the market
- Industry-leading integration, performance, power

**Increasing software investment**
- Innovative Software-aware solutions
- Industry-leading virtualization support

**Strong enablement & ecosystem**
- True programmability and Ease of use focus
- Rich set of reference designs and development platforms

**Trusted partner**
- Longevity program
- Global support resources, quality, service

---

**QorIQ**
- 2-24 Core CPU
- QorIQ AMP T4240 T2080, T1042

**QorIQ**
- 1-8 Core CPU
- QorIQ P10xx – P50xx
- PowerQUICC Host Processors

---

**Service Provider**  
**Enterprise**  
**Consumer Access**  
**Industrial and Aerospace**
Multicore Power Platforms Evolution

**Single Core with Hardware Accelerators**
- CPU
- Shared Bus
- I/O
- I/O
- I/O
- I/O
- I/O
- Accel

Hardware acceleration provides better performance/power efficiency than general purpose compute.

**Homogeneous Multicore**
- CPU
- CPU
- CPU
- CPU
- I/O
- I/O
- I/O
- Accel

General-purpose processing with some parallelism; Hardware acceleration for specific tasks.

**Performance Density Multicore**
- CPU Cluster
- CPU Cluster
- CPU Cluster
- CPU Cluster
- CPU Cluster
- CPU Cluster
- Shared L2
- Shared L2

“Heavy” Threading and Clustering improves performance/power density.

Increasing Demand for Software-Awareness

The Power Architecture and Power.org word marks and the Power and Power.org logo and related marks are trademarks and service marks licensed by Power.org.
**Multicore Power Platform— T4240 as Example**

**Processor**
- 12x e6500, 64b, up to 1.8GHz
- Dual threaded, with 128b AltiVec
- Arranged as 3 clusters of 4 CPUs, with 2MB L2 per cluster; 256KB per thread

**Memory SubSystem**
- 1.5MB CoreNet Platform Cache w/ECC
- 3x DDR3 Controllers up to 2.1GHz
- Each with up to 1TB addressability (40 bit physical addressing)
- HW Data Prefetching

**CoreNet Switch Fabric**

**High Speed Serial IO**
- 4 PCIe Controllers, with Gen3
  - SR-10V support
- 2 sRIO Controllers
  - Type 9 and 11 messaging
  - Interworking to DPAA via Rman
- 1 Interlaken Look-Aside at up to 10GHz
- 2 SATA 2.0 3Gb/s
- 2 USB 2.0 with PHY

**Network IO**
- 2 Frame Managers, each with:
  - Up to 25Gbps parse/classify/distribute
  - 2x10GE, 6x1GE
  - HiGig, Data Center Bridging Support
  - SGMII, QSGMII, XAUI, XFI

**Device**
- TSMC 28HPM Process
- 1932-pin BGA package
- 42.5x42.5mm, 1.0mm pitch

**Power targets**
- ~30W typical power at 1.8GHz with IO

**Datapath Acceleration**
- **SEC**- crypto acceleration 40Gbps
- **PME**- Reg-ex Pattern Matcher 10Gbps
- **DCE**- Data Compression Engine 20Gbps
E6500 Core Complex

- 64-bit Power Architecture
- Upto 2.0 GHz operation
- Two threads per core
- Dual load/store units, one per thread
- 40-bit Real Address
  - 1 Terabyte physical addr. space
- Hardware Table Walk
- L2 in cluster of 4 cores
  - Supports Share across Cluster
  - Supports L2 memory allocation to Core or thread

Power Management
- Drowsy: Core, Cluster, Altivec
- Wait-on-reservation instruction
- Traditional modes

AltiVec SIMD Unit (128b)
- 8,16,32-bit signed/unsigned integer
- 32-bit floating-point
- 192 GFLOP (2GHz)
- 8,16,32-bit Boolean

Virtualization
- Hypervisor
- LRAT
  - Logical to Real Addr. translation mechanism for improved hypervisor performance

CoreMark™ Benchmarks
- Dual x12 thread server processors @2.266GHz *
- 32 core processor @1.5GHz *
- T4240 @1.8GHz

*Source: www.coremark.org
System Characteristics in Multicore Platforms and Challenges on Software Design
System Characteristics in Multicore Platforms

- System Complexity (As Shown in T4240 Block Diagram)
- Heterogeneous CPUs with multilevel cache hierarchy
  - Threaded cores, co-located control/data planes
  - Operating systems, bare metal apps running on virtual partitions
- Specialized hardware accelerators, I/O ports with support for load spreading and selective data stashing
- Performance
- Power Consumption
- Usability
Challenges on Software to Support Multicore Power Platforms

• Legacy software written for non-symmetric multiprocessing (SMP) systems
  - Multithreading is a must for scaling with multicore
  - Spin-lock safe
  - Possible to affine application threads and interrupts to a core
  - Locks are a must for shared resources in multithreaded applications

• Parallel thread execution in real time
  - Difficult to reproduce bugs
  - Data corruption more difficult to nail down

• Cache efficiency requires careful software design
  - Cache line sharing
  - Cache line thrashing due to threads running on different cores

• Performance Optimization
  - Performance nonlinear improvement
  - Workload balancing between difference cores

• Power Consumption
  - Power management scheme and algorithm on different cores.
Key Technologies in Software on Power® Architecture

- Multicore Software Solution Model
- Virtualization
- User Space I/O
- Performance Optimization
- Power Management
- Distribution and Linux SDK
Multicore Software Solution Model

- **Virtualization:**
  - Hypervisor
  - KVM
  - Linux Container

- **Linux:**
  - Control Plane Processing
  - SMP Support

- **Linux SDK:**
  - Silicon Optimized
  - Full Featured
  - Open Source

- **User Space:**
  - Data Path Acceleration Architecture
  - Other Key IP Blocks (RMAN, XMAN...)

- **Communication Stacks/APIs:**
  - Silicon Optimized
  - Open And Scalable

- **Multicore Applications:**
  - SMP and AMP programming Models
  - Component Model For Scalability
Partitioning vs. Virtualization

**Partitioning**
- Hardware consolidation
- Partitioned/dedicated resources, minimal sharing.
- Dedicated CPUs, I/O devices

**Virtualization**
- N virtual machines
- Resource sharing, oversubscription
- Virtual I/O
- Highly virtualized environment
- Live migration

The Power Architecture and Power.org word marks and the Power and Power.org logo and related marks are trademarks and service marks licensed by Power.org.
Virtualization Software Solutions

- KVM
- Linux Containers
- Topaz
- ePAPR
Key Virtualization Technology

- KVM is a Linux kernel driver
- User space tool, QEMU, is used in conjunction with KVM
- Solution is open source
- Number of virtual machines is only limited by available resources (CPU cycles, memory)

- A lightweight framework for partitioning an SoC
- Gives you the best of both worlds—bare metal performance with enforced partitioning and fully architected approach to meeting AMP requirements
- Solves many of the headaches of running multiple unsupervised OSes
- Threads appear as cores to OS

- Containers provide OS level virtualization
- Provides low overhead, lightweight, secure partitioning of Linux applications into different domains
- Can control resource utilization of domains—CPU, I/O bandwidth
Freescale Software Architectures for Power Based QorIQ - Evolution

- **Linux SMP**
  - Multicore Hardware
  - Embedded Hypervisor
  - Topaz
  - KVM: Linux-based Hypervisor
  - Linux SMP
  - Partitioning / supervised AMP, failover
  - Consolidation, high performance user space DPA engines

- **Unsupervised AMP**
  - Multicore Hardware
  - Linux
  - OS Virtualization
  - CPU, I/O virtualization
  - Isolated Containers, resource control & monitoring
  - Leverage both KVM & LXC
  - Linux OS Convergence

- **USDPAA on Linux SMP**
  - Multicore Hardware
  - Linux
  - OS Virtualization
  - Isolated Containers, resource control & monitoring
  - USDPAA on Linux SMP
  - Leverage both KVM & LXC

The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.
User Space I/O (USDPAA) – Features and Benefits

User space apps get direct zero-copy access to all DPAA Hardware

Benefits:

• Rich and flexible environment – use Linux standard services rather than inventing new ones (C++, 36-bit, 8th core, debug, etc).
• Standard – Linux is ubiquitous and supported from multiple sources.
• Provides high-performance run-to-completion but other use-cases also – a use case super-set.
USDPAA Purpose

Low-level, high-efficiency programming for subset of system software that represents bulk of workload and is worth a lot of optimization

- Fast-path offload of a router stack (like a software ASIC)
- Layer 2 and up in a cell base station
- Software fed by bump-in-wire net stack (like IPSEC), often with special-case assumptions allowed because network is controlled.

Bare metal software migrates from running alongside Linux on its own cores to running within a Linux user space process.
### Linux User Space DPAA Core Affinity

#### USDPA Application Can Use 1 to 8 Cores (P4080)
Each thread has a dedicated portal and is affine to a core, 1 thread per core

**USDPA Memcached**

- Thread 1: QPortal, BPortal
- Thread 2: QPortal, BPortal
- Thread 3: QPortal, BPortal
- Thread 4: QPortal, BPortal
- Thread 5: QPortal, BPortal
- Thread 6: QPortal, BPortal
- Thread 7: QPortal, BPortal

**Core 0 has portal for kernel use and standard Linux networking**

**Core 1 to Core 7**

**Other Processes**
- Other Processes
- BQMan / QMan
- FMan Ethernet Ports
- kernel
- Net Stack
- Eth Driver
- Other Driver

**7 cores are isolated but 1 can run an USDPA thread as well as other processes.**

*The Power Architecture and Power.org word marks and the Power and Power.org logo and related marks are trademarks and service marks licensed by Power.org.*
Layer 2 Optimizations

**Core Affinity**

- Control Plane Apps (iptables, iproute, IKE etc)
- VortiQa CP + NMS
- Linux Network Stack
  - ASF Linux Control
  - ASF VortiQa Control
  - ASF API
- VortiQa Network Stack
- VortiQa CP + NMS
- Linux User space
- VortiQa Network Stack
- VortiQa CP + NMS

**Bypass**

**Hugetlfs**

- target hugetlb page
- memory defragmentation

**User Space IO - UIO**

- Thread
- Thread
- API
- Setup Code in Kernel
- Core
- Accelerator
- Core
- Accelerator

**Layer 2 Optimizations**

- Net Stack
- Other Driver
- BMen / QMen
- FMen Ethernet Ports
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal

- C0
- C1
- C2
- C3
- C4
- C5
- C6
- C7

- Net Stack
- Other Driver
- BMen / QMen
- FMen Ethernet Ports
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal
- QPortal

- Other Processes
- USDPA Memcached
- thread
- thread
- thread
- thread
- thread
- thread
- thread
- thread

- isolated cores

The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.
Layer 3 Optimizations: TCP Performance Optimization Roadmap

- TCP receive hardware offload
  - Non-DPAA SDK
  - Unified SDK

- FAST-TCP-ACK
  - Non-DPAA SDK
  - Unified SDK

- L4 SKB-recycling
  - Non-DPAA SDK
  - Unified SDK

- Soft-TSO
  - non-DPAA SDK
  - Unified SDK

- Receive file support
  - non-DPAA SDK
  - Unified SDK

- Cache optimizations (L2SRAM, data prefetch)
  - Unified SDK

- Adaptive load-balance
  - Unified SDK

Timeline:
- 2010
- 2011
- 2012
- 2013

Status:
- Done
- Implementing
- Planning

Notes:
- Non-DPAA
- DPAA

Power.org™

The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.
Goals of Power Management on Power Architecture

• Enabled: provide software support to the hardware PM features in our silicon

• Easy-to-use: provide OOB software solution for our customers to improve power efficiency of their product with no change or little change to their
  – Hardware Design
  – Software Design

• Measurable: make the power consumption and temperature measurable and monitored.

• Optimized: save as much as possible power with no impact or acceptable impact on the system performance
Standard Linux PM Features

User Space

PM control utilities (DPM)/cmdline

Sysfs

Kernel Space

Linux device model
- pm_ops
- runtime_pm_ops
- wakeup source
- Clock
- Power domain
- Qos

PM core
- suspend
- hibernation
- Runtime PM
- Autosleep
- Wake lock
- Pm_qos

CPU mgmt
- CPU freq
- CPU idle
- CPU hotplug
- SCHED_PM

Misc
- hwmon

The Power Architecture and Power.org word marks and the Power and Power.org logo and related marks are trademarks and service marks licensed by Power.org.
Freescale Linux PM Components on Power Platforms

Linux device model
- pm_ops
- runtime_pm_ops
- wakeup source
- Clock
- Power domain
- Qos

Freescale Domain
- Device drivers
  - Suspend hooks
  - Runtime_pm hooks
  - wakeup source hooks
  - PM Qos hooks
  - TMU driver
  - MPIC timer driver
  - Power Monitor

SoC platform support
- Power domain
- Clock domain
- Qos

Core support
- e500v2 PMC
- e500mc RCPM
- e6500 new states

Misc
- hwmon

PM core
- Suspend

CPU mgmt
- CPU freq
- CPU idle
- CPU hotplug
- SCHED_PM

In Planning
- In development
- Already done

The Power Architecture and Power.org word marks and the Power and Power.org logo and related marks are trademarks and service marks licensed by Power.org.
Power Management Software Roadmap on Power Platforms

MPC83xx
- MPC83xx suspend
- e500v2 sleep
- e500v2 deep sleep
- e500v2 JOG

e500v2
- e500v2 idle
- eTSEC wakeup on network
- Compatibility of legacy drivers

e500mc/e5500
- e500mc sleep
- e500mc frequency scaling

PH30 and PCL10 states
- TMU
- T1040 Deep Sleep
- Compatibility of DPAA drivers
- T1040 Auto response

e6500
- Synology on-demand NAS server product
- Cascade Power Management demo
- Benchmarking and optimization

Dynamic power management

Already available: 2012
- Finished
- Proposal
- Planning
- Execution

2013

SDK 1.2
SDK 1.3
SDK 1.4
## Software Distribution: Four Primary Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Approach</th>
<th>When to Use</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Native on Eval Board</strong></td>
<td>Provide evaluation boards with complete native GNU Tool environments right on the board.</td>
<td>Desire zero “getting started” effort to building and running FSL and standard OSS</td>
<td>Easy to use.</td>
</tr>
<tr>
<td><strong>BSP/SDK</strong></td>
<td>This embedded distribution helps customer create entire Linux system. Package both as ISO image and also in virtual machine.</td>
<td>Need a tool to generate a complete Linux environment including tailored file system.</td>
<td>Comprehensive, but very flexible and powerful.</td>
</tr>
<tr>
<td><strong>A la Carte</strong></td>
<td>Simplify customer access to just the major Freescale-created Linux components. Perfect for integration into Linux distributions from other sources, home-brew or 3rd party. Supports fast delivery of patches.</td>
<td>Desire to integrate Freescale Linux components into a Linux development environment that the customer already has.</td>
<td>Simple when the customer is also the integrator.</td>
</tr>
<tr>
<td><strong>Opensource</strong></td>
<td>Committed all the patches to opensource community and push to get them upstreamed. Users directly use the opensource to build.</td>
<td>Desire to use software purely from the community</td>
<td>Simple and flexible</td>
</tr>
</tbody>
</table>
Linux is an open-source integration of many components from many sources, most of which are architecture-independent and don’t originate with Freescale.

Customers (and FSL internally) cannot use Linux without a complete kit.

- Many distros exist.
- Some customers create their own.
- Major FSL SW must be usable with arbitrary distros.
- But FSL also must use and ship one.
- FSL choice: BSP/SDK
Freescale Linux SDK/BSP for Power Platforms

What is an SKD/BSP?
- Software Development Kit
- Board Support Package

What does a BSP include?
- Boot loader
- Kernel
- ToolChain
- File System
- RAM disk
- NFS
- Hard disk
- Applications
- Deployment mechanism
- Documentation

Where can you get BSPs?
- External users
  http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=SDKLINUXDPAA
  l&nodeId=0152100332BF69