Coherent Accelerator Processor Proxy (CAPI) on POWER8
Session objectives

This session provides a technical overview and market discussions for the Coherent Accelerator Processor Interface, or “CAPI.”

The audience should learn:

• How CAPI works in a Power8 system
• Market Opportunities for CAPI Solutions
• Details on IBM’s first CAPI-App: IBM Data Engine for NoSQL
Agenda

• Business Motivation for CAPI
• What is CAPI?
• CAPI Eco-System
• CAPI Flash NoSQL Solution
Heterogeneous Computing is Here

We get the utmost performance when we innovate across the entire system.

Microprocessor measurements must be strong.

But the highest levels of system performance requires strength in all parts.

CAPI is an open platform for innovative solutions to assist in processing data....in order to achieve higher system performance.

....and we have dozens of OpenPOWER Foundation Partners with whom we are innovating.
CAPI impact on Business IT

Higher Performance

Faster time to Value

Lower IT Costs

40x Code Reduction

24:1 Less Infrastructure
What is CAPI?

• CAPI is an innovative method to add a processing engine to a POWER8 system

  – The new processing engine can be configured to do…
    ….ANYTHING!
  
  • A Specialty Engine such as MonteCarlo, Vision Processing
  • An Extended Storage or Flash Device
  • Or Edge of Network Processing

Customize a processor to fit your business need
How CAPI Works

CAPI Developer Kit Card

Acceleration Portion:
Data or Compute Intensive, Storage or External I/O

Sharing the same memory space
Accelerator is a peer to POWER8 Core

Application Portion:
Data Set-up, Control

POWER8 Processor
The CAPI Innovation

CAPI Developer Kit Card

- FPGA
  • IBM Supplied PSL
- PCIe
  • CAPP
- Memory (Coherent)
- PowerP C Core
- OS
- POWER8 Processor

Proprietary Hardware to enable CAPI

Operating System Enablement
Ubuntu LE Kernel Extensions
libcxl function calls

CAPI and PSL - Heart of CAPI Innovation
• Coherent Accelerator Processor Proxy (CAPP)
  • Maintains directory of cache lines held by Accelerator
  • Snoops PowerBus on behalf of Accelerator
• Power Service Layer (PSL)
  • Performs Address Translations
  • Maintains Cache
  • Simple, but powerful interface to the Accelerator unit
CAPI Technology Connections

- Application sets up data and calls Accelerator Functional Unit (AFU)

- Accelerator Functional Unit (AFU) reads/writes coherent data across PCIE and communicates with Application
  - PSL Cache holds coherent data for quick AFU access
CAPI Solution Flow

1. Connect to accelerator
   - Open device `cxl_afu_open_dev`

2. Set Work Element Descriptor (WED) at AddrX – may contain addresses of other data structures
   - Understands WED content and any other addressed data structures

3. Start accelerator
   - Attach device `cxl_afu_attach`
   - Reset AFU
     - `PSL_WED_Ax` is set to AddrX
     - `AFU_CNTL_An[E]` is set
     - `jea` gets AddrX
     - `jcom` gets start

4. AFU continues to work using this interface
   - CMD interface
   - Buffer interface
   - Resp interface
   - MMIO interface
   - CTL interface

5. If required, App can read or write AFU registers

6. App knows AFU is finished (Mechanism is user defined)
   - Assert DONE
   - App can start again from top or free AFU
   - Free device `cxl_afu_free`

AFU reserved for work

AFU continues to work using this interface

AFU fetches AddrX (the WED) starts operation

AFU continues to work using this interface

AFU finishes (Mechanism is user defined)
   - De-assert RUNNING
   - Assert DONE
<table>
<thead>
<tr>
<th>IBM Innovation</th>
<th>Customer Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA is a peer to the processor -- Caching and translations by PSL</td>
<td>Simple Programming paradigm Higher performance</td>
</tr>
<tr>
<td>Architecture allows for any kind of FPGA or even an ASIC</td>
<td>Flexible solutions Connection to Flash, FC, EN…….</td>
</tr>
<tr>
<td>Virtualization in the Architecture</td>
<td>Applications can share Accelerator</td>
</tr>
</tbody>
</table>

**CAPI Differentiation**

**CAPI vs. I/O or Socket FPGA Solution**

- **I/O Paradigm**
  - Virtualization Enablement
  - Lower Performance
  - Programming Complexity
  - Device Driver Overhead
  - FPGA Design

- **CAPI Paradigm**
  - Programming Simplicity
  - FPGA Design
How does CAPI Work?

CAPI vs. I/O Device Driver: Data Prep

Typical I/O Model Flow: Total ~13µs for data prep

- DD Call
- Copy or Pin Source Data
- MMIO Notify Accelerator
- Acceleration
- Poll / Interrupt Completion
- Copy or Unpin Result Data
- Ret. From DD Completion

- 300 Instructions: 7.9µs
- 10,000 Instructions: Application Dependent, but Equal to below
- 3,000 Instructions: 4.9µs
- 1,000 Instructions: 4.9µs

Flow with a Coherent Model: Total 0.36µs

- Shared Mem. Notify Accelerator
- Acceleration
- Shared Memory Completion

- 400 Instructions: 0.3µs
- Application Dependent, but Equal to above
- 100 Instructions: 0.06µs
Two Paths to the CAPI Ecosystem

CAPI Developer Kit

Clients create their own, proprietary business solution.

CAPI Market Solutions

IBM & Partners create business solutions for the CAPI Market. Clients buy pre-packaged solutions from the CAPI Market.

CAPI App Solutions

IBM Data Engine for NoSQL

© Copyright IBM Corporation 2014
CAPI Availability

• See: http://www.ibm.com/support/customercare/sas/f/capi/home.html

• First CAPI Solution:
  IBM Data Engine for NoSQL
  – Procure through IBM (see next page)
  – November 21, 2014

• CAPI Developer Kit
  – Procure through Nallatech
  – Link from IBM Developer Works
    – Requires POWER8 Server
    – October 31, 2014
Potential Markets for CAPI

- **Edge of Network; JPEG & Video processing**
  - Network Packet Processing
  - Database Acceleration/KVS
  - Machine Learning
  - Bitwise Data Manipulation
  - Compression/Encryption

- **Data Analytics Pattern Recognition**
  - Big Data/Database/Compute
  - Social/Media

- **Database Acceleration & Fast Storage**
  - Medicine
  - Radiation Therapy
  - Pharmaceuticals
  - Public Health Image Analysis
  - Genomics

- **Weather**
  - Ensemble Calculations of Numerical Weather Prediction

- **Oil & Gas**
  - Reverse Time Migration

- **Manufacturing/EDA**
  - Fluid Dynamics
  - 3D Modeling CAD
  - Pipeline Analysis & Flow

- **Visual / Biometric Analysis**
  - Retail Security
  - Facial Recognition

- **Finance/Insurance**
  - Risk Analysis
  - Monte Carlo
  - Pattern Analysis

- **Deep Computation and Critical Runtime Jobs**

- **Specialized Algorithms**
IBM Data Engine for NoSQL

- Attach TMS Flash to POWER8 via CAPI coherent Attach
- Issues Read/Write Commands from applications to eliminate 97% of code pathlength

► Saves 20-30 cores per 1M IOPs
Innovative “In-Memory” NoSQL/KVS Integrated Solution

IBM Data Engine for NoSQL

Today’s NoSQL in memory (x86)

- Load Balancer
- 512GB Cache Node
- Backup Nodes

4Q14 - Differentiated NoSQL (POWER8 + CAPI Flash solution)

- POWER8 Server
- Flash Array w/ up to 40TB

Target LoB / solution architects and MSPs

- Supporting or building mobile/web/social apps
- Leveraging Key Value Store (KVS) for fast lookups
- Require high performance in-memory data access

Power + CAPI Flash Advantage

- 24:1 physical server consolidation
- 6x less rack space (2U server + 2U Flash vs. 24 1U servers)

- Regain infrastructure control
- Dramatically reduce costs to deliver services

24:1 Reduction in infrastructure
2.4x Price reduction
12x Less Energy
6x Less rack space
40TB of extended memory

© Copyright IBM Corporation 2014
Demonstrating the Value of CAPI

Attachment

IBM Data Engine for NoSQL

Identical hardware with 2 different paths to data

Flash System 840

Conventional PCIe I/O

CAPI

Power S822

IBM Data Engine for NoSQL

IOPs per HW Thread

PCIe I/O  CAPI

108,438

16,150

199

Latency (us)

PCIe I/O  CAPI

466

© Copyright IBM Corporation 2014
Monte Carlo Simulations

- **Medicine:**
  - Monte Carlo simulations are used in leading edge cancer treatment for radiology
  - Current simulation runtimes for an individual is multiple weeks

- **Financial:**
  - Monte Carlo methods are used to price complex financial derivatives which in turn allows financial institutions determine the risk (e.g. VaR) of their investment portfolios
  - Required by regulators of risk compliance
Monte Carlo on CAPI

Running 1 million iterations
At least 250x Faster with CAPI FPGA + POWER8 core

Reduced C code 40x compared to non-CAPI FPGA