OpenPOWER& Kernel Features

IBM Liang ZHUANG/Jia HE

2015/10

OpenPOWER – Why, What, How

IBM Liang ZHUANG
2015/10

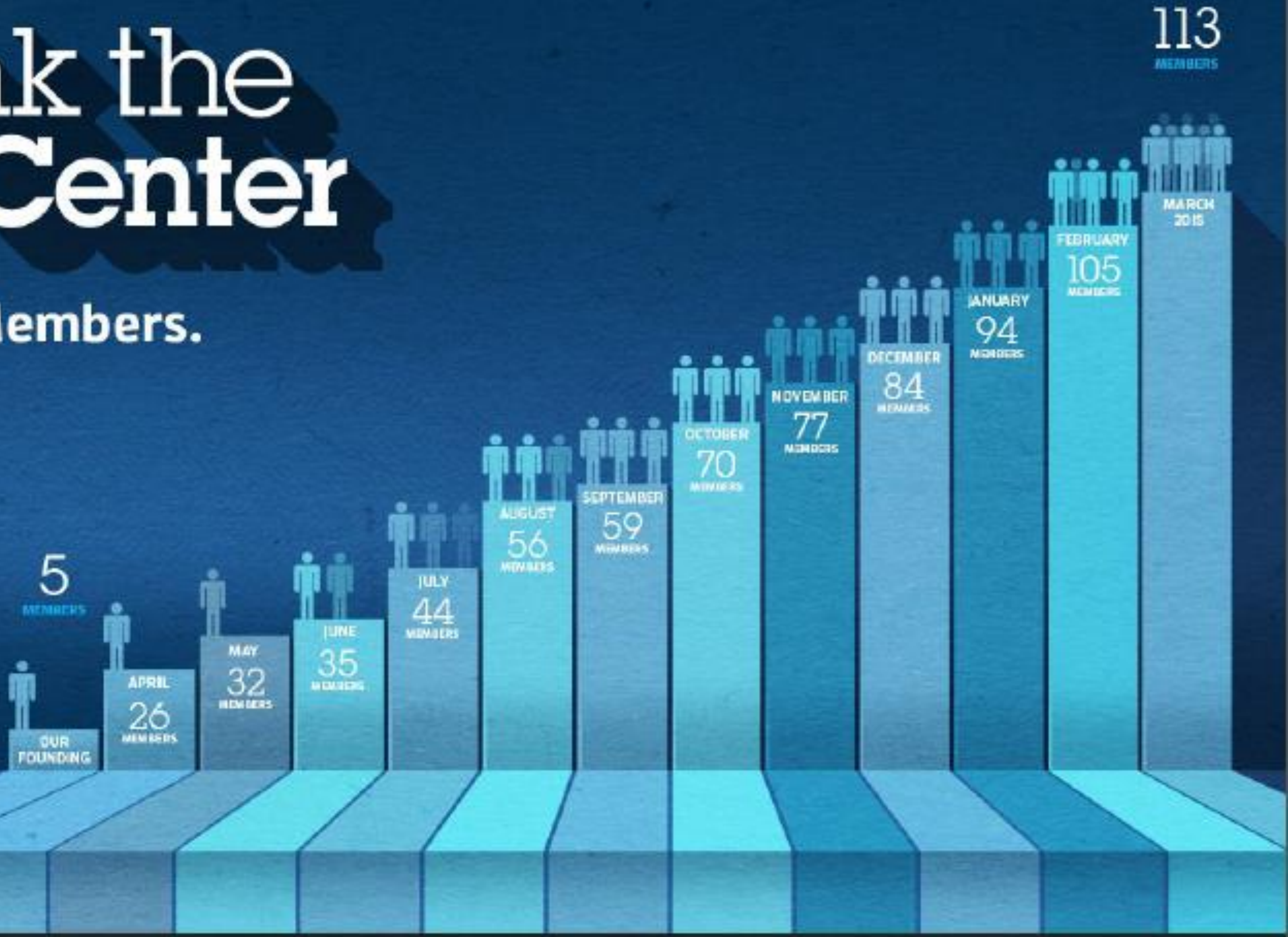
OpenPOWER

Rethink the Data Center

One Year. 113 Members.

Join the revolution at openpowerfoundation.org

#OPENPOWERSUMMIT



The Shared Vision

To create an open development community dedicated to transforming the approach to scale out server design based on the POWER architecture.

Rethink the Data Center

The End of Moore's Law

Cost-performance benefits are diminishing

Improve integration of CPU with I/O and Accelerator subsystems

Maximum impact through collaboration with technology leaders

Open Processor Architecture with Scale-Out Performance

Proposed Work Groups and Projects

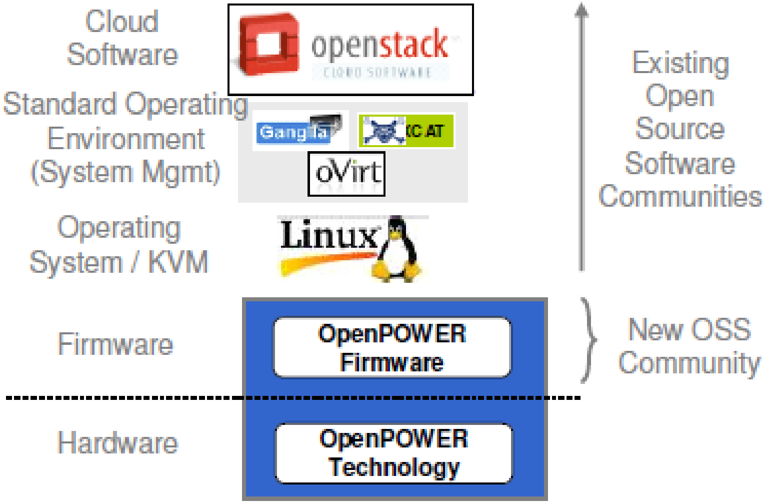
Work Group	Projects	Participants
System Software (Open Source)	• Linux LE	Public
	• KVM	Public
	• Firmware –OpenPOWER FW interface	Public
	• POWER LE ABI	Public
Application Software (Open Source)	• System Operating Environment – OpenPOWER Software ecosystem enablement	Public
	• Toolchain	Public
Open Server Development Platform	• Power 8 Developer Board	Member
	• POWER 8 Reference Design	Member
Hardware Architecture	• OpenPOWER profile of architecture – Power8 ISA Book 1, 2, 3	Member
	• Coherent Accelerator Interface Architecture (CAIA)	Member
Compliance	• Compliance	Member

Proposed Ecosystem Enablement

OpenPOWER™

Software

Power Open Source Software Stack Components



System Operating Environment Software Stack

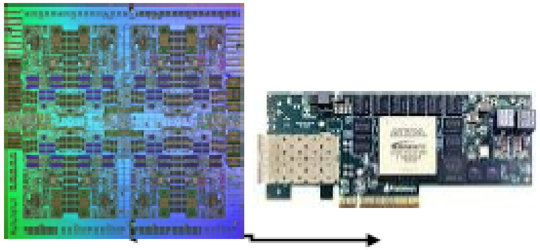
A modern development environment is emerging based on tools and services

Bootstrap
node
SOCKET IO
redis
Cassandra
RabbitMQ
NGINX
mongoDB
MariaDB
HA PROXY

Hardware

Multiple Options to Design with POWER Technology Within OpenPOWER

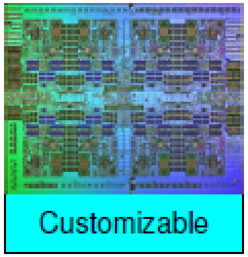
POWER8



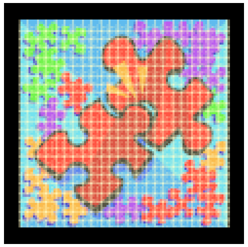
CAPI over PCIe

"Standard POWER Products" – 2014

Framework to Integrate System IP on Chip



Industry IP License Model



"Custom POWER SoC" – Future

Open Source Ecosystem

Dev. Env /Tools

Available:
Backbone, Bootstrap, Docker , Eigen lib
Erlang, Ganglia, GCC, gccgo, GDB, Jenkins, Jruby, LLVM, logstash, logstash-forwarder
Maven, Nagios, NGINX, node.js, OpenJDK, PHP, phpMy Admin, Perl, Python, Python-Django, Python-Pip ecosystem, R/R, RabitMQ, rsyslog, Ruby, Ruby on Rails (rbenv), Ruby Gems, scala, snappy, Socket.io (npmjs) Supervisord, SpiderMonkey, SystemTap, Vagrant, V8, wireshark, Xerces

Port In Progress:
Apache Gump, GoLang, kibana
Pubsub.io (3Q)
Phantom.js,

Evaluating:
Cloudbees
OpenJDK - optimize
Ruby - optimize

Databases

Available:
Accumulo (column), Cassandra
CouchDB (document)
Derby
MariaDB (v10 optimized)
Memcached (KVS)
MongoDB (document)
MySQL
PostgreSQL
RabbitMQ
Redis (KVS), Riak
SQLite
TokyoCabinet
Virtuoso (graph)

Port In Progress:
Voldemort (KVS)
Neo4J (graph)
MongoDB 3.0

Optimizing:
PostgresSQL (1.86x)
CouchDB

Evaluating:
Couchbase (noSQL)
InfiniSQL
MarkLogic(document, ISV)
OrientDB

Big Data & Analytics

Available:
Hadoop Core, Hive, HBase, Accumulo, Ambari, Avro, ElasticSearch
Falcon, Flume, Hue, Knox, Lucene-Solr, Mahout, Oozie, Parquet, Phoenix
Pig, Riak, Sqoop, Storm Tez, Zookeeper

Port In Progress:
Spark

Optimizing:
Hadoop (3Q15)

Evaluating:
Clusterpoint

Cloud Managemnt Stack

Available:
Apache Web Server
Apache tomcat
Ceph, Chef server
Jetty,
Juju & Juju gui
Landscape client
MAAS, OpenStack
Puppet
Apache Qpid
Thrift
Ceilometer client, Sensu Server & Client

Port In Progress:
Glassfish

Other HA, Security etc.

Available:
BTRFS, Bootstrap
Chroma-key
Cluster Glue, corosync, DRBD
Evolution data svr
HAProxy, Heartbeat
keepalived
Ldirectord,
Linux-HA, mesos
OpenSSL
Pacemaker, REAR
Samba, Tophat
WordPress

Port In Progress:
CentOS

Evaluating:
Cluster-Network
CoreOS (distro)
MondoRescue
Open Identity Stack (forgerock.com)

Technical Computing

Available:
ALLPATH-LG, Bedtools, bfast, BioConductor, BioConductor-base,
BLAST, BOOST, Bowtie, Bowtie2, BWA, bzip2, Cufflinks-2.2.1, FASTA, FastQC, HMMER, HTSeq, IGV, iRODS (beta), ISAAC, LibGD(partial), libpng, Mothur, nose, NumPy, OpenSSL, PICARD, PLINK, Python, RNAStar, SAMTools, SAMTools 1.0, SeqAn, setuptools(Python), SHRiMP, SOAPAligner, SOAP3-DP, SOAPDenovo, tabix, TMAP, TopHat, Trinity, Velvet/Oases, Zlib, ABySS, Balsa, Bioconductor, GATK, GMP, Google Double-Conversion, GROMACS, NAMD, Quantum Espresso, spice

Port in Progress:
iRODS,

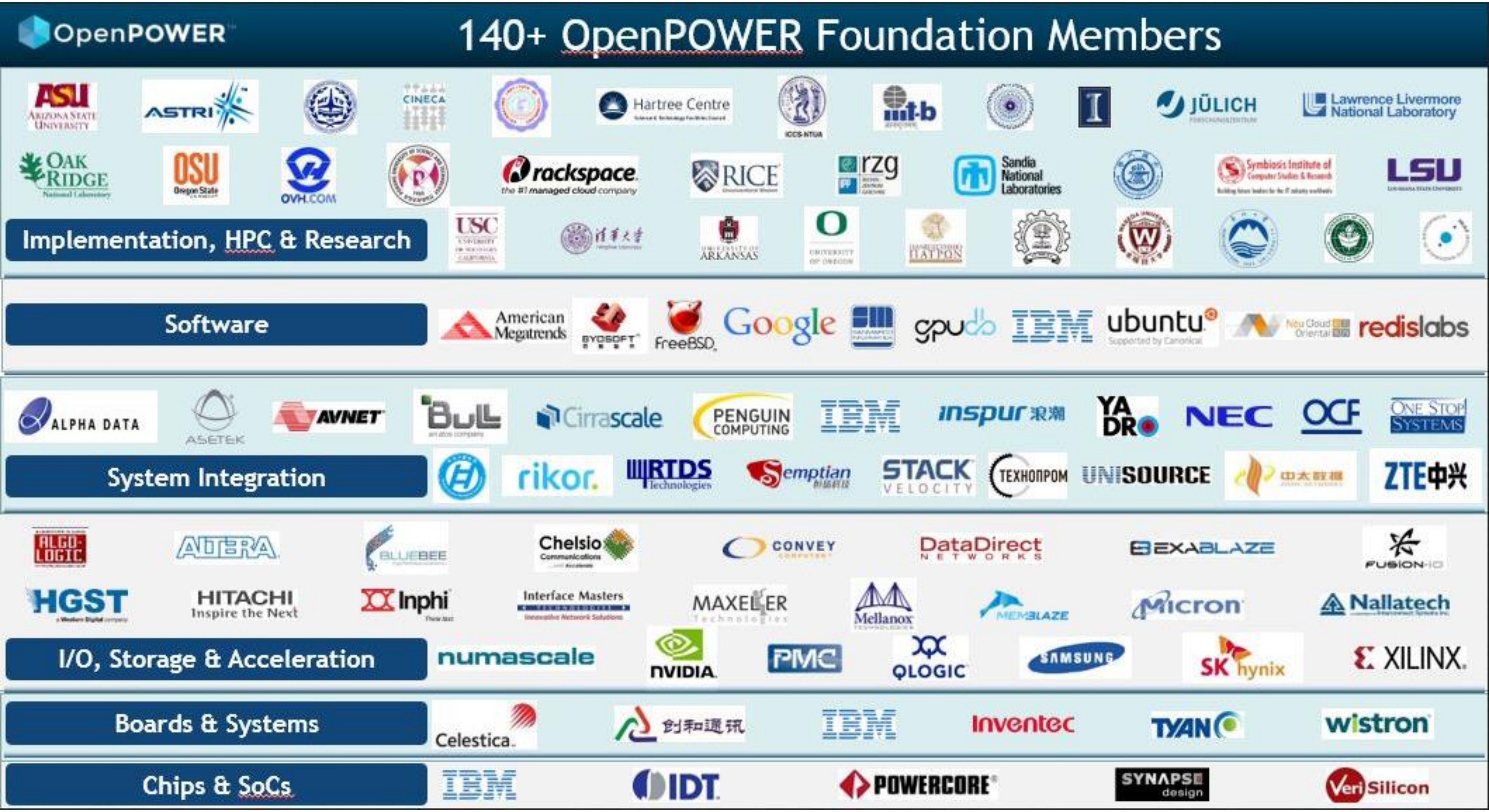
Optimizing:
NAMD, GROMACS, ABySS

Evaluating:
CP2K, HOOMD, miRdeep2, Galaxy, Terachem (ISV), ucsctools, ViennaRNA, AMBER14 (plan 2015)

Available: Open source application is ported and available on distro (Ubuntu or RHEL or SLES) (black), in community (purple), Lab7 (green) or Veristorm (orange). Does not mean it is optimized. Does not mean that a commercial ISV version is available.

Evaluating: Needs to be vetted in new business development prioritization process. Some of these are available codes that need optimization to be competitive with x86.

Foundation Members (147 members over 22 countries, 2015/8)



OpenPOWER

140+ OpenPOWER Foundation Members

Implementation, HPC & Research

ASU Arizona State University
ASTRI
CINECA
Hartree Centre
ICCS-NTUA
iit-b
JÜLICH
Lawrence Livermore National Laboratory
Oak Ridge National Laboratory
OSU Oregon State University
OVH.COM
rackspace the #1 managed cloud company
RICE
rzg
Sandia National Laboratories
Symbiosis Institute of Computer Studies & Research
LSU
USC University of Southern California
University of Arkansas
O University of Oregon
ΠΑΤΡΩΝ
W

Software

System Integration

I/O, Storage & Acceleration

Boards & Systems

Chips & SoCs

Kernel New Features

on OpenPOWER POWER8 CPU

IBM Jia He 2015/10

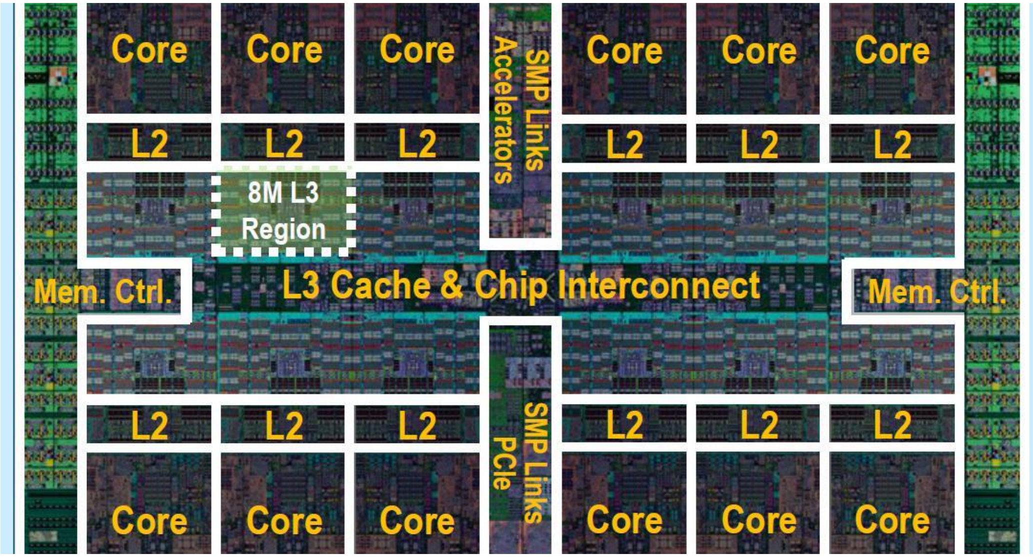
Agenda

- OpenPOWER CPU (Power8)
  - Hardware Highlights
  - Linux Distro Support Status
- Kernel-related Software Enhancement
  - BE/LE
  - Hardware acceleration/crypto
  - Parallel programming
  - Energy management
  - Java Performance enhancements

OpenPOWER CPU (Power8) - Hardware Highlights

Cores

- **12 cores** (SMT8) **96 threads** per chip
- 8 dispatch, 10 issue
- 16 execution pipes



Caches

- 64K data cache, 32K instruction cache
- **512 KB L2 /core** (SRAM)
- **96 MB L3** (eDRAM shared)
- Up to **128 MB L4** (eDRAM, off-chip)

Accelerators

- Crypto & memory expansion
- Transactional Memory
- **Coherent Accelerator Processor Interface (CAPI)**

Memory

- Up to **230 GB/s** sustained bandwidth

New instructions

- Quadword new atomic instructions
- Vmx. vsx

Linux Distro for OpenPOWER – Guest Status

Linux	Release (start from)	Endian	KVM guest
Redhat	7.0	Big	✔
Redhat	7.1	Little/Big	✔★
SUSE	12	Little/Big	✔
Ubuntu	14.04	Little/Big	✔

✔	Supported
✖	Not Supported
★	Firmware 8.30 or higher

Linux Distro for OpenPOWER – Bare Metal& Host Status

Linux	Release (start from)	Endian	Bare Metal	KVM Host	Comment
Redhat	7.2	Little	✔	✔*	*Only in RHEV firstly
SUSE	12SP1	Little	✔	✔	
Ubuntu	14.04	Little	✔	✔	First supported distro
PowerKVM*	2.1	BE		✔	Guest BE/LE
PowerKVM	3.1	LE		✔	Guest BE/LE

* PowerKVM™ is IBM's hypervisor distro.

Summary

All major distros are supporting in LE mode, whatever BML, Host or Guest.
（所有主流发行版，都支持LE mode工作在OpenPOWER CPU上）

Kernel-related Software Enhancement
for OpenPOWER P8 CPU

- **High performance BE/LE support（大小端问题）**
- **Hardware Acceleration/Crypto (硬件加速）**
  - CAPI(Coherent Accelerator Processor Interface)
  - Hardware acceleration/Encryption
- **Parallel Programming Productivity（并行计算）**
  - SMT/Split Core
  - Hardware Transactional Memory
- **Energy Management（能耗）**
- **Java Performance Enhancements（JAVA性能）**

High performance BE/LE support

- **Why do we need to support both Big/Little Endian?**
  - A new eco-system
  - Easy for applications to migrate to powerpc platform
  - Driver codes integration between ppc64le and other le arch

- **How to?**
  - Firmware
  - Kernel 3.13 (ubuntu 14.04’s kernel)
  - Designed a new (ELF V2) ABI to simplify and improve performance
  - Advance Toolchain 7.0 for BE，7.1 for LE
  - Gcc 4.8
  - Glibc 2.17

硬件加速 – CAPI Use Case

- **Use Case**
  - Hadoop needs 3 data copy
  - More disks = more costs in server
  - Erasure code algorithm = cpu usage 99.9%

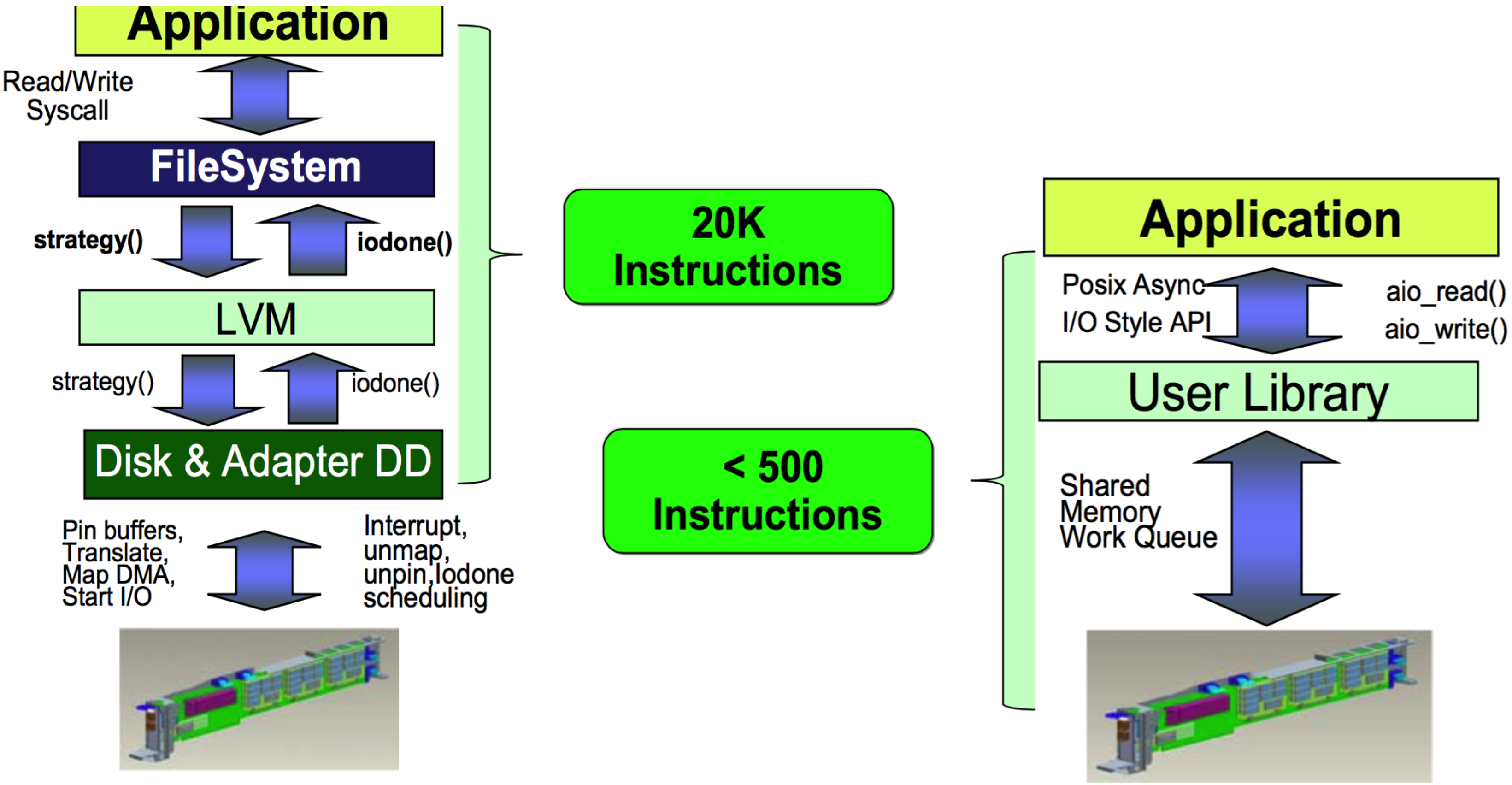
- **Solution: CAPI FPGA accelerator over PCIe**
  - Coherent Accelerator Processor Interface
  - No changes on server hardware configuration
  - No additional rack space, no more AC power support,

- **Benefit**
  - Disk number decreased 50%, server cost decreased 30%
  - 1 CAPI card equals10-20 CPU cores’ compute capacity
  - Cost performance improved 300-400%

CAPI – Why it rocks!

传统方式

CAPI方式



Application

Read/Write
Syscall

FileSystem

strategy()

iodone()

LVM

strategy()

iodone()

Disk & Adapter DD

Pin buffers,
Translate,
Map DMA,
Start I/O

Interrupt,
unmap,
unpin,Iodone
scheduling

20K
Instructions

< 500
Instructions

Application

Posix Async
I/O Style API

aio_read()
aio_write()

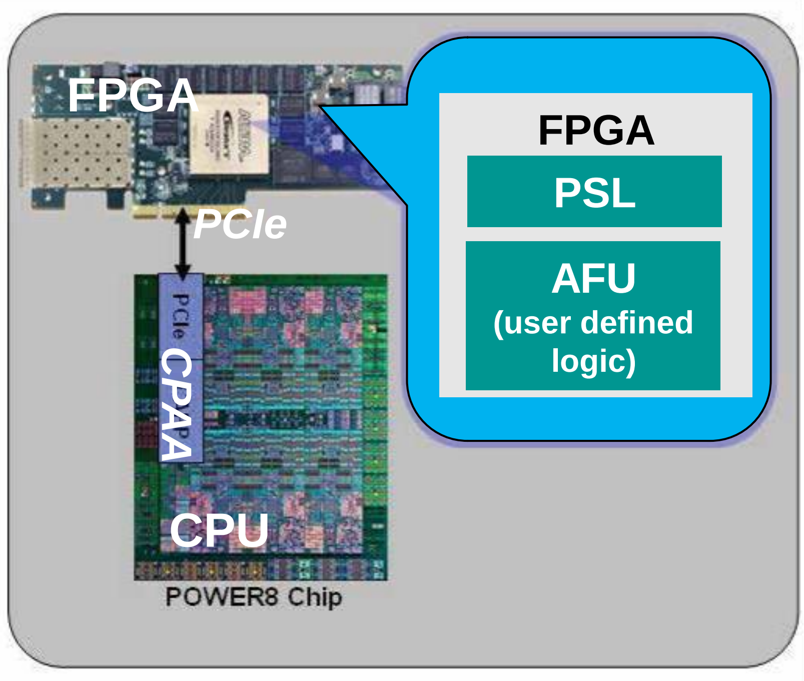
User Library

Shared
Memory
Work Queue

Attach flash memory to POWER8 via CAPI coherent Attach

CAPI – HW Components

- **CAPP** - Coherent Accelerator Processor Proxy
  - Maintains directory of cache lines held by Accelerator
  - Snoops PowerBus on behalf of Accelerator
- **PSL** - Power Service Layer
  - Performs Address Translations
  - Maintains Cache coherence
- **AFU** - FPGA chip self defined by vendors



Benefits

- Accelerator can work with same memory addresses that the processors use
- Pointers de-referenced same as the host application
- Removes OS & device driver overhead

CAPI – Software Components

- CAPI support in kernel
  - Regard CAPI device as a PCIe device, match/init/configure
  - Maintenance, system protection, and communication functions.
- Application library
  - Libcxl

Hardware Acceleration/Encryption

- **Nest Accelerator unit (NX)**
  Comprises cryptographic and memory compression/decompression engines (co-processors) with support hardware.
  - AES (Advanced Encryption Standard) engine
  - SHA (Secure Hash Algorithm) engine
  - RNG (Random Number Generator)
  - 842 Compression/Decompression
    - IBM-proprietary algorithm
    - Performance shows >300% performance improvement with eCryptfs
- Open for OpenPower
- Kernel support
  - Driver for nx842
  - crypto API
  - zswap, dm-crypt, eCryptfs, IPsec
- Benefits for applications
  - userspace support in libica.
  - Openssl
  - Java, new crypto instructions

SMT(Simultaneous Multithreading)/Split core

- Virtualization benefits from smt/ split core
- SMT = 8 (8 threads/core) is enough?

  Need context switch if

  guests number > core number, because mmu is shared in one core.
- Split core = 2, 4

  => 2 or 4 guests can run in 1 core at the same time
- Dynamic Split Core

HTM (Hardware Transactional Memory)

- **Motivation: to scale out**
  - Hardware managed atomicity to shared data
  - Light weight locking
  - Parallel thread concurrency
- **Use Cases**
  - Optimistic Execution of Lock-Based Applications
  - Transactional Programming in High-Level Languages
- begin, end, abort, suspend, resume
- **New in power8 for HTM**
  - New instructions mark beginning and end of transaction.
  - Hardware ensures atomicity

HTM

- How to determine potential benefits?
  - If the transactions reference a large amount of data, TM is not helpful.
  - If read-only transactions typically do not reference the same data as concurrent transactions that write data, TM may help.

- Limitations
  - No syscall
  - No reset stack/context
    - getcontext(), setcontext(), makecontext(), swapcontext(), setjmp(), and longjmp()

HTM examples

To enter a critical section (pthread_mutex_lock):

```
if (__TM_begin(tm_buff) == 0) {

  long val = mutex->mt_lock;
  if (val == UL_FREE) {
    /* Free */
    /* Enter critical section using TM */
    return 0;
  }

  /* Busy */
  __TM_abort();
}
else {
  /* Not in a transaction */
  ...

  /* Giving up - Not using TM - Need to acquire lock */
  ... <acquire lock> ...

  /* Enter critical section holding lock - Not using TM */
}
```

To exit critical section (pthread_mutex_unlock):

```
if (__TM_end() == 0) {
  /* Was inside transaction - No need to do anything
*/
  return 0;
}
else {
  /* Must have acquired lock instead of using TM */
  ... <release lock> ...
}
```

HTM examples

```
num_retries = 10 ;
while (1) {
   if (__TM_begin (TM_buff) == 0) {  /* Transaction State Initiated.  */
      if(shmlock.isLocked()) {
         num_retries=0; /*resort to locks*/
         __TM_abort();
      }
      sum = a + b;  //add transaction code here.
      __TM_end ();
      break;
   }
   else {
      /* Transaction Failed.  Use locks if the transaction failure is “persistent” or tried too many times.  */
      if (num_retries-- <= 0 || __TM_is_failure_persistent (TM_buff)) {
         /*resort to conventional lock*/
         while (shmlock.readLock() != 0);
         sum = a + b;
         shmlock.unLock();
         break;
      }
   }

  …….
}
```

Energy Management – Idle states

- Idle states

Name	Description	Power Savings	Exit Overhead
Snooze	Software defined polling state	Low	Low
Nap	Core is clockgated	High	Low
Fastsleep	Voltage to Core and L2 cache is brought to minimum	Higher	High
Winkle	Voltage to Core, L2 and L3 cache is turned off. Enabled only for offline CPUs	Highest	Highest

- /sys/devices/system/cpu/cpuX/cpuidle/stateY
- cpupower idle-set -d <state_number>
- cpupower idle-set -e <state_number>
- cpupower idle-info

Energy Management - DVFS

- Dynamic Voltage and Frequency Scaling/DVFS

Name	Pstate at low load	Pstate at high load
OnDemand	Lowest	Highest
Userspace	Default:Nominal	Default: Nominal
Powersave	Lowest	Lowest
Performance	Highest	Highest

- The higher the pstates, the more performance the CPU gets, but at cost of higher power consumption.
- /sys/devices/system/cpu/cpuX/cpufreq

Energy Management – tuned-adm

- CPU Power Management using tuned-adm
- tuned-adm
  - Balanced profile for workloads which require a fine balance between performance and power savings
  - Latency-performance profile for latency sensitive workloads
  - Throughput-performance profile for workloads that expect steady performance
  - Powers avings profile for workloads that care mostly about power savings

Java Performance enhancements

- IBM Java will transparently provide support for EBB/BHRB profiling during JVM startup to improve JIT code optimization.
- **PMU** (Performance Monitor Unit): New types of counters
- **EBB** (Event-Based Branching)
  - Generates event-based exceptions when a certain event criteria is met. Following an EBB exception, the BESCR register tells which kind of event triggered the exception.
  - Asynchronous userspace interrupt based on events.
  - PMU EBB event, work together with PMU.
  - Signal handler like
- **BHRB** (Branch History Rolling Buffer)
  - Rolling list of recent branches
  - Can be used as a call trace leading up to Performance Monitor interrupt
  - Can be used to detect branch prediction problems
- **DSCR** (Data Stream Control Register)
  - DSCR is a register which controls the prefetching data between RAM and caches
  - Dynamic degree of aggressiveness, per process

Summary

- POWER8 is the first OpenPOWER CPU
- **Open**
  - BE/LE kvm
  - CAPI
  - Hardware accelerator（Encrypt/Compression)
- **Power**
  - Parallel programming: smt/split core, HTM
  - Java Performance enhancement
  - Energy management

Thanks
Q&A