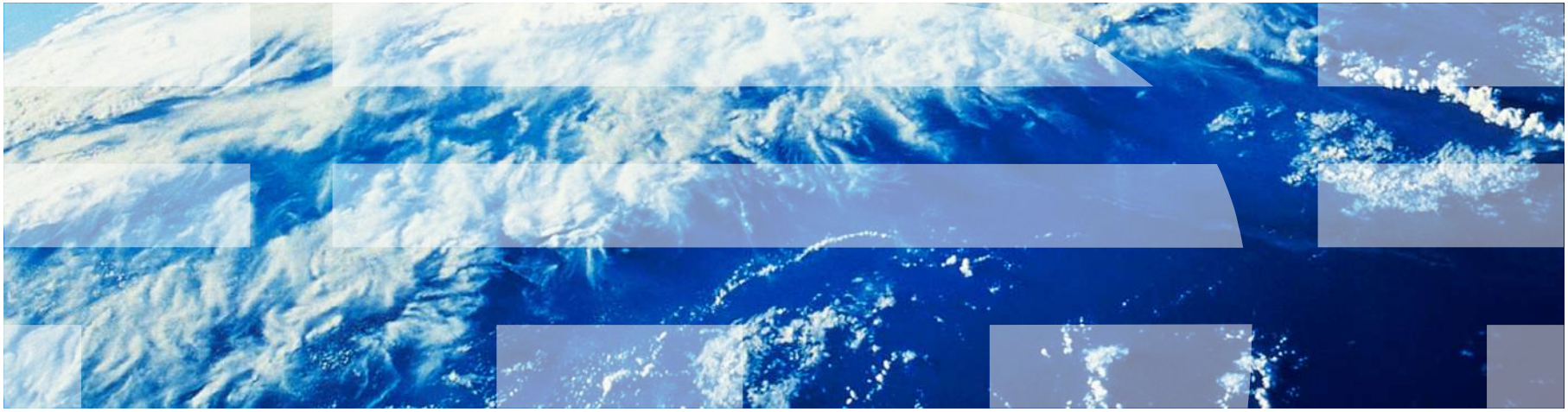
The IBM Power Edge of Network™ Processor:

A novel System-on-a-Chip for Wire Speed computing

Massimiliano Meneghin and Karol Lynch
High Performance Computing Group
IBM, Dublin



Agenda

. What do we mean when we refer to wire speed computing and why do we need to design a new family of computer chips to meet its needs?

. What kind of features does a chip designed for wire speed computing have (i.e. the PowerEN™ Architecture)?

. How does one program the PowerEN™?

. How will IBM deploy PowerEN™?

Why wire speed computing?

Data volumes are rising exponentially
- Network traffic capture, distributed sensor networks, GPS enabled smart phones, etc.

Increased analysis complexity requires more computing power
- Threat analisys, tracking of people and/or veicles, problem determination on utility grids, etc..

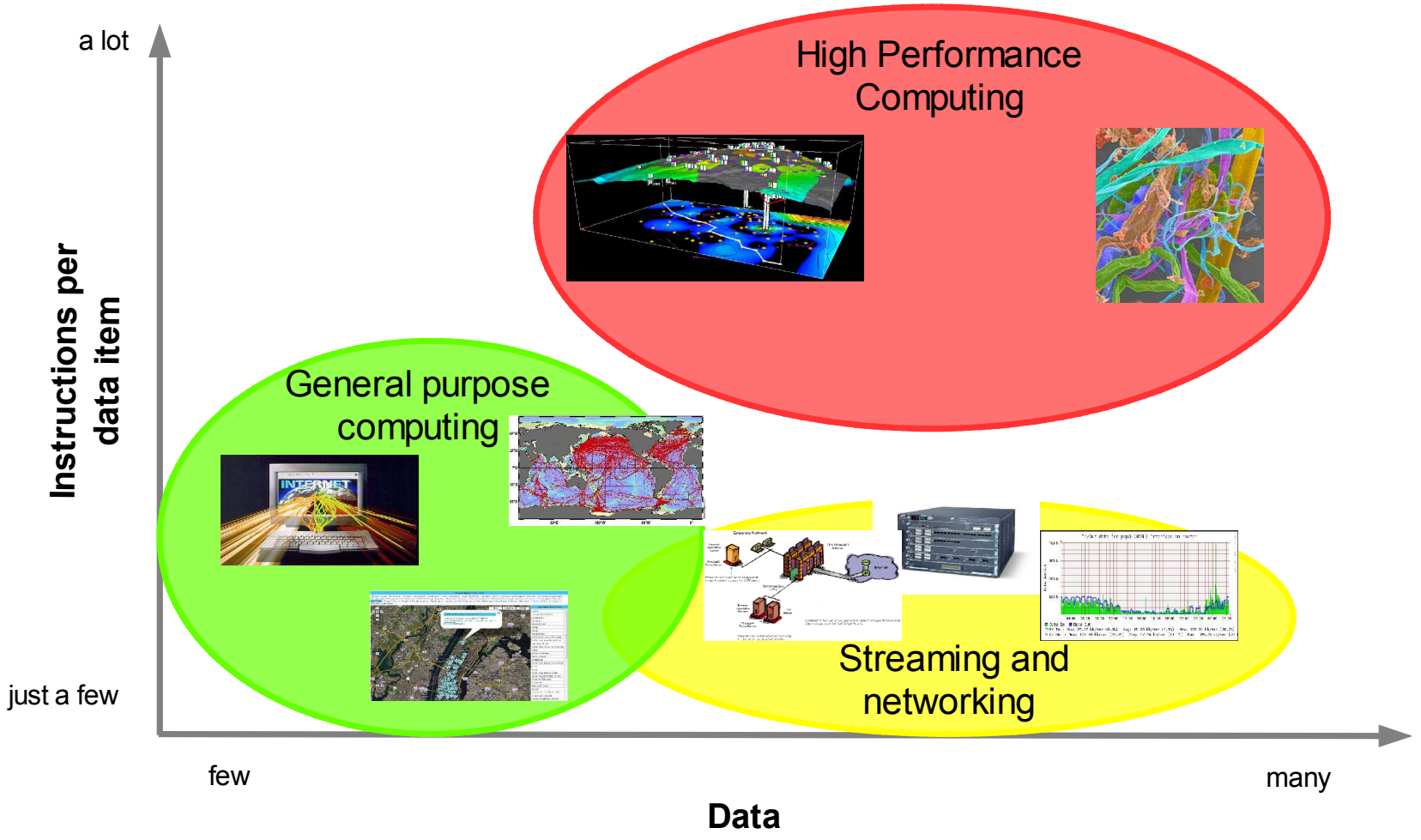
Applications require a fast response time to enhance their Business Value
- Trading markets, threat response, market insights, etc.

We face the challenge of rethinking how we build systems!

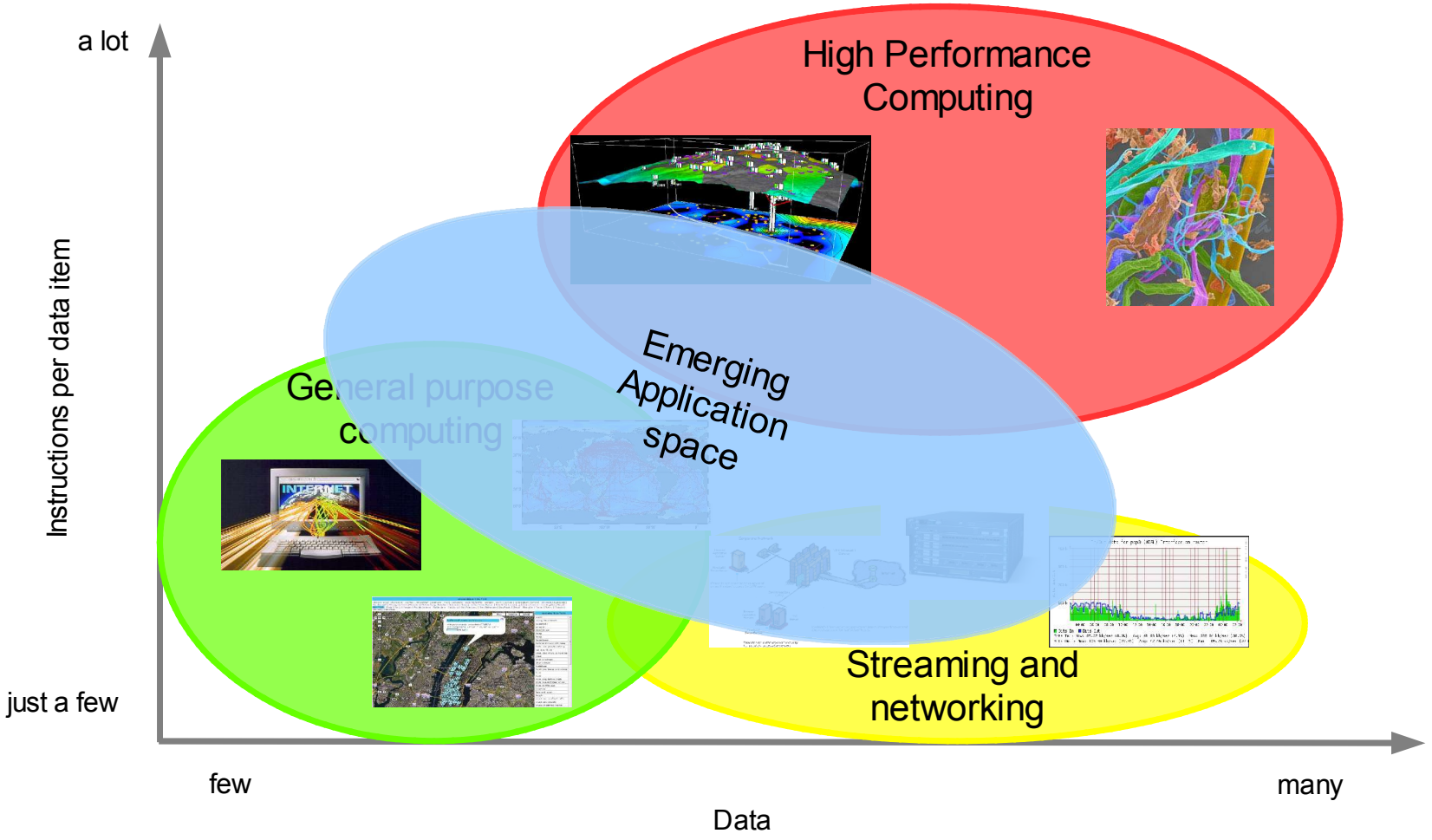
A new philosophy, the Wire-Speed processor project, which defines a generic processor architecture in which

General purpose Cores - HW Accelerators - I/O functions are closely coupled

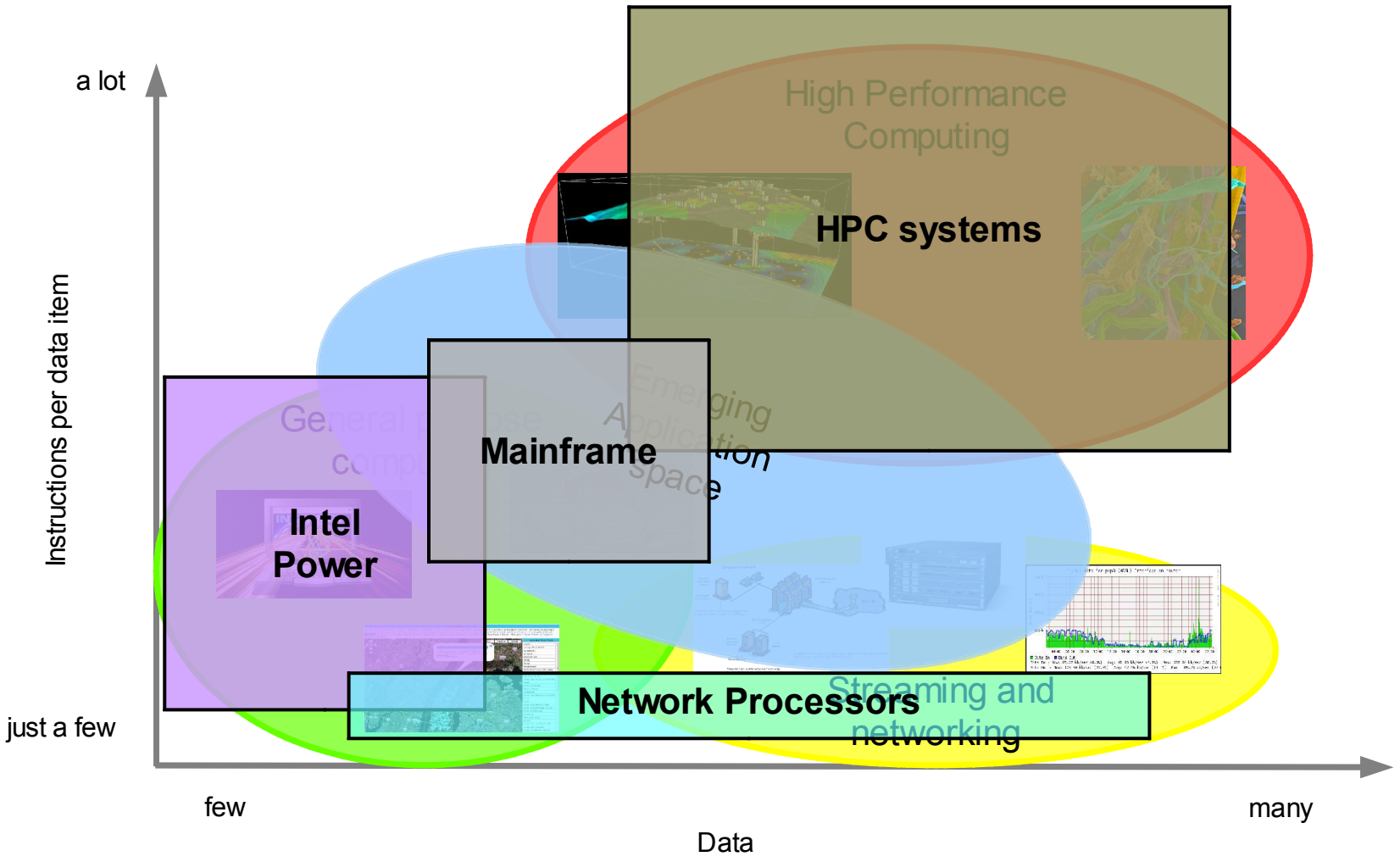
Why Wire Speed Computing?



Why Wire Speed Computing?



Why Wire Speed Computing?



What is Wire Speed Computing?

A blurring of the Network and Server worlds

Highly Multi-threaded low power cores with full PPC ISA.

Standard programming models with OS's and hypervisors

Virtualisation support for application consolidation.

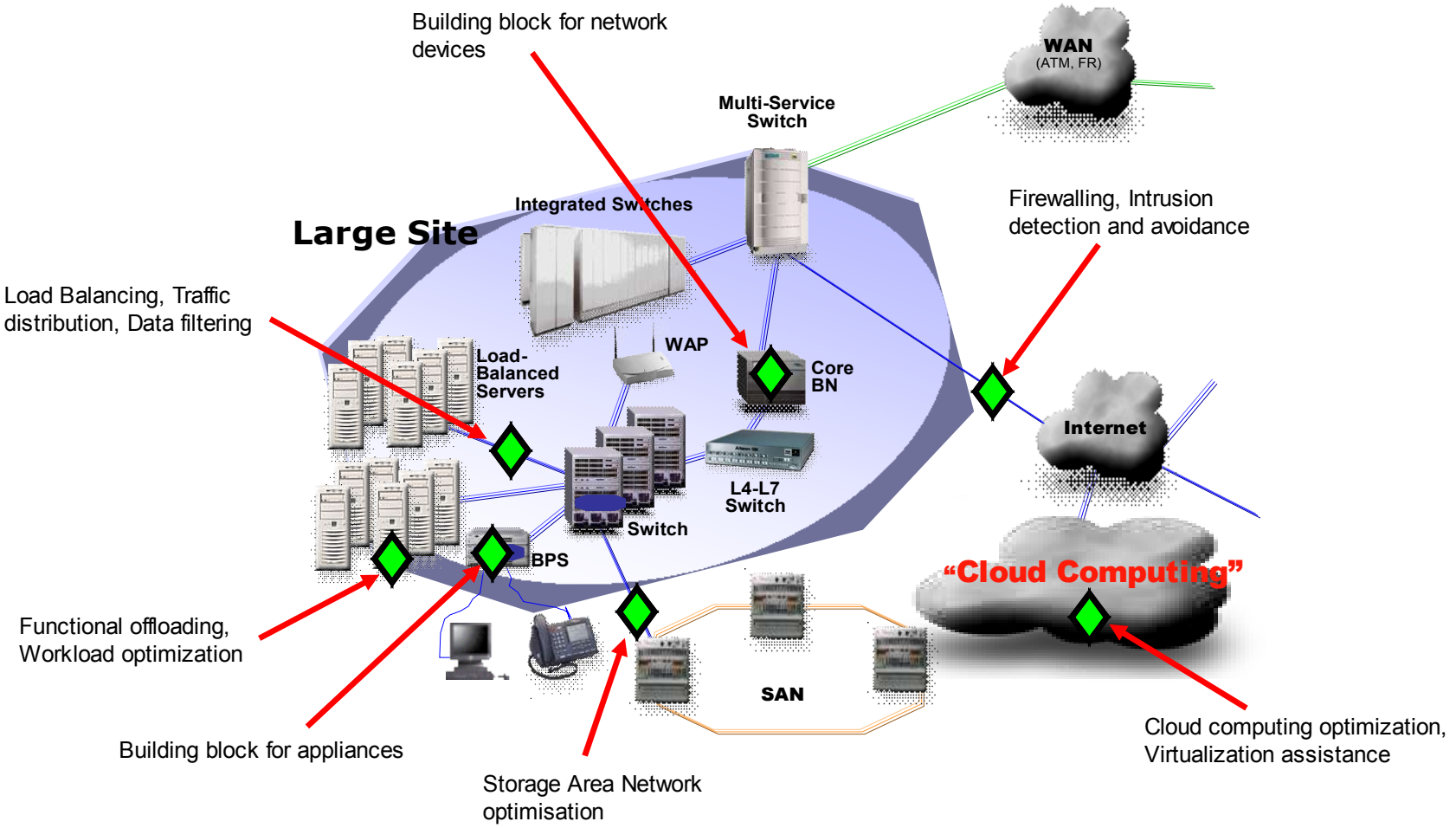
Accelerators: for Networking and Application tiers.

Integrated Network system & Memory I/O.

Server RAS & infrastructure.

Low total power solution based on throughput optimisation

System Platforms



Evolution of Chip Technology: Past (not so long ago...)

Objective: Increase performance from a single chip → Increase Single Thread Performance → Exploitation of: Frequency scaling ILP + out of order execution

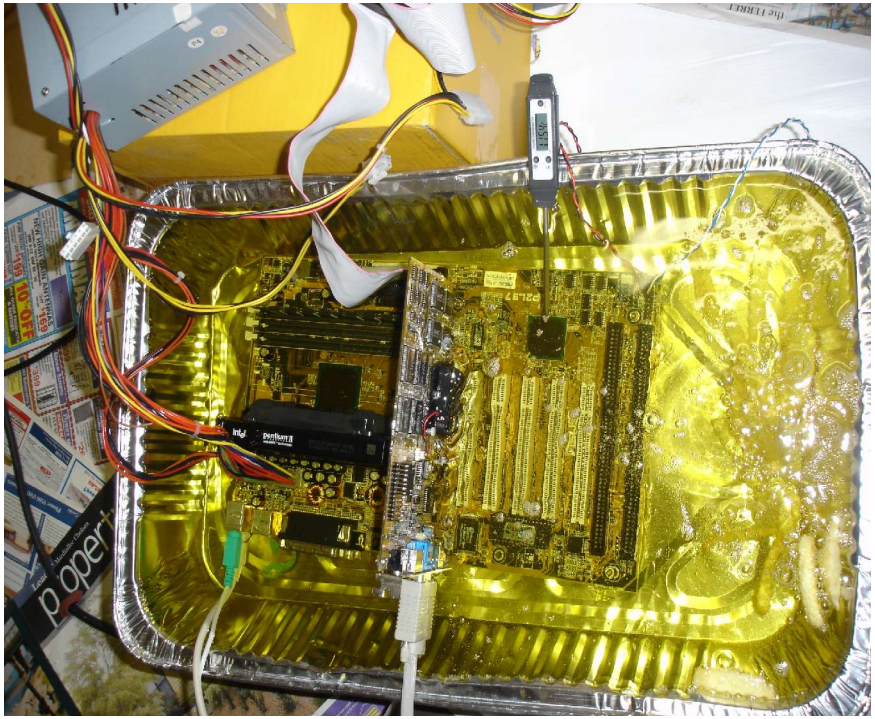
This approach has reached its limit:

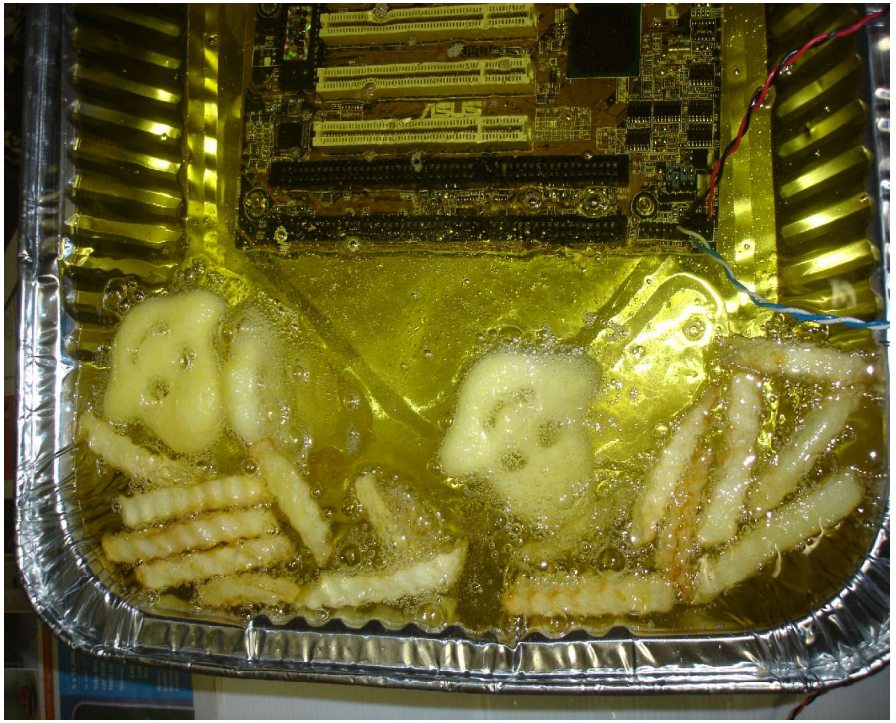
Power Wall

ILP Wall

End for
CMOS technology & Uniprocessor combination

That is unless you like toxicated chips!



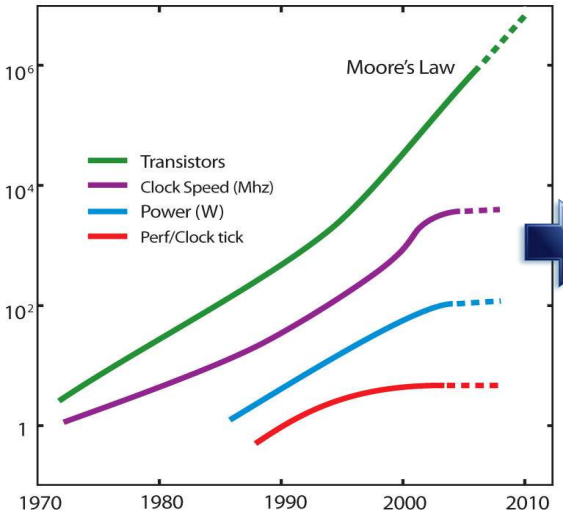


Evolution of Chip Technology: Present

Objective:
Increase performance from a single chip

→ Increase Total Throughput

→ Exploitation of:
Thread-Level Parallelism(TLP)
Moore's law



→ More Transistors Per Chip unit area

→ **CMP arch**:
+ CPUs on the same chip

Few Complex CPUs
(up to 4)
Intel** dual/quad core
IBM Power 6/7

Many Simple CPUs
(up to 9)
IBM Cell B.E
Sun** Niagara

Evolution of Chip Technology: Coming soon

Observation:
Where you would start optimising?

```
while(!done){
...
for(int i=0; i<10; i++){
        Alpha(...);
}
for(int i=0; i<10^8; i++){
        Beta(...);
}...}
```

Some computations are performed more times than others.

The effort of optimising one is rewarded N times

And back to the Silicon?
Off-loading of core cycles to **domain specific** hardware accelerators

Specific HW:
+perf
-power
-flexibility

Final Solution:
System on Chip(SoC):
GP Cores
Specific HW accelerators
I/O functionality
All closely coupled

The challenge...

Maximisation of the power/performance ratio

smartly understand the maximum performance for a certain thermal limit

Maintenance of the system **programmability against heterogenous arch**

a easier life for programmers saves time and money

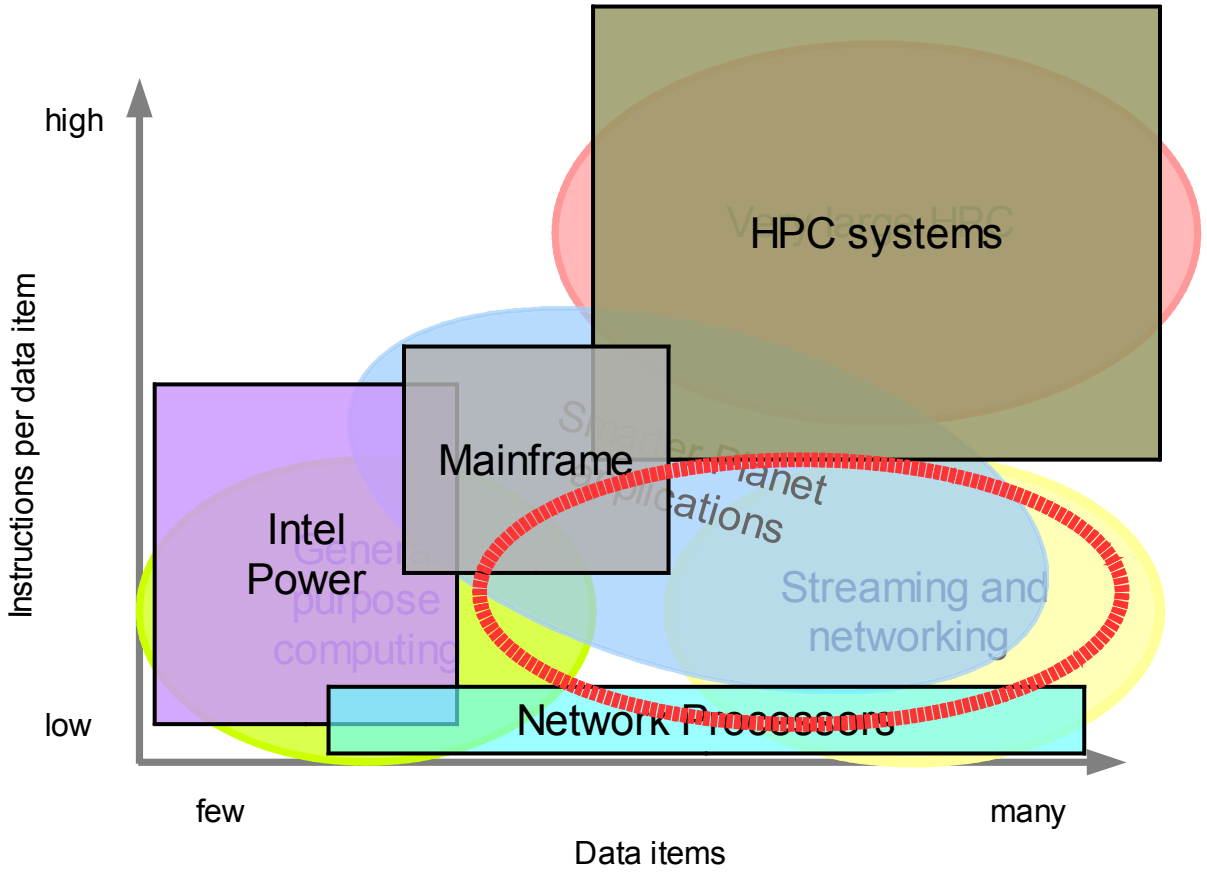
Bypassing architectural levels to **avoid unnecessary overhead**

system call overhead can be critical (Linux packet filtering)

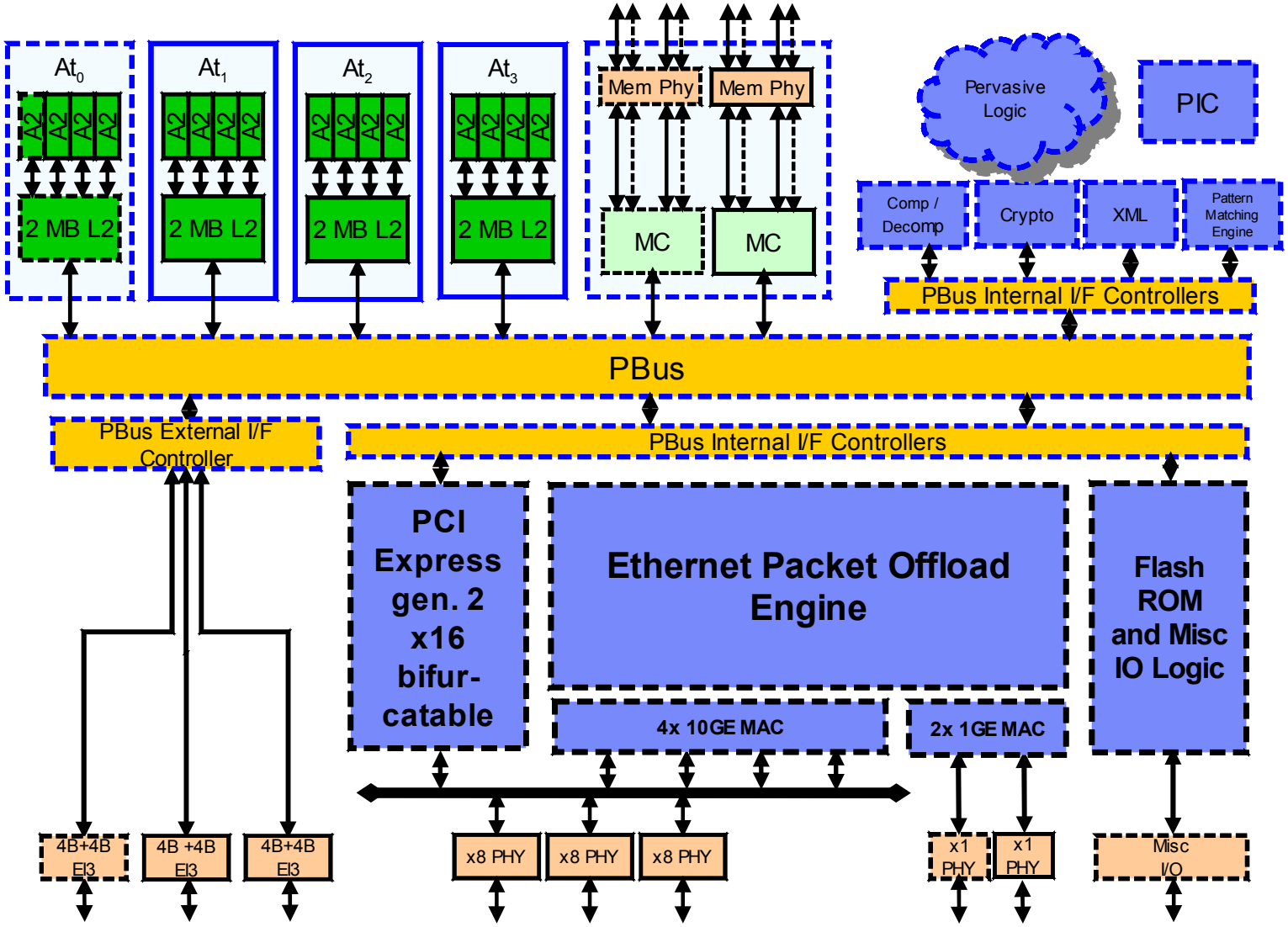
Efficient management of the **I/O**

fast (controller as close as possible to the CPUs) smart (it can be tailored with respect to specific exigences)

The challenge: the target application set



PowerEN Architecture - Overview



PowerEN Architecture - Overview

Full System on a chip whose components are highly integrated.

16 x 2.3Ghz PowerPC Cores known as A2 cores.

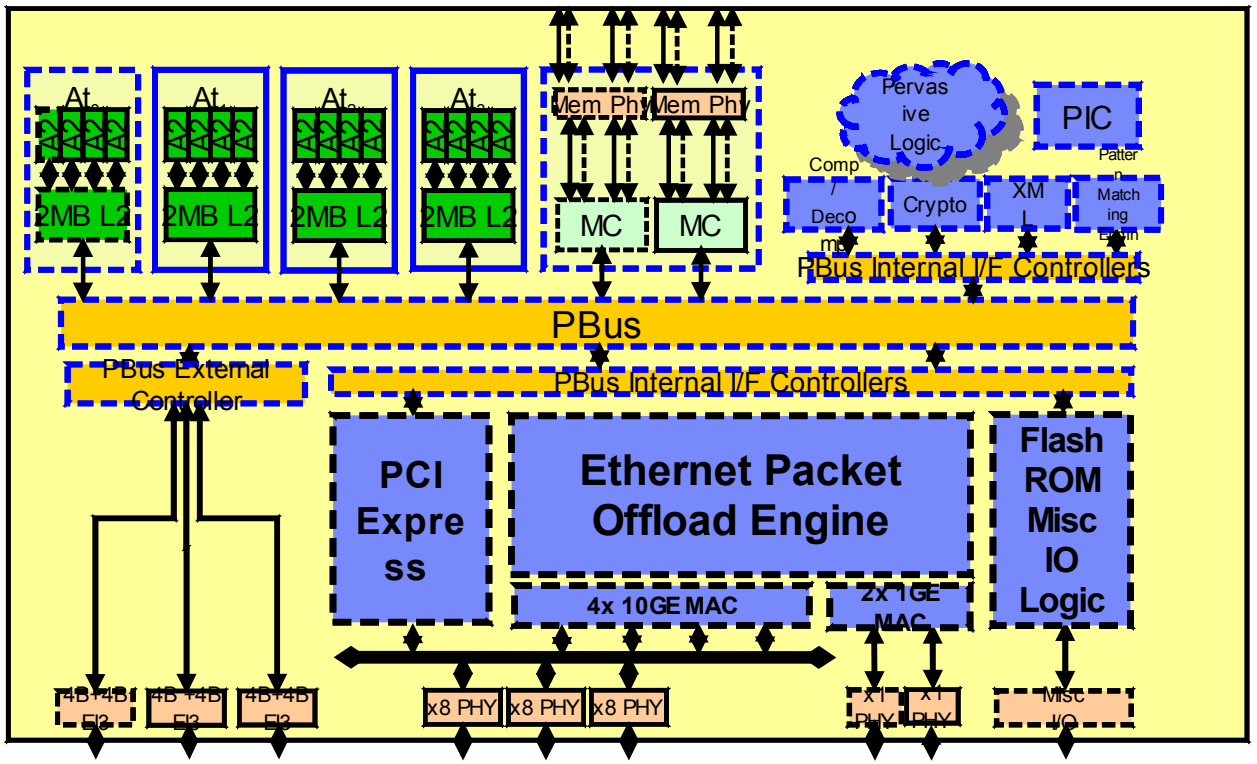
A2 cores are 64-bit and come with full PowerPC ISA support.

Each A2 core comes with support 4 hardware threads.

A2 cores are packaged in groups of 4 known as AT chiplets.

Each AT chiplet has its own 2Mb L2 cache.

2 x integrated memory controllers



4 x 10Gb ethernet interfaces which are part of an integrated optimized Ethernet Offload Engine (HEA) that handles advanced packet processing functions.

XML, Cryptography, Regular Expressions and (De)Compression accelerators.

Highly optimized for OS bypass operations, this integrated architecture is designed to reduce the overhead of inter-component communication.

A2 compute node

64-bit CPUs + SMT + CMP + SMP

Power Instruction Set Architecture

4-way fine grained simultaneous multi-threaded

2 way concurrent issue
- 1 Branch/Integer/Load/Store unit
- 1 FP unit
- instruction from different threads / cycle

In-order dispatch and execution

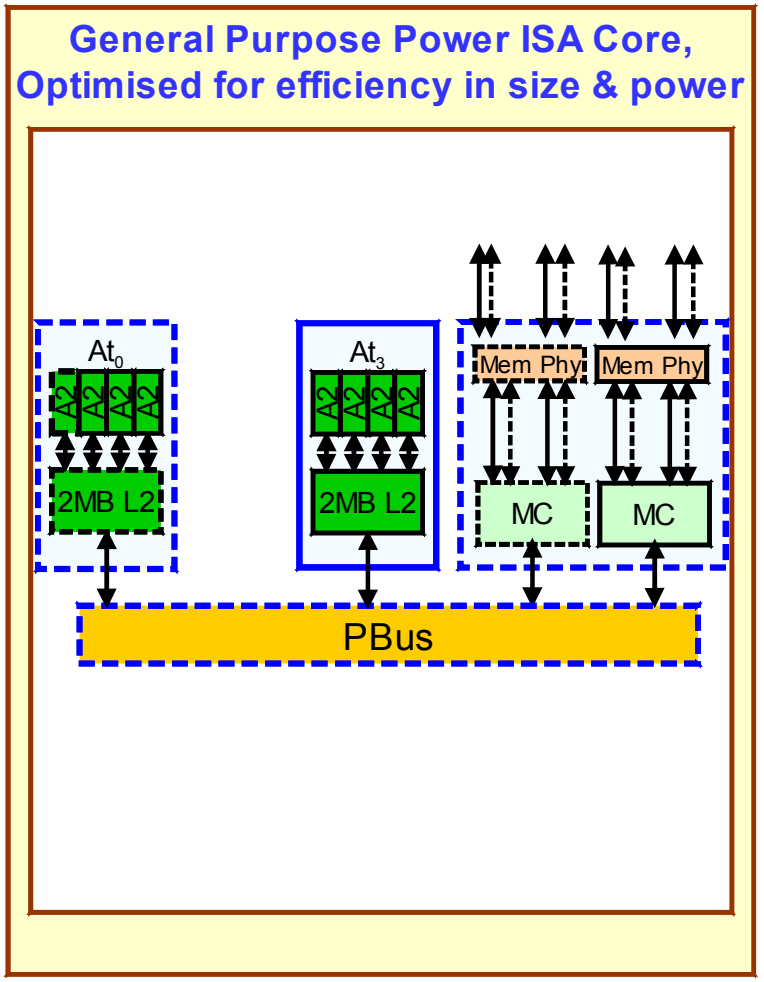
16KB 4-way set associative L1 I/D caches

64B cache line

Binary compatibility for application level code

Performance monitoring interface

Multiple low-power states supported



Host Ethernet Adapter (HEA)

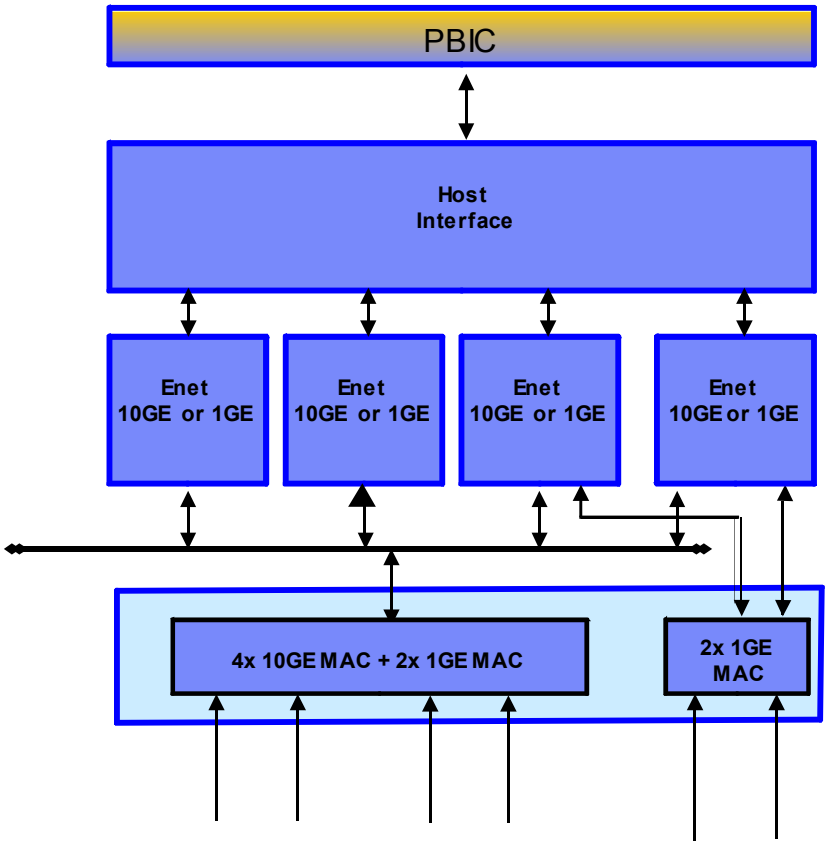
The HEA is an integrated intelligent ethernet controller with 4x10G ports which provides high bandwidth network connectivity for the PowerEN.

HEA could also be considered to be a hardware accelerator because it accelerates packet processing tasks.

When we say that it is intelligent we mean that is programmable and can do lots of useful things such as off-loading of protocol processing to hardware.

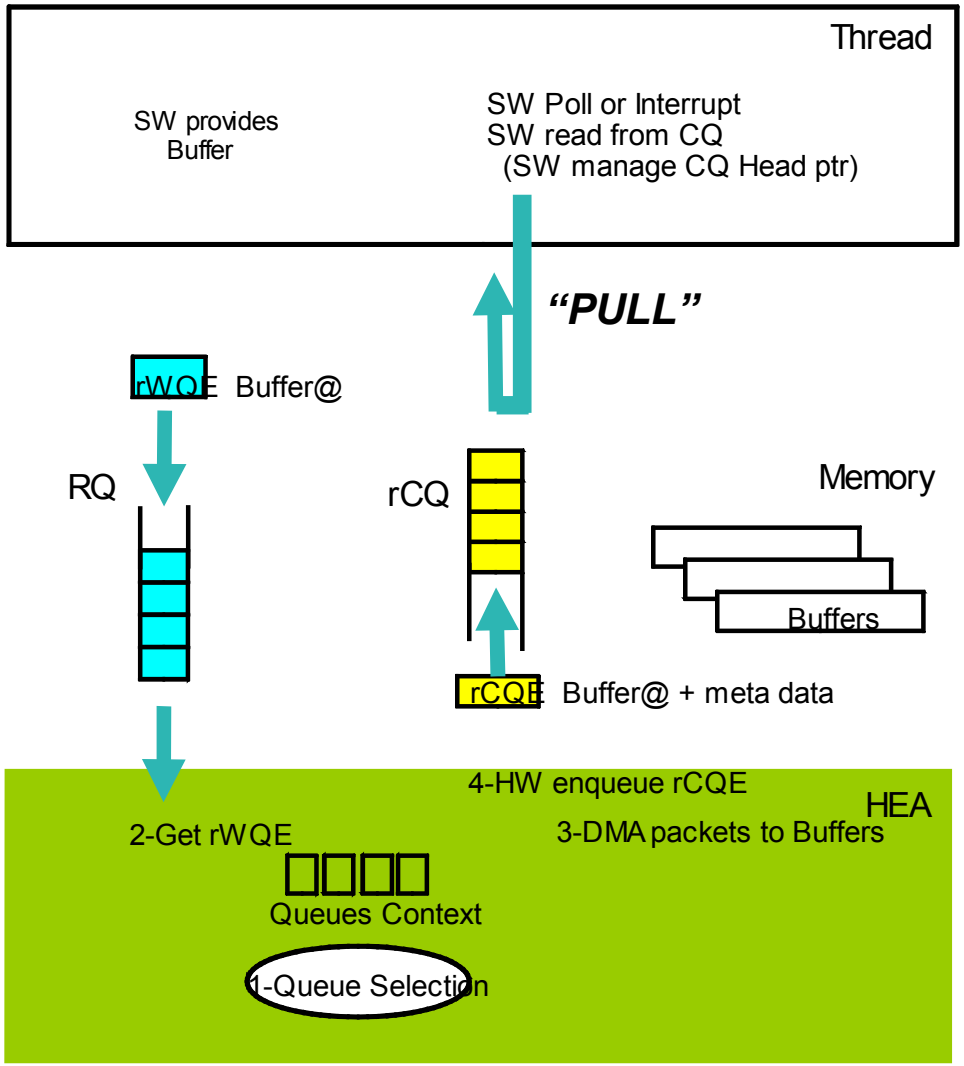
In regular computing systems simply copying packets that arrive off the network from kernel to user space is a massive overhead. The HEA avoids this though by giving user space applications direct access to packet buffers.

Support for virtualization.



HEA – Endpoint Mode

- Receive Software interface
  - Pull Model
  - Flexible queue interface
    - Queue Pairs, Completion Queues, and Event Queues
  - Scatter/Gather descriptors
  - Immediate data in queue elements
- Receive path
  - Flexible Packet Parsing
    - Cksum, Queue selection, metadata
    - IPv4 and IP6
  - Checksum offload
    - TCP / UDP Checksum validation
  - Multicast/VLAN filtering.
  - Queue Selection.
  - Cache Injection.
  - Header / Payload separation.



Receive Side

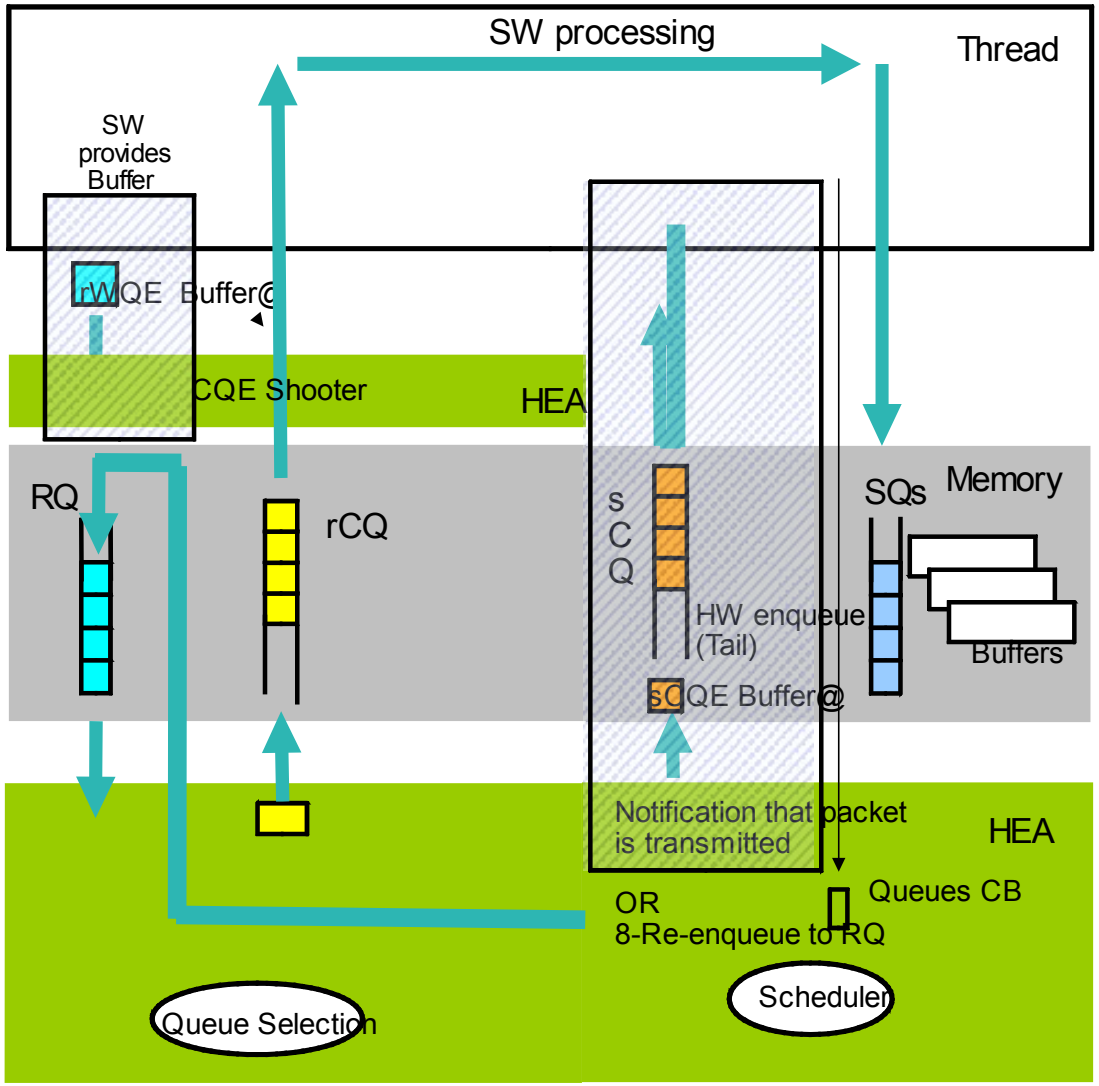
HEA – Network Node Mode

Network node targets appliance boxes situated in the middle of the network.

Network node is not the desination point for packets, rather just an intermediate node in the network.

Example applications are routers, firewalls, and intrusion detection systems.

Features:
- Hardware assistance for maintaining packet ordering.
- Packet Queus managed by hardware.
- Small WQE (PBus bandwidth saving).
- Hardware Scheduler.
- Packet Classification/Parsing functionality.



PowerEN Architecture - Hardware Accelerators

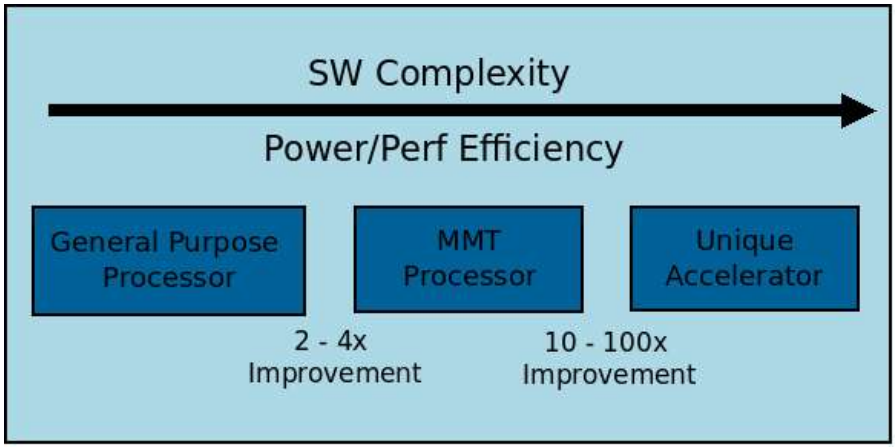
A *hardware accelerator* is a specialised computer chip that is capable of performing specific tasks much faster than is possible using a general purpose CPU but at the cost of decreased flexibility. One of the best well known example of a hardware accelerator is a video or graphics card.

Most existing accelerators are devices attached to an I/O bus (e.g. PCI Express).

For their targeted tasks accelerators can provide significantly more computation performance and chip density at a lower power budget.

Since accelerators differ from general purpose CPUs, the way in which they are programmed also differs (e.g. OpenGL/Direct3D** for graphics cards).

This can make the task of developing software more difficult (e.g. specialised graphics programmers are needed to develop video games).



PowerEN™ Architecture - Hardware Accelerators

The PowerEN™ contains the following accelerators:

- XML Accelerator.
- Regular Expression and Pattern-matching Accelerator.
- Compression and Decompression Accelerator.
- Cryptographic Accelerator

These accelerators have been specifically chosen based on the requirements of network-facing applications.

In later chips based on the WSP architecture these specific accelerators could be replaced by other accelerators with different functionality.

The PowerEN has many features that simplify the task of programming accelerators.

Application programs will immediately be able to take advantage of the accelerators in the PowerEN because the software environment will provide accelerated versions of standard libraries.

PowerEN™ Architecture - Hardware Accelerators

The PowerEN™ provides the performance advantages of accelerators while reducing the software development costs associated with the heterogeneity of accelerators.

Cache injection into the L2 cache is supported so that data from special-purpose accelerators and packets from the network do not need to be stored on off-chip memory before being accessed by an A2 thread. Having data readily available in the cache reduces the latency associated with memory load operations.

In contrast to existing systems, in which most accelerators are devices attached to an I/O bus and are programmed through system-level interfaces, the individual complexes in the WSP (compute, accelerator, and packet processing) all operate using the same application address space (i.e., virtual address space):

Principle of uniform addressability in a heterogenous environment

PowerEN™ Architecture - Hardware Accelerators Features

PowerEN™ is **heavily optimised for user space operation** on accelerators:

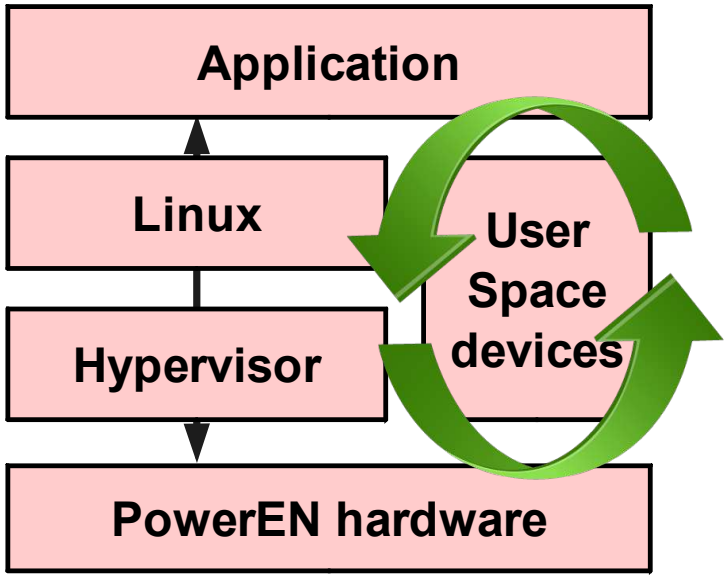
Operations on the accelerators can be initiated directly from user-space, thus avoiding the overhead of a system call which invoke a task switch to the kernel.

This is achieved by using a new instruction that has being added to the PowerPC ISA for the WSP architecture.

The accelerators also “understand” virtual addresses and thus user space processes and accelerators “work at the same desk”.

Accelerator input data can be directly DMA'd from application virtual address spaces and the output from the accelerators after compution has completed can be directly DMA'd back to application virtual address space.

Thus all accelerator/application intereaction can bypass the kernel.



PowerEN Architecture – Cryptography Accelerator

Cryptography is key component of many networking applications as many require secure communication (online banking, secure remote access).

As cryptography is highly computationally **expensive** and is **difficult to parallelise**, specialised hardware accelerator is required to support these networking applications at wire speed.

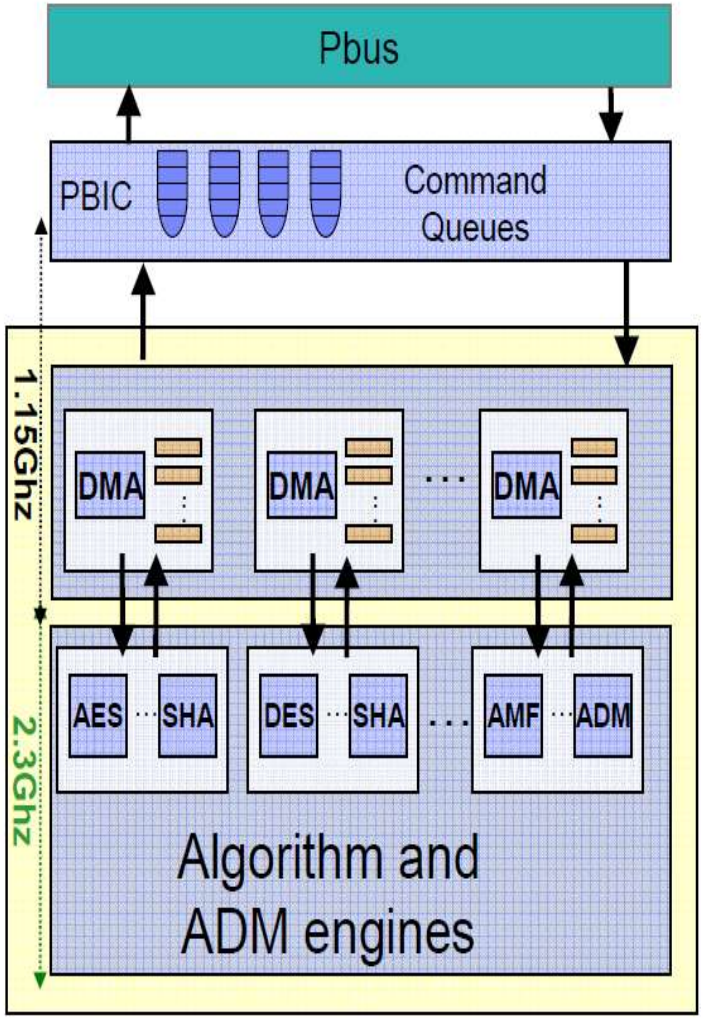
Variety of algorithms supported.
- Cryptography – DES/3DES, AES, GMAC, RC4
- Signature/Hash- SHA-1, SHA-2, MD5, HMAC, AES-XCBC-MAC-96

Supports full Coherency Protocol.

Data can come/go from Cache and/or Memory, **No alignment restrictions**.

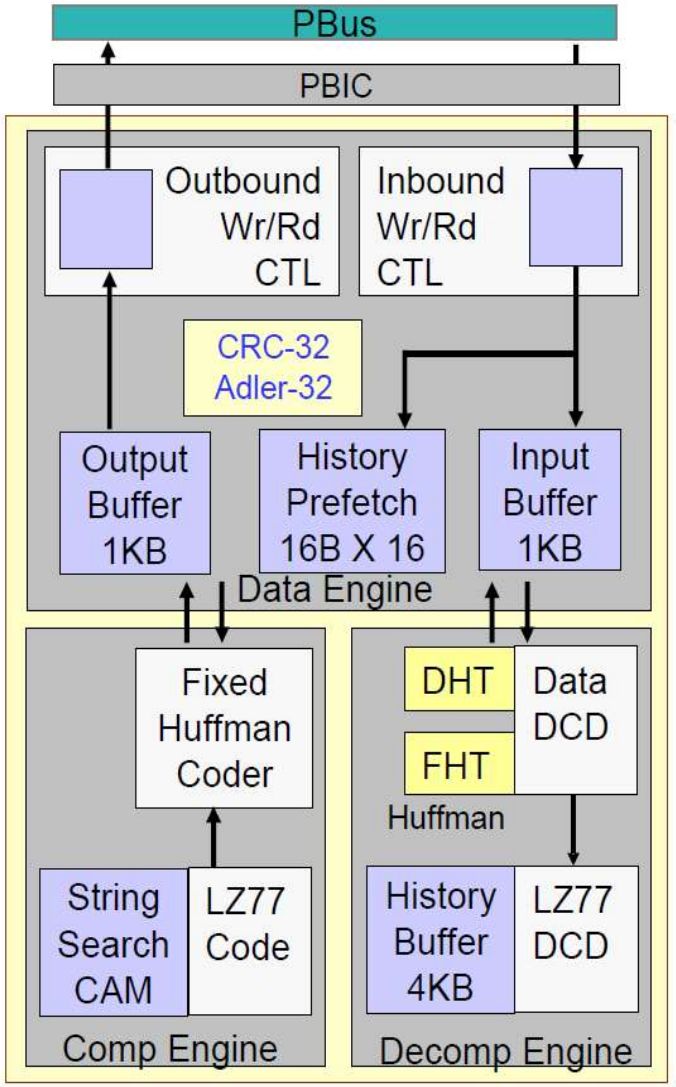
Random Number Generator -
- Supplies a 64b random number, supports FIPS 140 compliance.

Can be leveraged either using an optimized implementation of OpenSSL** or a special library developed for programming PowerEN™ accelerators.



PowerEN Architecture – (De)Compression Accelerator

- Compression is a useful method of over coming limited bandwidth (for example on wireless networks).
- Compression can also be used to reduce storage requirements (e.g. SAN).
- The (De)compression Accelerator provides hardware support for implementing DEFLATE algorithm:
  - LZ77 coding
  - Huffman Coding.
- Can be used to develop optimised implementations of gzip and zlib for example.
- Can be utilised by applications either by using an optimized version of zlib or using some other userspace libraries/interfaces that have been especially developed for programming the accelerators.



PowerEN Architecture – Pattern Matching/RegX Accelerators

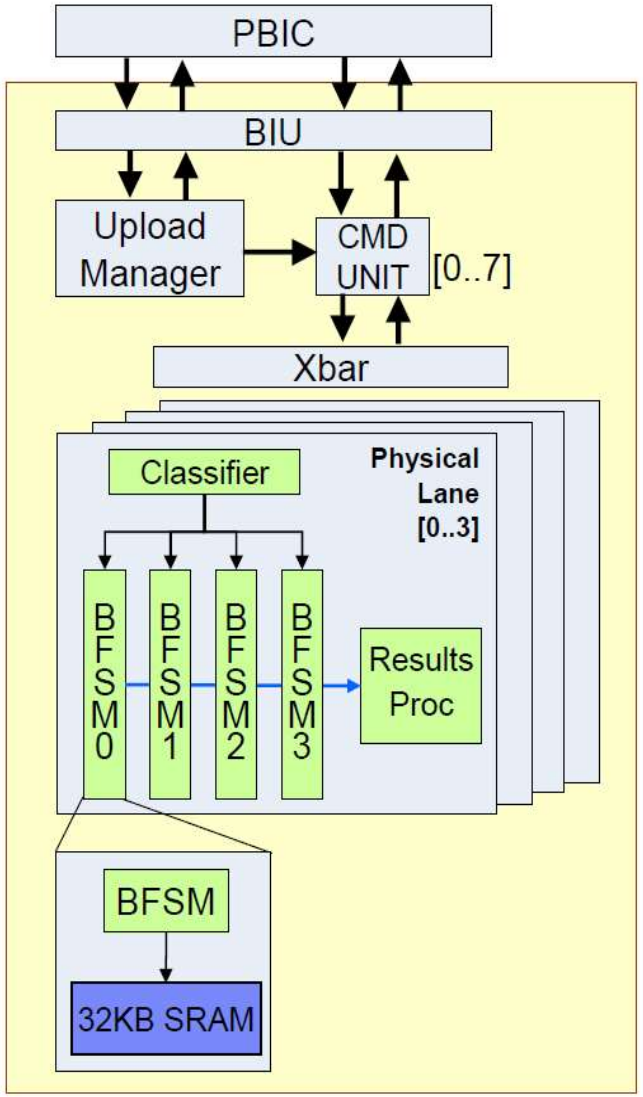
Regular expressions are a formal method for specifying sets of strings.

Used by intrusion detection/prevention systems.

Can process 8 requests in parallel.

Implementation based on programmable state machines (known as Bart Finite State Machines).

Again application developers that wish to leverage this accelerator will have a choice between an optimised port of a standard pre-existing library (e.g. Perl** Compatible Regular Expression) and a tailored interface developed especially for this accelerator.



PowerEN Architecture – XML Accelerators

Extensible Markup Language (XML):
a simple standard text-based format for representing structured information: data + semantic

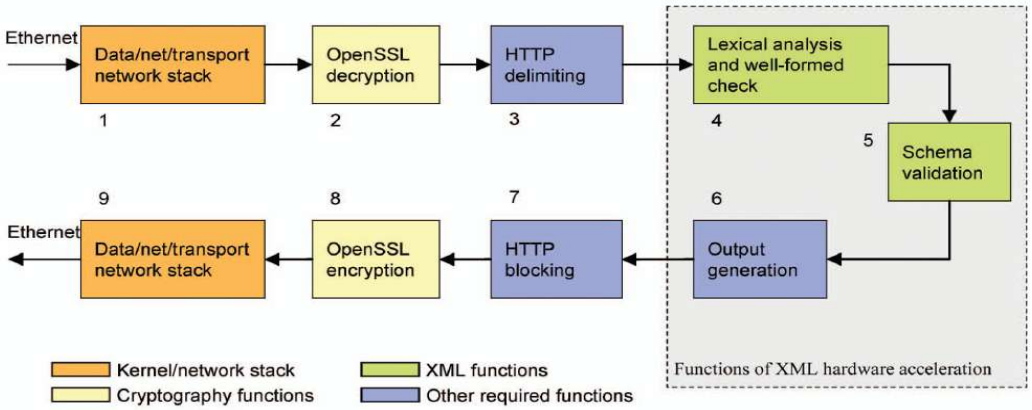
Some Features: Standard, Human readability, machine independent, extensibility …

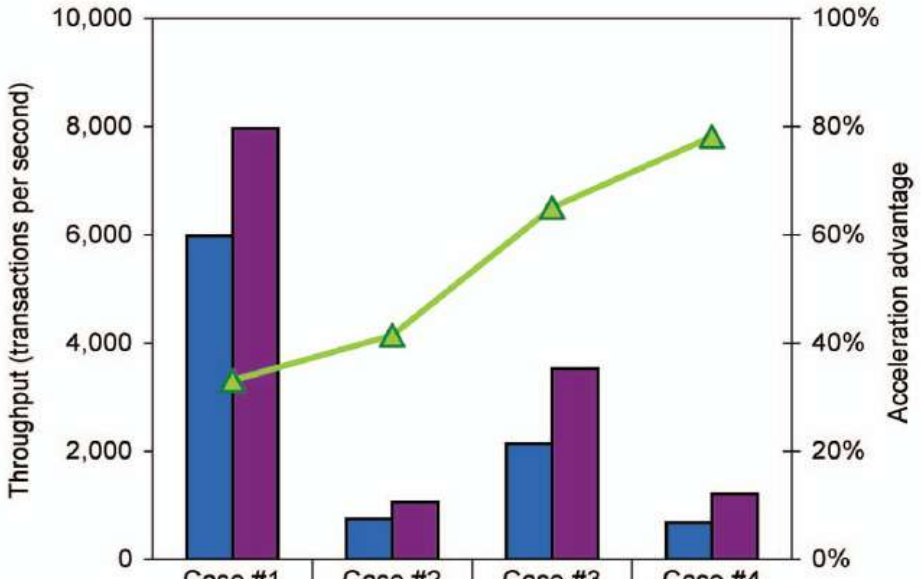
```
<part number="1976">
  <name>Windscreen Wiper</name>
  <description>The Windscreen wiper
    automatically removes rain
    from your windscreen, if it
    should happen to splash there.
    It has a rubber <ref part="1977">blade</ref>
    which can be ordered separately
    if you need to replace it.
  </description>
</part>
```

Where it is used?

Communication: XMLRPC, Web Services(SOAP), Globus (grid infrastructures)
Word processing: ODF and OOXML formats
Graphics: formats such as SVG
Querying: Xquery (Xpath+SQL-like expressions)

PowerEN Architecture – XML Accelerators





	Case #1	Case #2	Case #3	Case #4
TPS via software	5,987	748	2,137	682
TPS via hardware	7,970	1,058	3,527	1,215
Speedup	33%	41%	65%	78%

Franke, H. Nelms, T. Yu, H. Achilles, H. D. Salz, R. ***Exploiting heterogeneous multicore-processor systems for high-performance network processing*** IBM Journal of Research and Development *2010*

PowerEN Architecture – XML Accelerators

The most clear example showing the power of the ***Principle of uniform addressability in a heterogenous environment***

Parse XML document and **post-processing** the results:

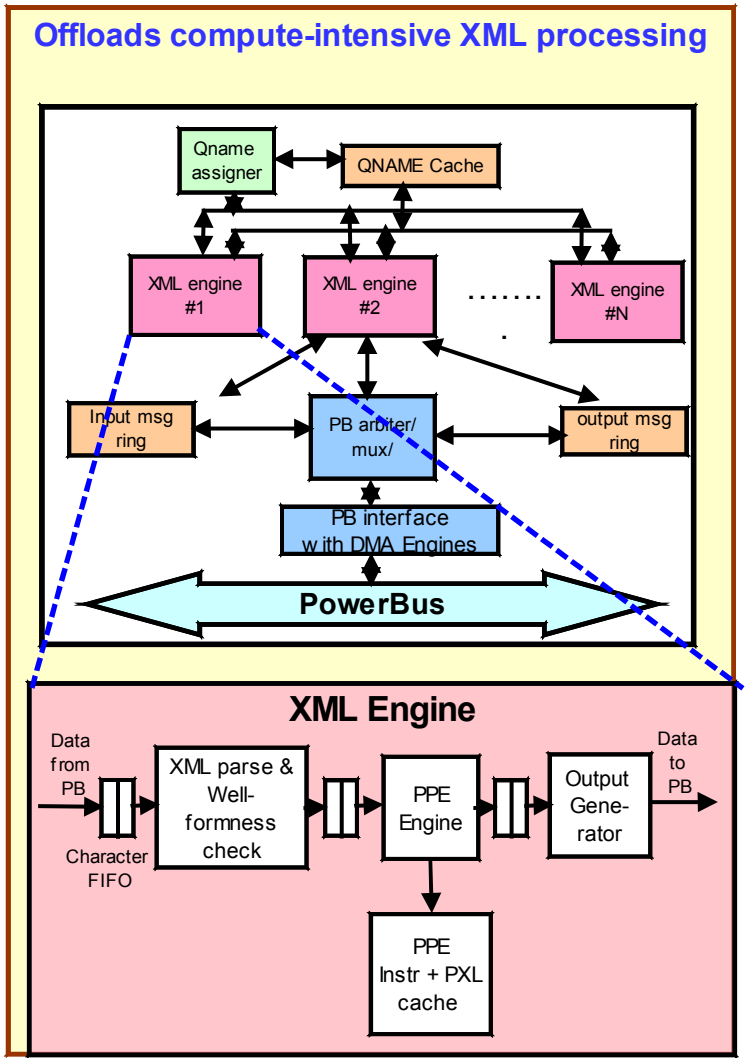
- Soap validation, Schema validation, XPATH validation, XSLT Assist
- **Programmable Post Processing**

Support a Fully stream and concurrent model

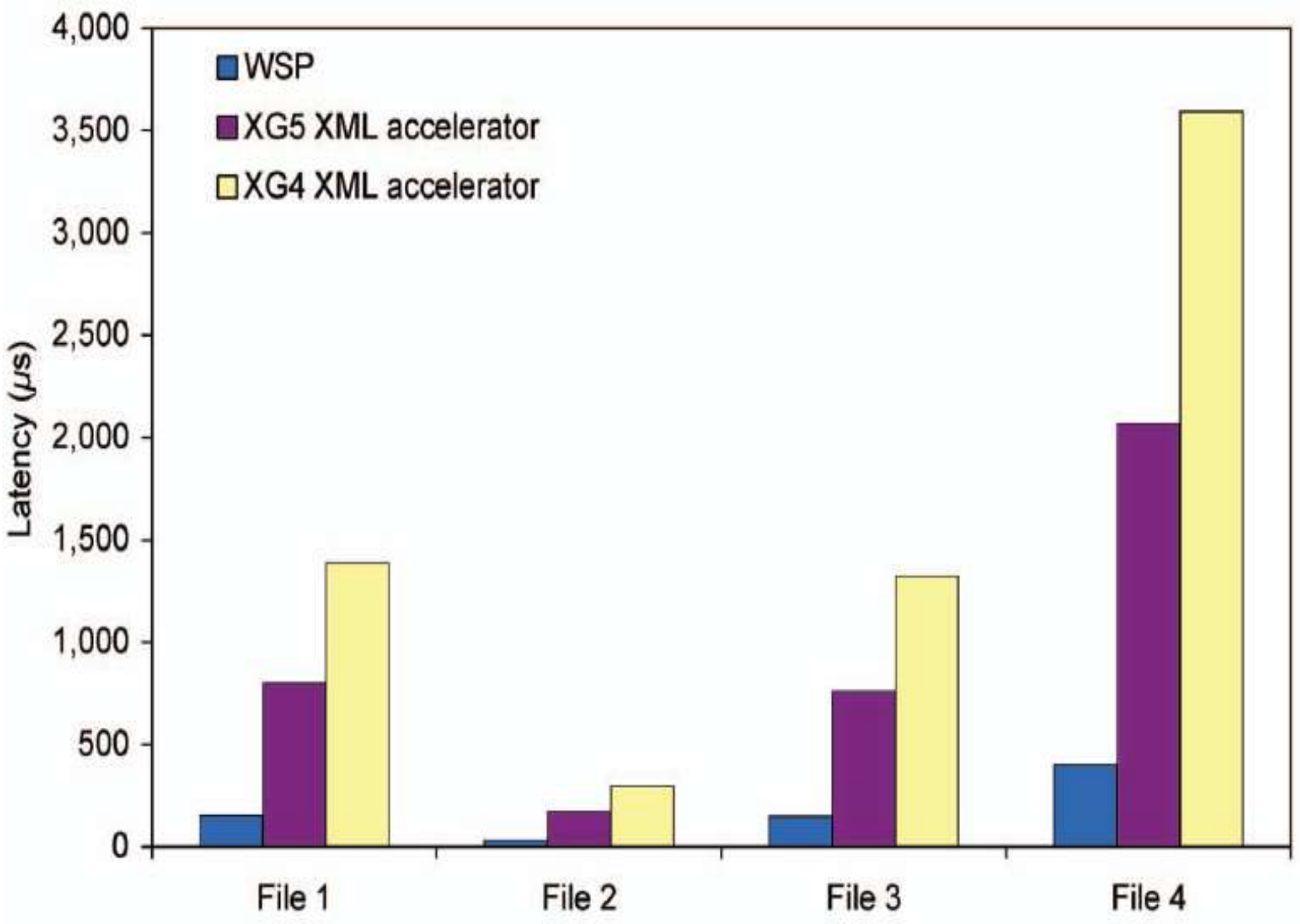
- input in chunks up to 64KB in size, directly from user space documents (**zero-copy**)
- incremental output results
- more than thousands of concurrent documents

Two types of output results

- TLA: incrementally consumed
- **Tree**: consumed at the end

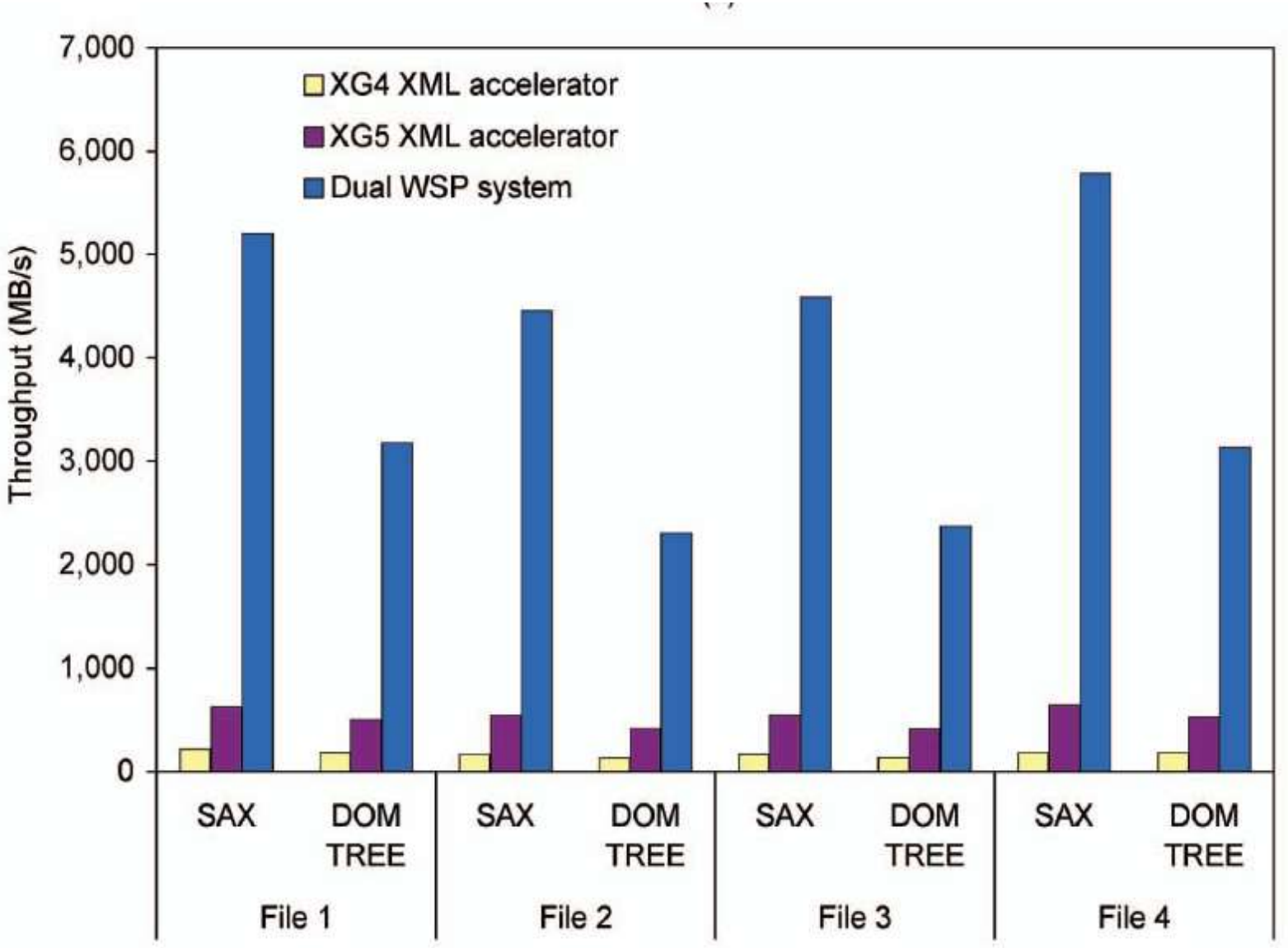


PowerEN Architecture – XML Accelerators



Franke, H. Nelms, T. Yu, H. Achilles, H. D. Salz, R. ***Exploiting heterogeneous multicore-processor systems for high-performance network processing*** IBM Journal of Research and Development *2010*

PowerEN Architecture – XML Accelerators



Franke, H. Nelms, T. Yu, H. Achilles, H. D. Salz, R. ***Exploiting heterogeneous multicore-processor systems for high-performance network processing*** IBM Journal of Research and Development *2010*

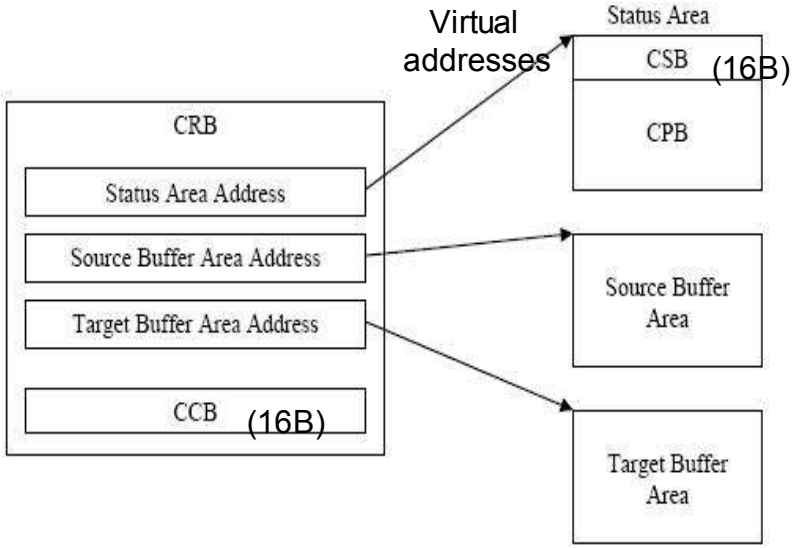
How are the Accelerators Initiated?

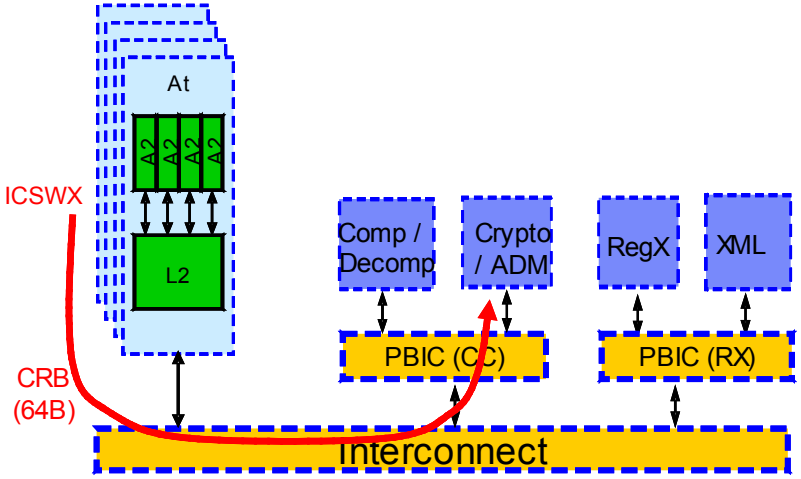
Software thread executes an ICSWX instruction

(ICSWX = Initiate Co-Processor Store Word Indexed)

Hardware adds the process and Guest OS ID information and turns the ICSWX instruction into a CRB (Co-processor Request Block)

CRB arrives at the destined PBIC, where it is validated, and queued





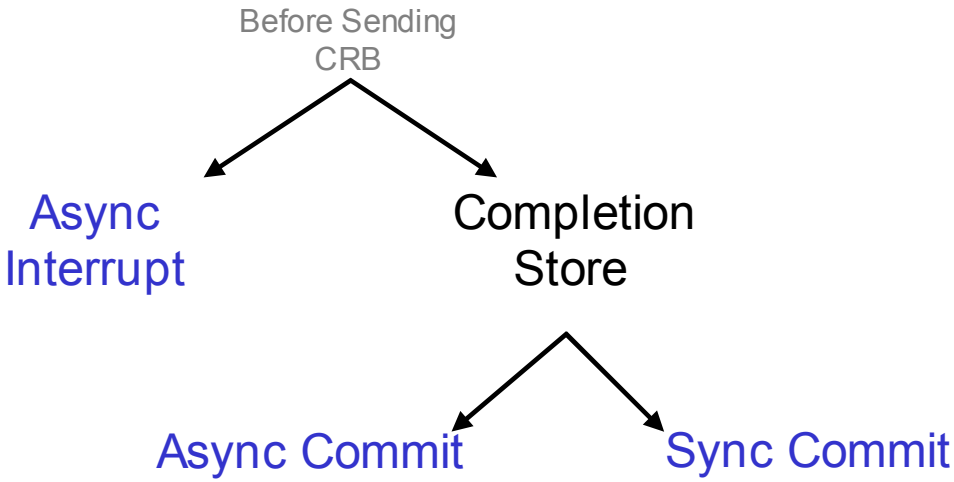
The accelerator services the request. Virtual addresses are translated to physical within PBIC, using a translation table. This is similar to the general-purpose core translation except that TLB-misses take longer to resolve.

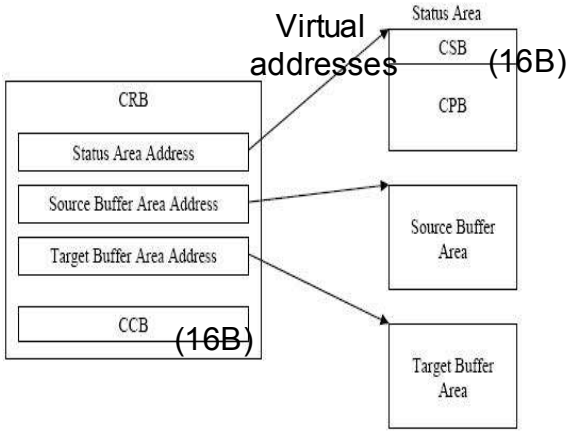
Accelerator sends back update into CSB

Programming Model

One synchronous model

Two asynchronous programming models





PowerEN Architecture – Smart Hardware Wait Features (1/2)

Synchronisations is a main concept in parallelism. The cost of synchronisations bounds the grain of the possible parallelism.

Force a thread to wait for a certain event is not a simple operation while targeting performance. In normal systems there are few possibilities, each one with some drawbacks:

pthread_cond_wait: advise the OS scheduler that the thread is waiting for an event.

- The cost of waiting on a condition variable should be little more than the cost for a context switch plus the time to unlock and lock the associated mutex
- The overhead is **not compatible** with syncronizations required by **fine grane parallelizations**.

Polling: repeatedly read a memory location, until a predefine value is found.

- Suitable for fine grain parallelization but ...
- This is an active waiting => waste CPU time + termal dissipation + memory bandwidth

PowerEN Architecture – Smart Hardware Wait Features (2/2)

The main issue of the previous approach is that the “wait” operation is **not a native mechanism of the hardware/firmware level**. PowerEN extends its instruction set with the ***waitrsv*** .

Signature:

waitrsv(addr, expected_val)

Semantics:

The A2 stops dispatching instructions belonging to the thread (no context switch).

The A2 hardware thread is put into a low energy consumption status.

The A2 restart dispatching the thread instruction when the target value is matched.

```
do{}while{(*addr_x)!=expected_val}
```

```
waitrsv(addr_x, expected_val)
```

Implementation Aspects:

Based on memory reservation Power mechanisms

What is libcop?

The Co-processor library part of PowerEN SDK (i.e. WSP SDK)

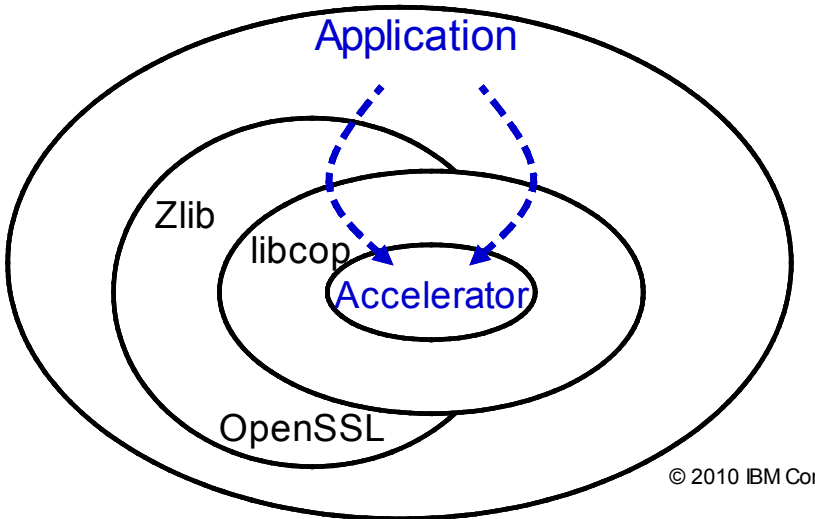
C library for programming one of PowerEN co-processors (accelerators)

Current beta SDK version (Sep. 2010), supports the following accelerators:

- Compression
- Crypto
- ADM (Asynchronous Data Mover)

The beta SDK is still being enhanced both for functionality and performance

Libcop supports the basics for sending requests (CRBs) to the co-processors. It is intended to be used by standard libraries, such as Zlib and OpenSSL, as well as performance-sensitive applications



Leveraging PowerEN

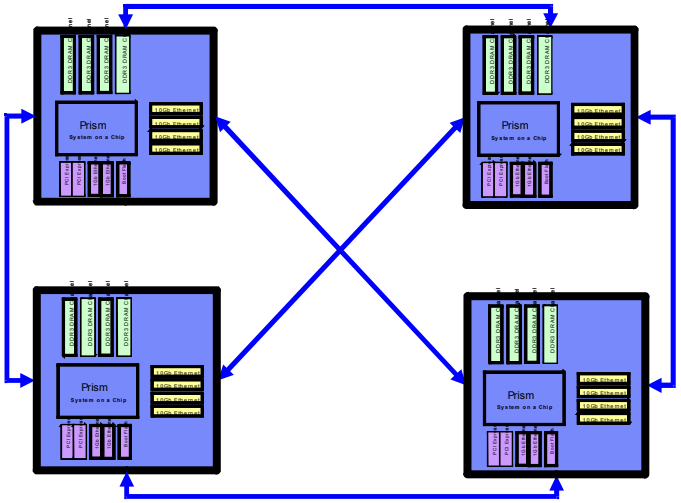
PowerEN is half way between a general purpose and a special purpose computing platform; as such it is well suited for specific classes of applications:

- Network I/O bound applications.
- Highly multi threaded applications with “lightweight threads”.
- Applications that require lots of XML processing.
- Solutions that require high volume crypto.
- Applications that must search patterns and regexp on large volume of data.
- Applications that require high speed compression and decompression.

PowerEN is not suited for:

- Heavyweight or floating point intensive computing,
- Vector processing.

Major Chip Interfaces and 4-chip Scaling



Memory

DDR3 DRAM Channel
DDR3 DRAM Channel
DDR3 DRAM Channel
DDR3 DRAM Channel

Network

10/1Gb Ethernet
10/1Gb Ethernet
10/1Gb Ethernet
10/1Gb Ethernet

Prism
System on a Chip

Chip-to-chip Connection
Chip-to-chip Connection
Chip-to-chip Connection

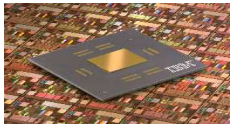
SMP Scaling

LPC/Boot Flash
PCI Express
PCI Express
I2C / GPIO
UART

Base I/O

System Platforms

Single-Chip Module



1 chip,
2.3-3.0 GHz
~65W

Chroma Card

PCIe

1 socket

16 cores

Blades

1-2 sockets,
16-32 cores per blade

1U-2U Rackmount

1-4 sockets,
16-64 cores

Summary of key features of PowerEN

PowerEN delivers a significant leap in chip- & system-level throughput optimization

Massive Multi-threading Processor: 4 to 256 threads in single memory image.

Throughput-optimized design: multi-10GbE I/O, cores, accelerators, high memory BW.

Integrated on-chip accelerators: crypto, RegEx, XML, compress/decompress, HEA.

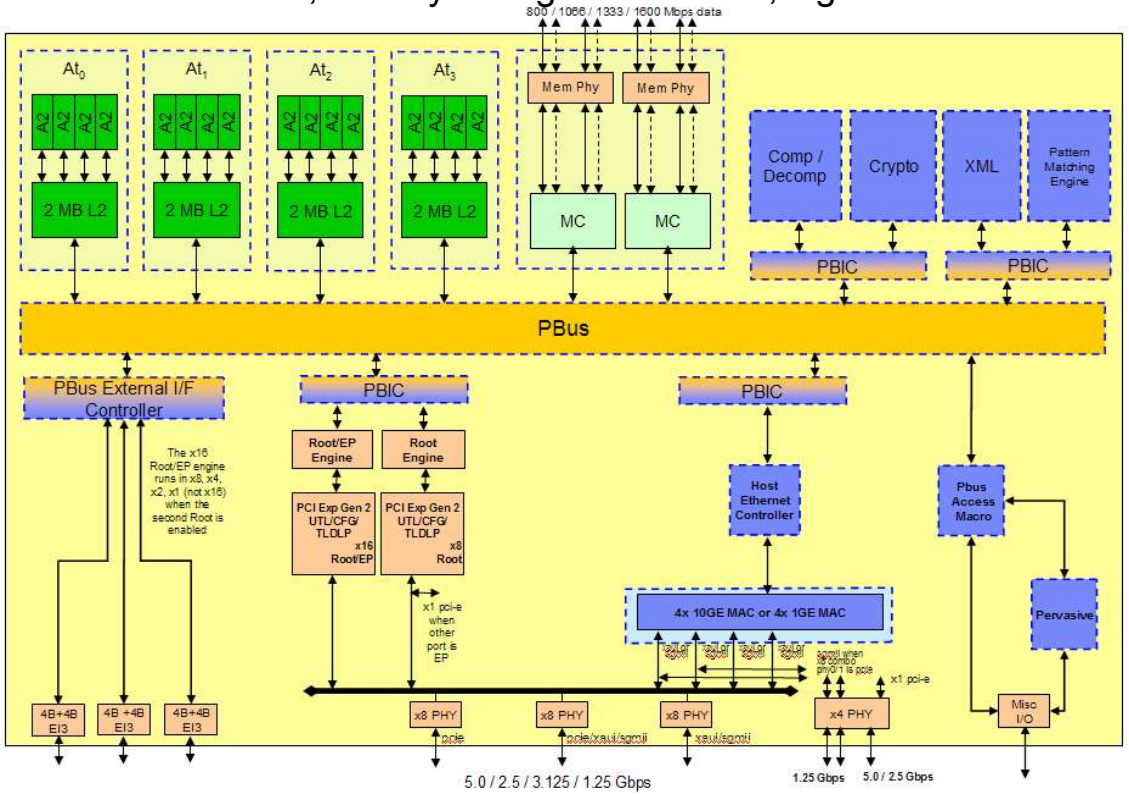
System-on-a-chip design flexibility with great performance efficiency

Configuration options (4-64 cores, SCM/DCM, 2.0-3.0 GHz, integrated PCIe & 1G/10G Enet, on-chip accelerators,..) **allow easy system designs optimizations**

System-optimized throughput: eDRAM for L2 caches, latency-hiding with threads, high BW on-chip buses

Family of design points: 2U rack-mount down to PCIe adapter cards

Priority focus on edge computing, infrastructure & integrated stacks



** Trademark, service mark, or registered trademark of OpenSSL Project, The Perl Foundation, Linus Torvalds, Sun Microsystems, Microsoft Corporation.

Any questions?