The Origin Architecture

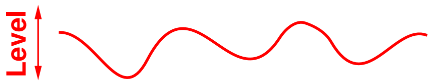
Ruud van der Pas

SiliconGraphics

European High Performance Computing Team

Topics

- *Introduction and Origin design issues*
- *The Origin Architecture*
- *Bandwidth and Latency*



Introduction

Goals of Origin design

- *Follow–on to (Power) Challenge SMP*
- *Aggressive remote memory latency/bandwidth*
- *Scalable to a large number of high–performance CPUs*
- *Continued leadership in high–performance I/O and graphics*
- *Scale cost–effectively down, as well as up*
- *Incremental – "Pay as you go" – modularity*

Microprocessor Issues

- *Amdahl's Law* ➡ *High performance uniprocessor needed*
- *Large–Scale MP* ➡ *High integration level needed*
- *Large shared address space*
  - *64 bit virtual, 40+ bits physical*
- *Memory system:*
  - *Integrated support for cache coherence*
  - *Multiple non–blocking prefetch memory operations*

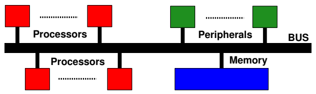
About Parallel Systems

Major issues to consider

- Economics (scalability, pricing, market, ..)
- Ease of use
- Design decisions
  - Interconnect and topology
  - Latency and bandwidth
  - Memory characteristics

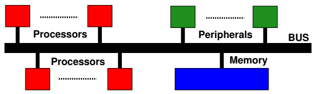
(cache coherency, NUMA, ...)

The Bus–Based Revolution



- System for throughput and parallel workload
- Affordable and expandable UNIX system
- Easy to use:
  - Runs well on dusty deck codes
  - Flexible and powerful parallel programming

The Next Revolution: ~~Bus–Based~~?



Bus based systems do not scale infinitely:

- Economically

  80 MB/s –> 1.2 GB/s –> 18 GB/s ?

  Entry price would be high

  important for small configs
- Performance–wise

  Very difficult to keep up with new processor speeds during the life–cycle of the bus

  important for large configs

Building a Parallel System

Key *technical* factors for a modern parallel system:

- **Runs well on dusty deck codes**
- **Flexible and powerful parallel programming**

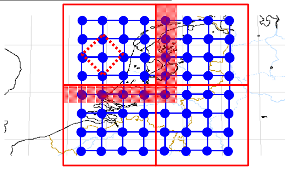
Hardware features required for this:

- **Low Latency**
- **High Bandwidth**

We will illustrate this with a (simplified) example

About Latency and Bandwidth

In many parallel applications, communication is required for:



- ***Small packets of data***

  E.g. synchronization, residuals of a solution, etc.

  Measured by: "the latency"

- ***Large packets of data*** ("bulk transfer")

  E.g. boundary conditions along the surface of the grid

  Measured by: "the bandwidth"

Building a *Usable* Parallel System

Low Latency (= efficient on short messages)

- **Critical for fine grained parallelisation**
- **Increases ease of use**

High Bandwidth (= efficient on long messages)

- **Needed in order to deal with large problems**

Building a *Scalable* Parallel System

Issues that need to be resolved:

- **Interconnection Topology**

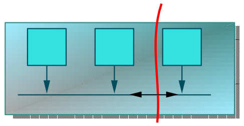
  "How do we connect the processors and memory together?"

- **Memory Characteristics**

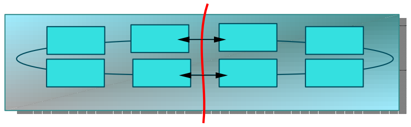
  "How do we design a scalable memory system?"

Bi–Section Bandwidth

How much data can be transferred from one half of the system to the other half?

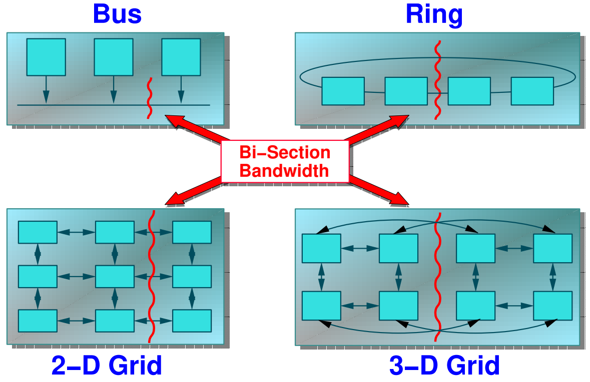


Bus Bandwidth =1.2 GByte/s
Bisection Bandwidth =1.2 GByte/s



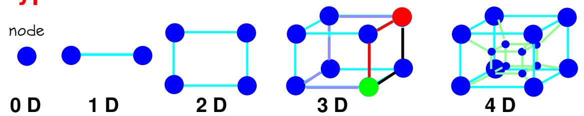
Bus Bandwidth =1.2 GByte/s
Bisection Bandwidth =2*1.2 GByte/s

Interconnection Alternatives



The Hypercube Topology

Hypercube = multi–dimensional "cube"



Advantages:

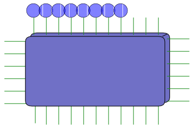
- *Very limited number of "hops"; even for large configurations*
- *Linear bisection bandwidth growth with the number of nodes*
- *Latency is proportional to ^{2}log of the number of nodes*
- *Multiple routing paths ( ) and fault tolerance*
- *Modularity*

Building a *Scalable* Parallel System

Issues that need to be resolved:

- ✓ Interconnection Topology *Hypercube*
- Memory Interconnect ?
- Memory Model ?

The Next Revolution: Crossbar?

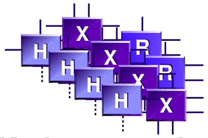


Conventional, monolithic crossbar is only effective when fully loaded

Crossbar:

Connect every input with every output

The Next Revolution:Modular crossbar



A modular crossbar scales both in performance and economically!

Modular crossbar to replace the bus:

- **It is a true crossbar**
- **Difference: *expandable* system component**
- **Scales well in economics** ("pay-as-you-grow")
- **Delivers scalable performance**

Building a *Scalable* Parallel System

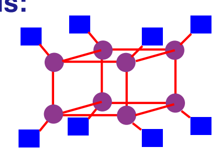
Issues that need to be resolved:

- Interconnection Topology — *Hypercube*
- Memory Interconnect — *Modular Crossbar*
- Memory Model — ?

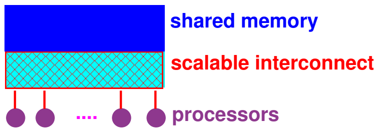
Memory Model: Shared Memory

The memory on the Origin is:

- **Physically distributed:**

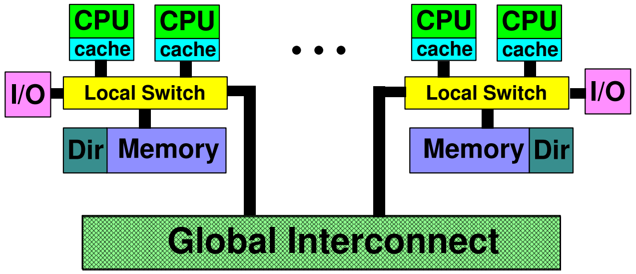


- **One address space:**



All handled in hardware, fully transparent

Distributed Shared Memory (DSM)

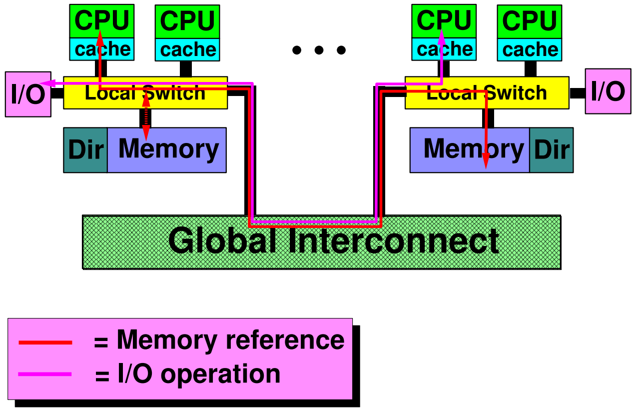


Note:

DSM is an <u>implementation</u>, not a model!

For the user this is a Shared Memory system

Memory and I/O *directly addressable*



Building a *Scalable* Parallel System

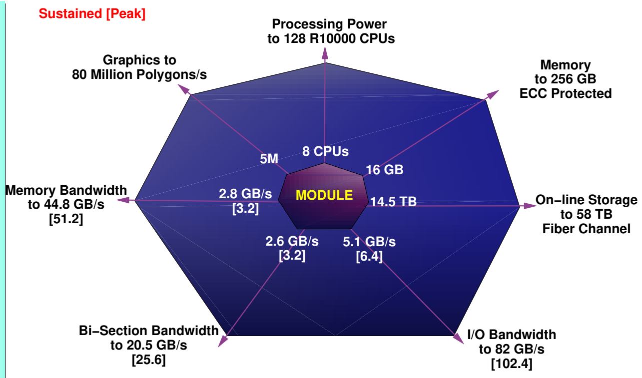
Issues that need to be resolved:

- **Interconnection Topology** — ***Hypercube***
- **Memory Interconnect** — ***Modular Crossbar***
- **Memory Model** — ***Shared Memory ****

*) Implemented as DSM (= Distributed Shared Memory)

Origin Architecture

Origin Grows in All Dimensions



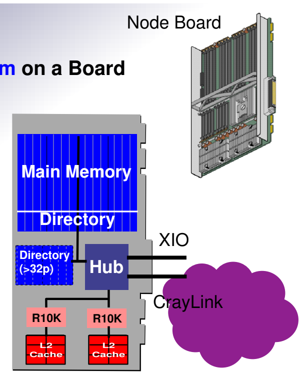
Node: S2MP building block

Node Board: A complete System on a Board

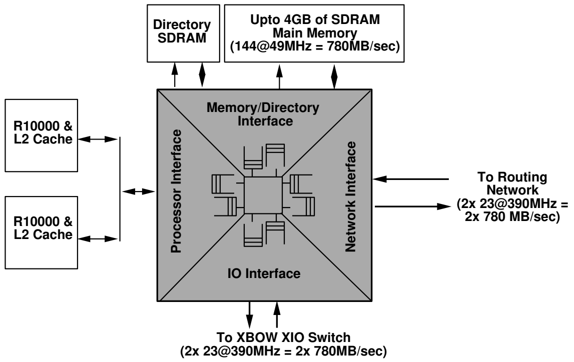
- 2 X R10000 Processors
- 64 MB to 4 GB memory

Hub B/W peak

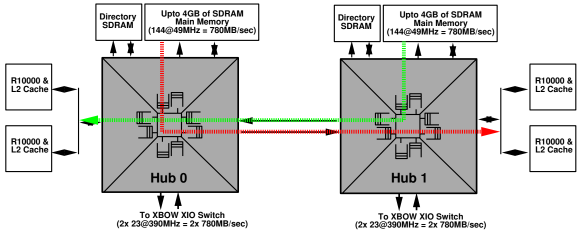
- 780 MB/s – CPUs
- 780 MB/s – memory
- 2*780=1.56 GB/s XIO link
- 2*780=1.56 GB/s CrayLink



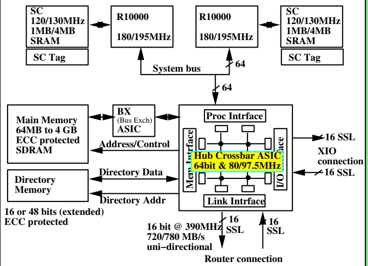
The HUB: Integrated Node Structure



The HUB: Full Remote Bandwidth



Node Board – Block Diagram



Notes:
BX ASIC demultiplexes Hub data into Memory data

The CrayLink Interconnect and XIO are identical at the physical level. They are different in higer levels of protocol.

1–2 R10000 Processors:
- 180MHz with 1MB cache or
- 195MHz with 4MB cache

64 MB to 4 GB SDRAM memory

Special Directory DIMMS for configurations >32 processors

Hub Crossbar:
- Allows processors to talk to memory, I/O devices (XIO) and other processors (CrayLink)

Directory Based Cache Coherency

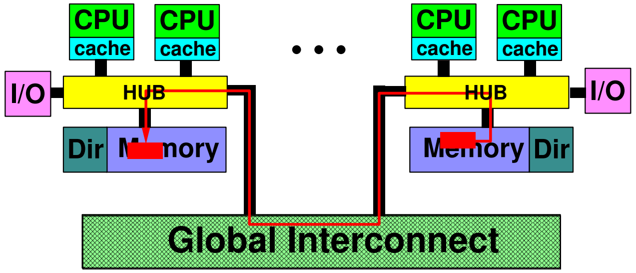
Block Transfer Engine (BTE)

64 page–reference counters/page

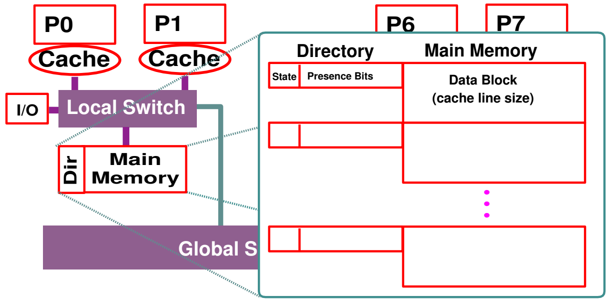
Support for R10K latency hiding:
- 4 outstanding read requests per proc
- 8 outstanding write requests per proc
- 8 incomming interventions per proc

HUB: Used to Migrate Pages

- *HUBs contain page reference counters (64/page)*
- *If the OS detects: remote > threshold% * local: migrate page*

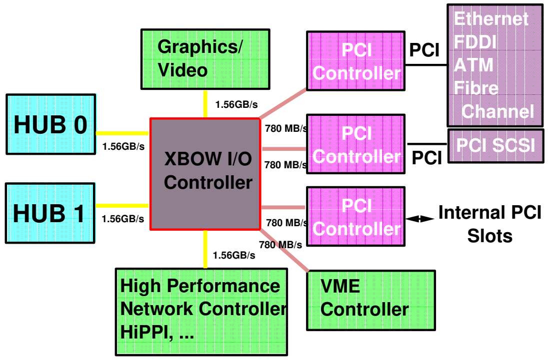


Directory Based Cache Coherence

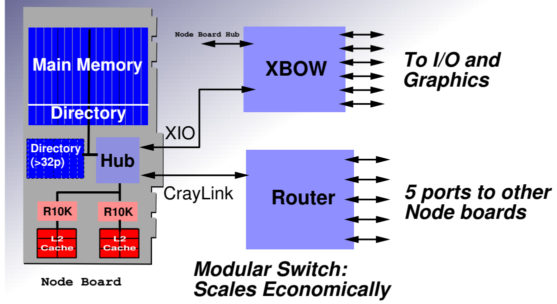


- *Hub that owns the data runs the cache coherency protocol*
- *Processor makes data request to its home Hub*
- *The local Hub sends requests for data to the HUB that owns the data*
- *That HUB sends invalidations as required by the presence bits*
- *The acknowledgements of the invalidations are sent directly to the requesting HUB (improves read/write latency by 25%)*

The I/O Subsystem



S2MP Switch Technology

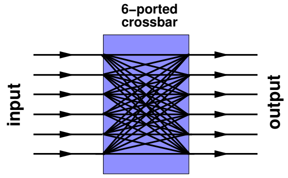


The CrayLink

- Cost-effective, scalable, high bandwidth, low latency interconnect
- State-of-the-art CMOS crossbar ASICs (> 1M gates)
- State-of-the-art pin densities (> 600 pins)
- Micro-packet error detection and correction
- Adaptive Routing

The Router: Scalable Bandwidth

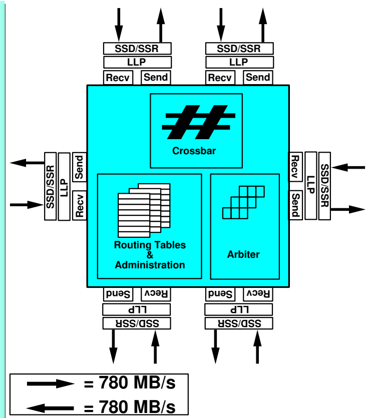
The router from a user point of view:



Each input and output channel runs at 780 MB/s

Total crossbar capacity is 6*0.78*2 = 9.36 GB/s

The Router – Implementation



The 6-way crossbar propagates messages based on preloaded routing tables.

The routing tables are programmable and are setup at boot time.

Wormhole routing is used: route messages without having to receive the entire message first.

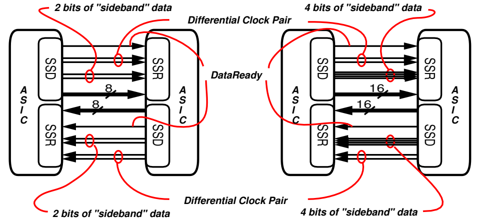
Vector routing (no usage of tables made) is used for network configuration and administration.

SSD/SSR multiplex/de-multiplex external 16bit@390MHz to the router internal frequency of 64bit@97.5MHz.

The Link Level Protocol (LLP) is implemented on chip in hardware and guarantees reliable data transfer (CCITT-CRC).

3 ports are single-ended STL (for communication between ASICS inside the enclosure), the other 3 ports are differential PECL for communication over the CrayLink cables.

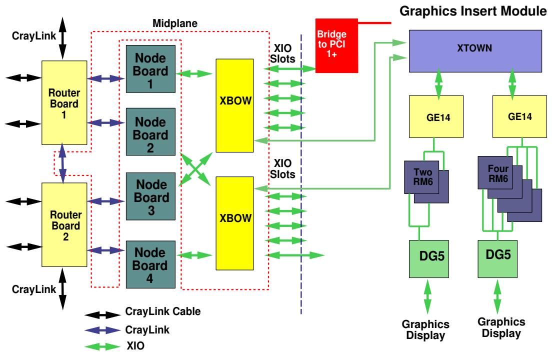
Source–Synchronous Links



Eight bits each way, 23 wires   *Sixteen bits each way, 46 wires*

- The eight–bit wide SSL is capable of 780MB/s peak transfer rate, and the sixteen–bit wide link is capable of 1.56GB/s peak transfers.
- Data are sent back and forth over the links in packets (like network packets), with a CRC checking code added to guarantee the data's integrity. A packet–switching protocol, the *Link Level Protocol*, is used to check the data, and retry (if needed).
- The links can be in PC Board trace or on cable wires, and although they are electrically different, the signals are the same in function and speed.

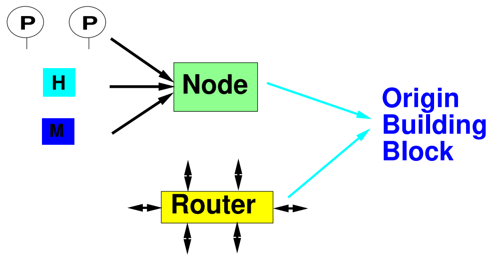
An Origin module (8 CPUs)



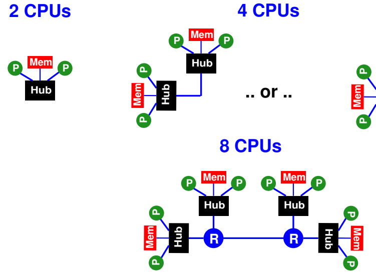
Origin components

R10000 Processor	P	CPU
Memory	M	RAM
Hub	H	node crossbar
Router	R	crossbar

Origin building block

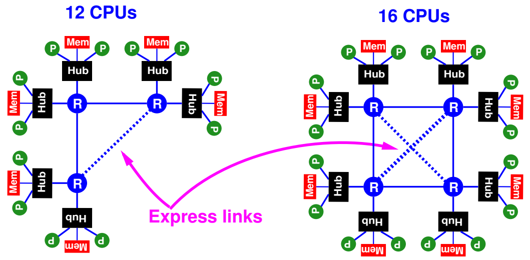


Shared Memory without a bus

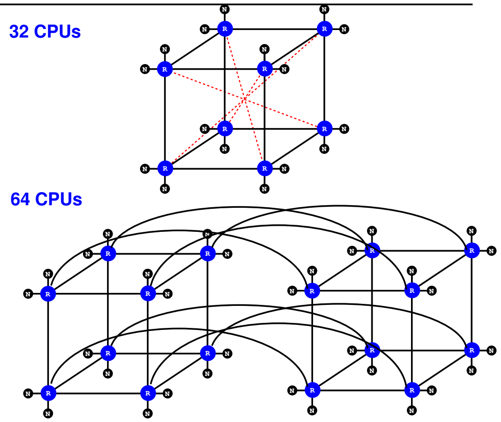


This is a Scalable Design: Every Hub and Router add bandwidth

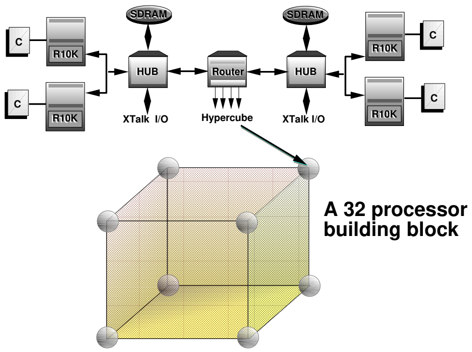
Shared Memory without a bus



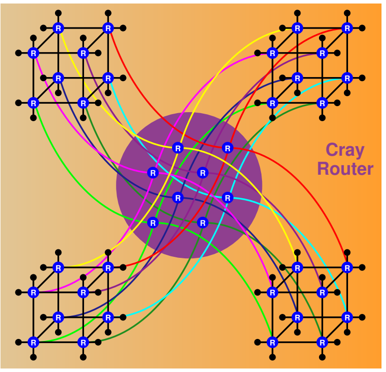
Shared Memory without a bus



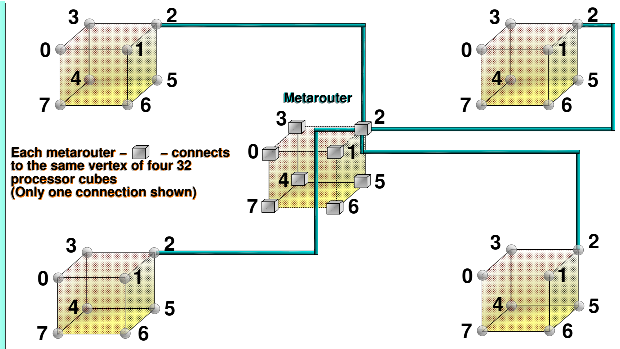
Above 64 CPUs: The CrayRouter



The CrayRouter: 65–128 CPUs

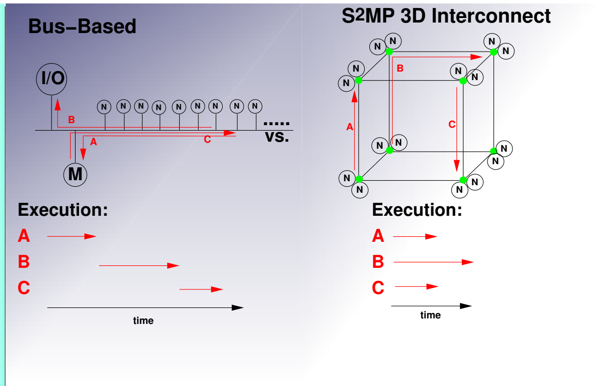


The CrayRouter Structure

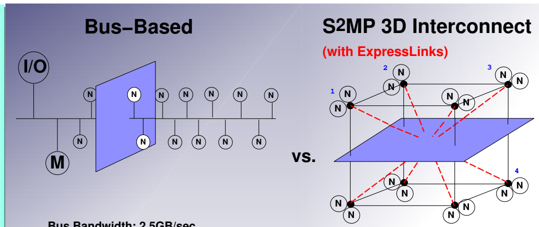


Bandwidth and Latency

Parallel Bandwidth



Bandwidth Increases Linearly!



Bus-Based

Bus Bandwidth: 2.5GB/sec
Bisection Bandwidth: 2.5GB/sec
Number of Processors: 32
Bandwidth/Processor: 2.5/32 = 78MB/sec

S2MP 3D Interconnect

Interconnect Bandwidth: 1.28 GB/sec sustained [1.6 GB/sec peak]
Number of Craylinks Bisected: 8
Bisection Bandwidth: 8 X 1.28 = 10.2 GB/sec [12.8 GB/sec]
Number of Processors: 32
Bandwidth/Processor: 10.2/32 = 320 MB/sec [400MB/sec]

Origin Latency

Multiprocessors inherently have higher memory latencies

Data sharing causes cache invalidations and extra misses because of the need to maintain coherence

Reduce the latency

- Caching
- Minimize cache miss penalties:
  - Lower levels of the memory heirarchy
  - Latency of the interconnect network

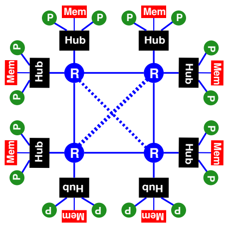
Hide the latency

- Overlap memory access with other work
- Pre–fetch data

Origin Latency

The S2MP Architecture adds a new NUMA level

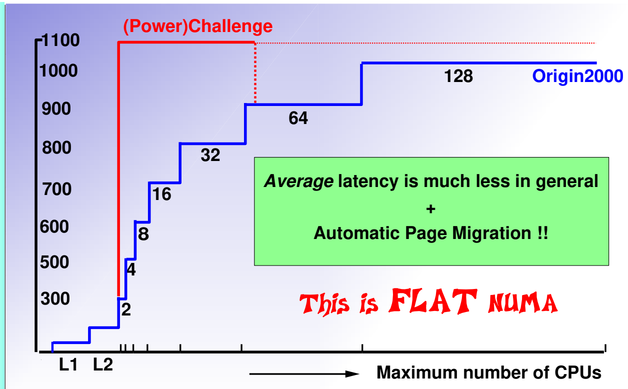
- **There is a notion of "close" and "far away" memory in the system**



Key element of the Origin2000 is that "far away" is more like "pretty close" !!

This is really a unique feature in the industry

Origin Latencies



Average **latency is much less in general**
+
Automatic Page Migration !!

This is FLAT NUMA