Comparing High-End Computer Architectures for Business Applications

Presentation: 493

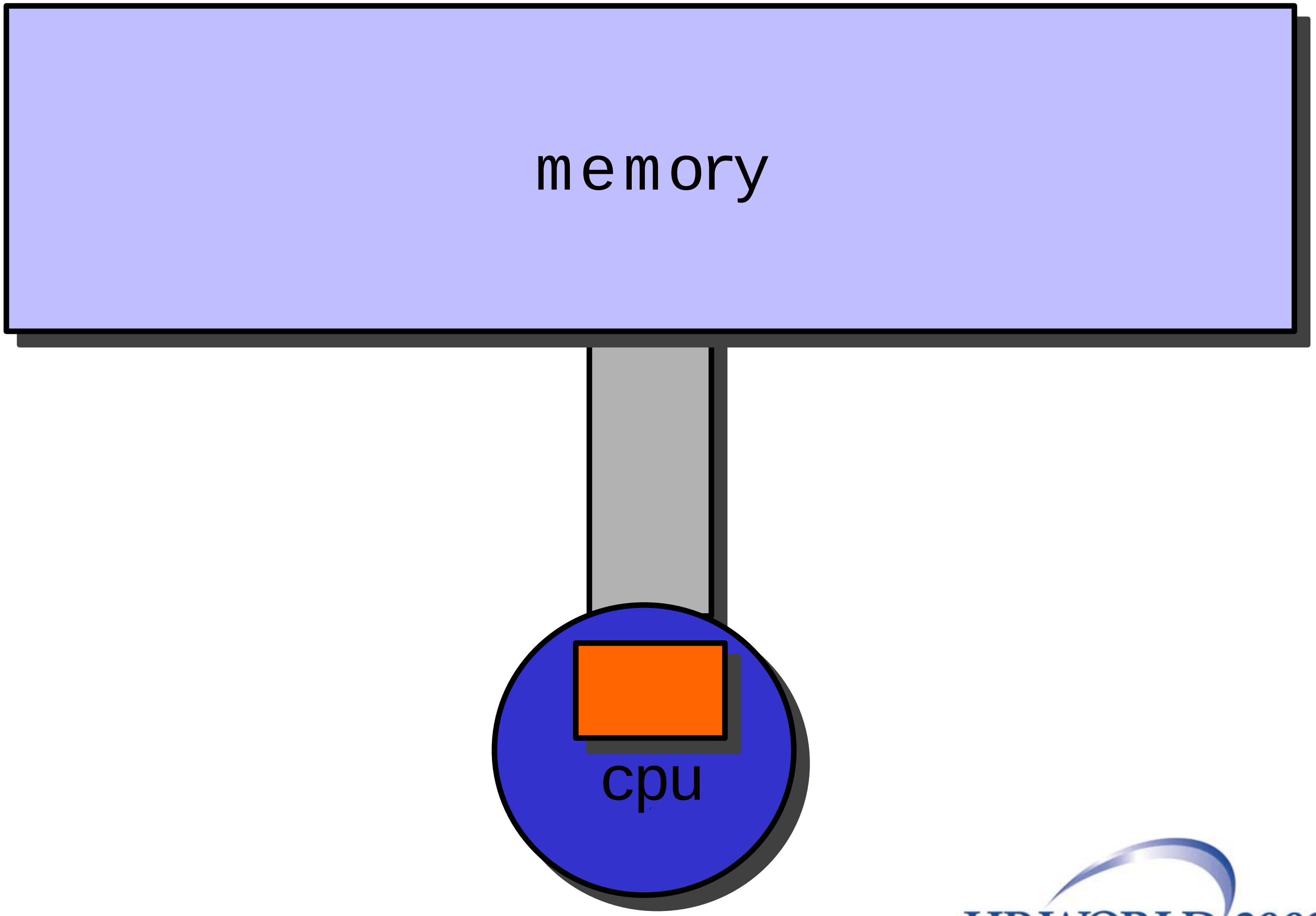
Track: HP-UX

Dr. Frank Baetke

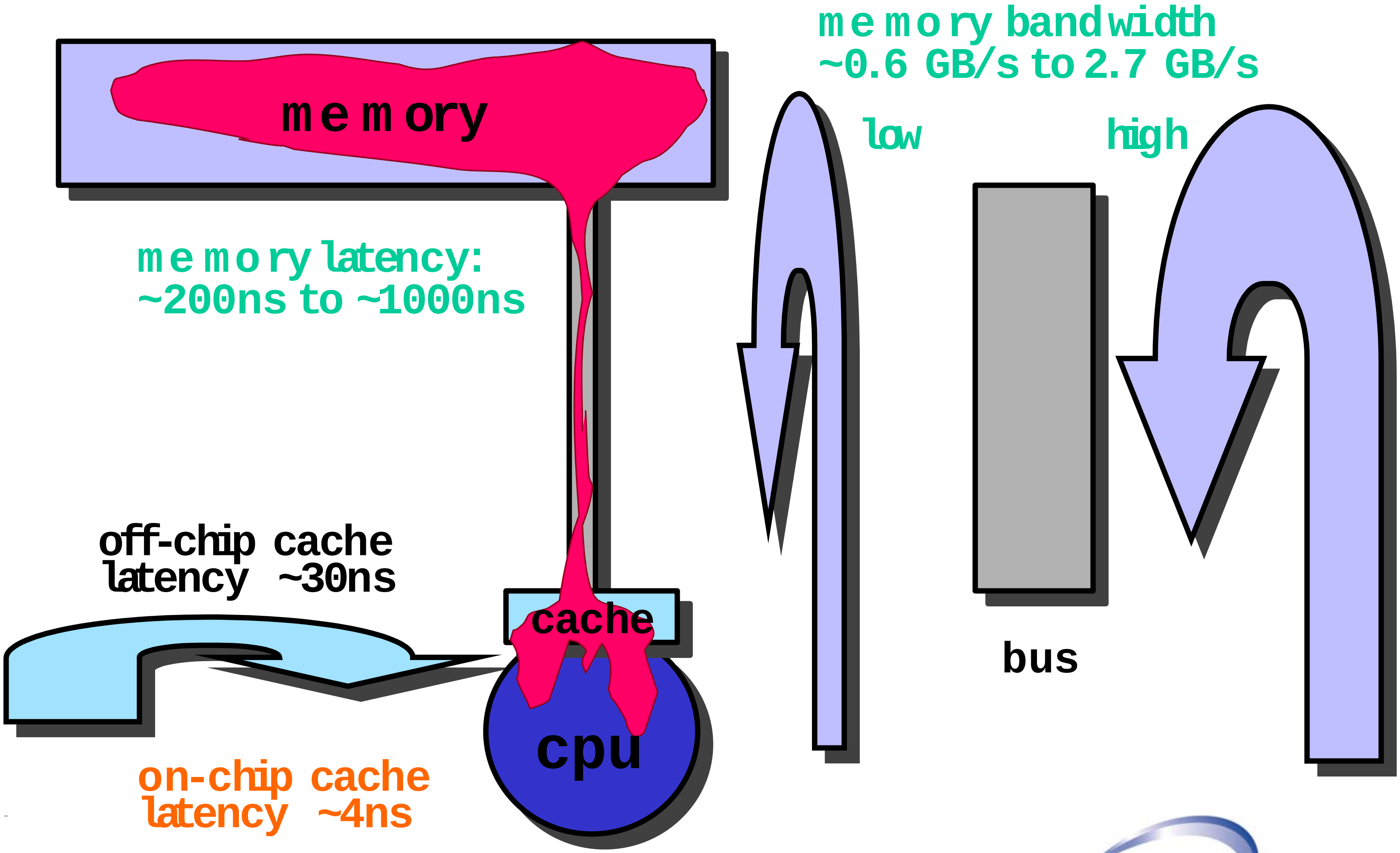
HP

basics

basic components/nomeclature



processors: bandwidth & latency



memory bandwidth
~0.6 GB/s to 2.7 GB/s

low

high

memory

memory latency:
~200ns to ~1000ns

off-chip cache
latency ~30ns

cache

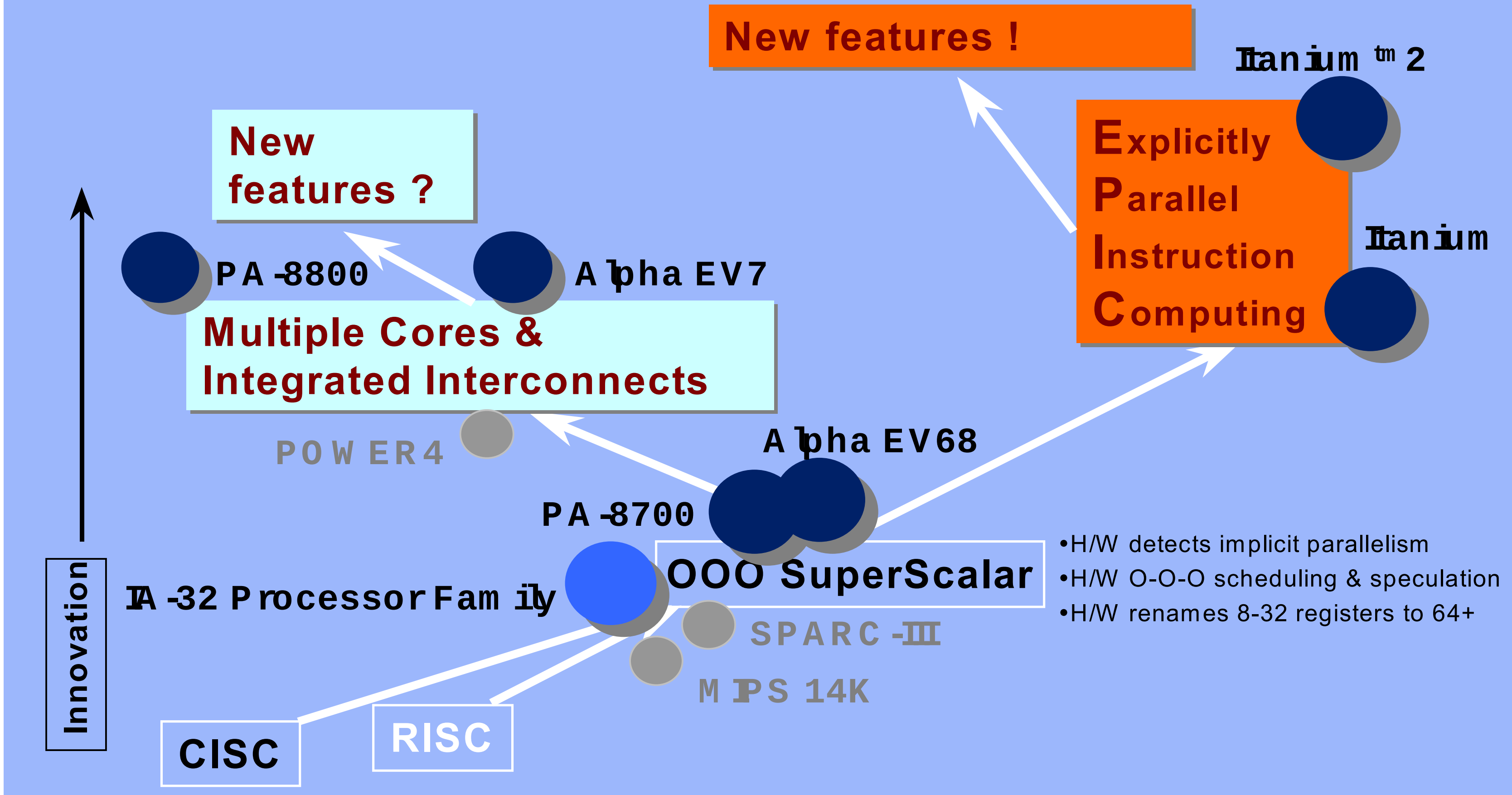
cpu

bus

on-chip cache
latency ~4ns

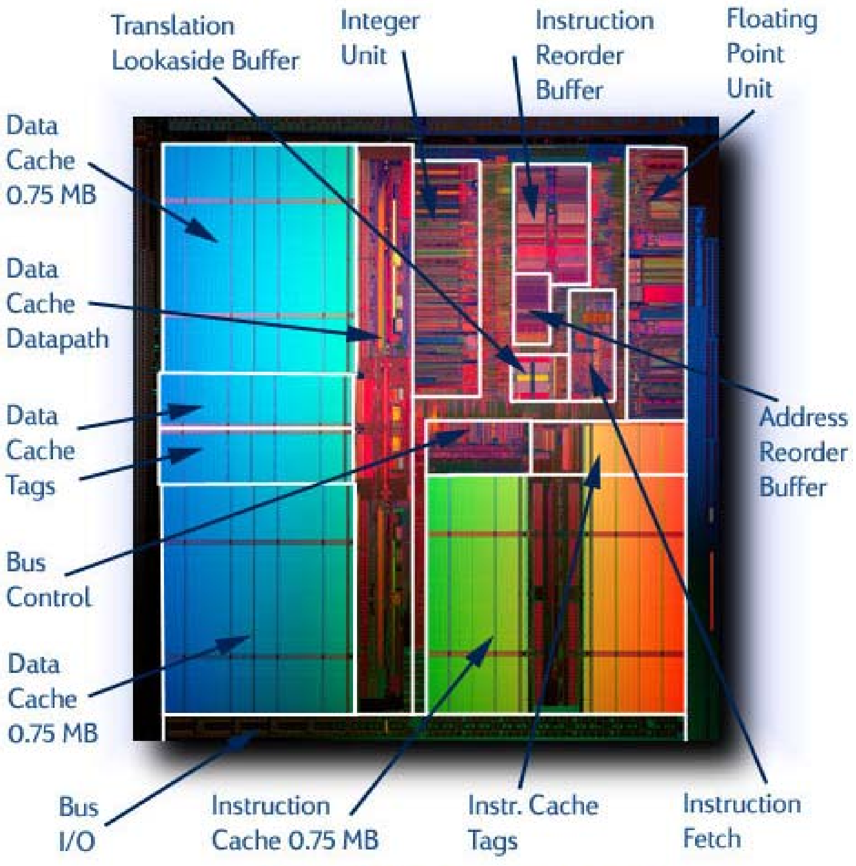
processors

next generation processor technology



- •H/W detects implicit parallelism
- •H/W O-O-O scheduling & speculation
- •H/W renames 8-32 registers to 64+

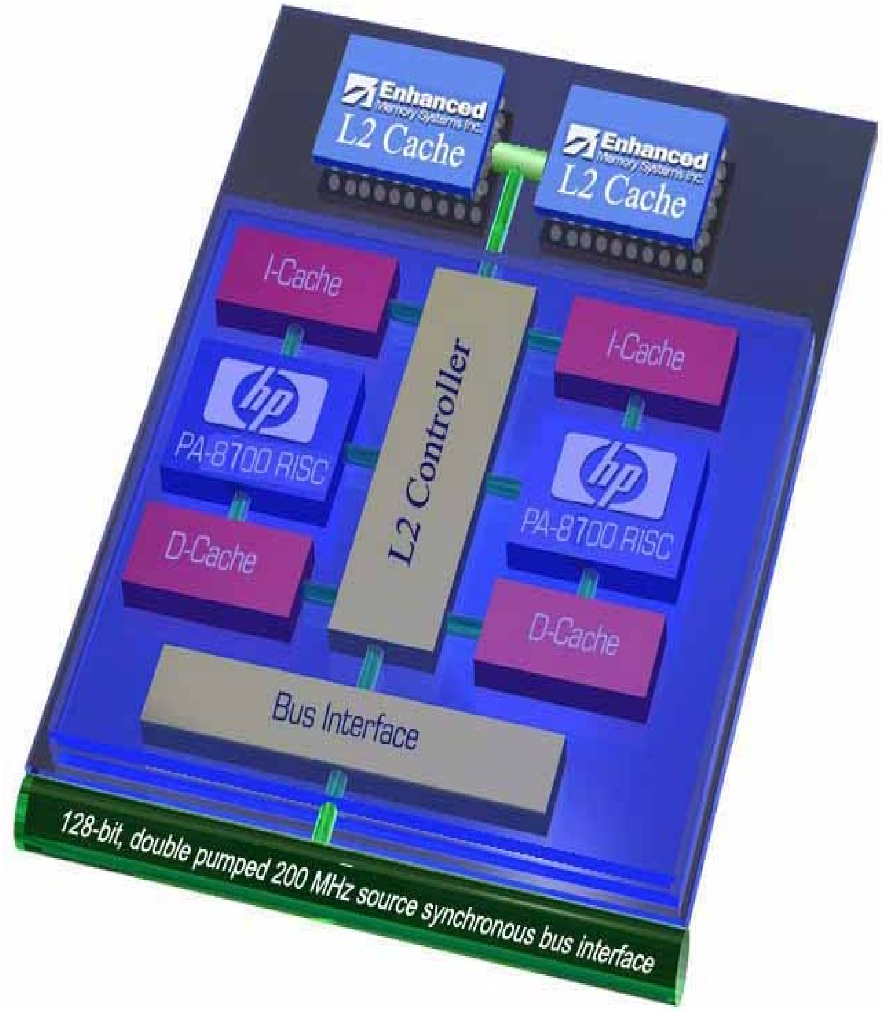
PA-8700



PA-8700 microprocessor

3 GFLOP/s RISC processor

Actually a PIM (a processor in memory)


Enhanced Memory Systems Inc.
L2 Cache
Enhanced Memory Systems Inc.
L2 Cache
I-Cache
L2 Controller
I-Cache
hp
PA-8700 RISC
hp
PA-8700 RISC
D-Cache
D-Cache
Bus Interface
128-bit, double pumped 200 MHz source synchronous bus interface

PA-8800 processor

Dual core technology

trend 1:

large on-chip caches

multi-core chips

new architecture Itanium®

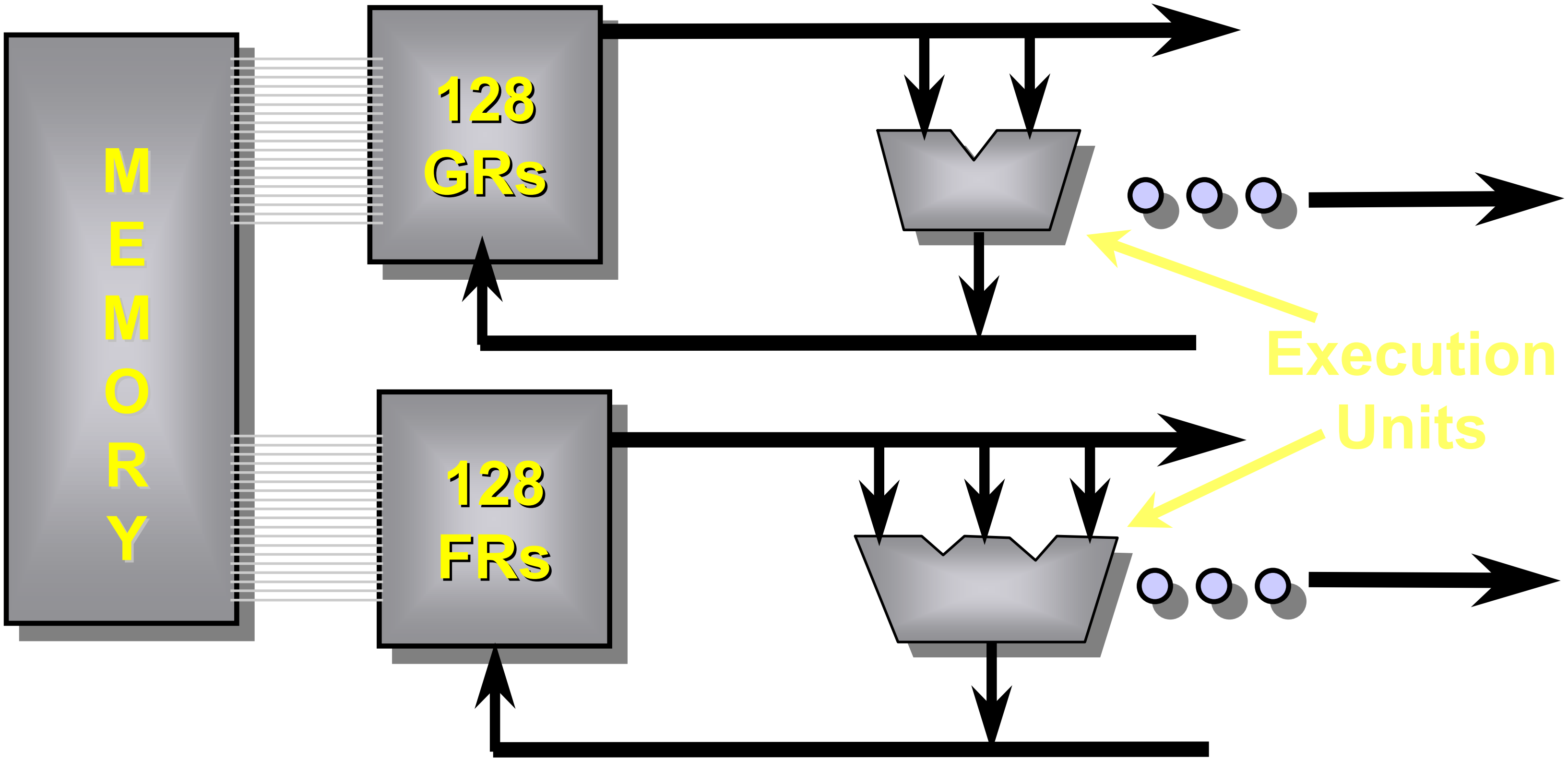
Itanium® architecture

formerly
IA-64, IPF

IA-64 (IPF) Key Features:

- Massive resources: 2* 128 64-bit+ registers.
  n* Integer Units, m*Floating-Point Units, lots of special registers for branches, predication, loop unrolling etc.
- Speculation: The processor can ‘pre-load’ data into the caches even if the access is potentially illegal - so it speculates that the data are valid. Correctness can later be checked in 1 cycle !
- Explicit Parallelization: The compiler ‘tells’ the processor what can be executed in parallel (and what requires sequential processing)
- Predication: The compiler tells the processor to run (for example) both parts of an ‘IF condition’ in parallel and then discard the ‘wrong part’.

Itanium: Principal CPU-Design

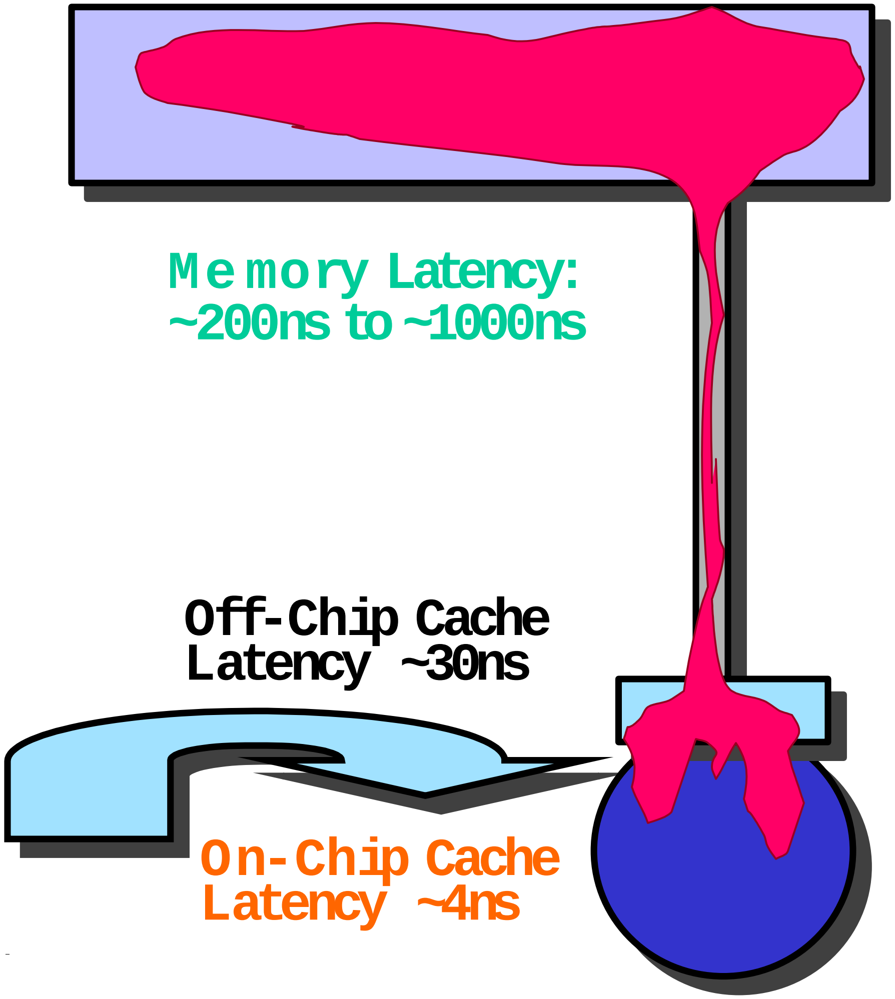


- Massively resourced - large register files
- Can use registers for ‘vector-like’ loops

IA-64 (IPF) Key Features:

- Massive resources: 2* 128 64-bit+ registers.
  n* Integer Units, m*Floating-Point Units, lots of special registers for branches, predication, loop unrolling etc.
- Speculation: The processor can ‘pre-load’ data into the caches even if the access is potentially illegal - so it speculates that the data are valid. Correctness can later be checked in 1 cycle !
- Explicit Parallelization: The compiler ‘tells’ the processor what can be executed in parallel (and what requires sequential processing)
- Predication: The compiler tells the processor to run (for example) both parts of an ‘IF condition’ in parallel and then discard the ‘wrong part’.

Speculation: preloading data into caches reduces (hides) latency



Memory Latency:
~200ns to ~1000ns

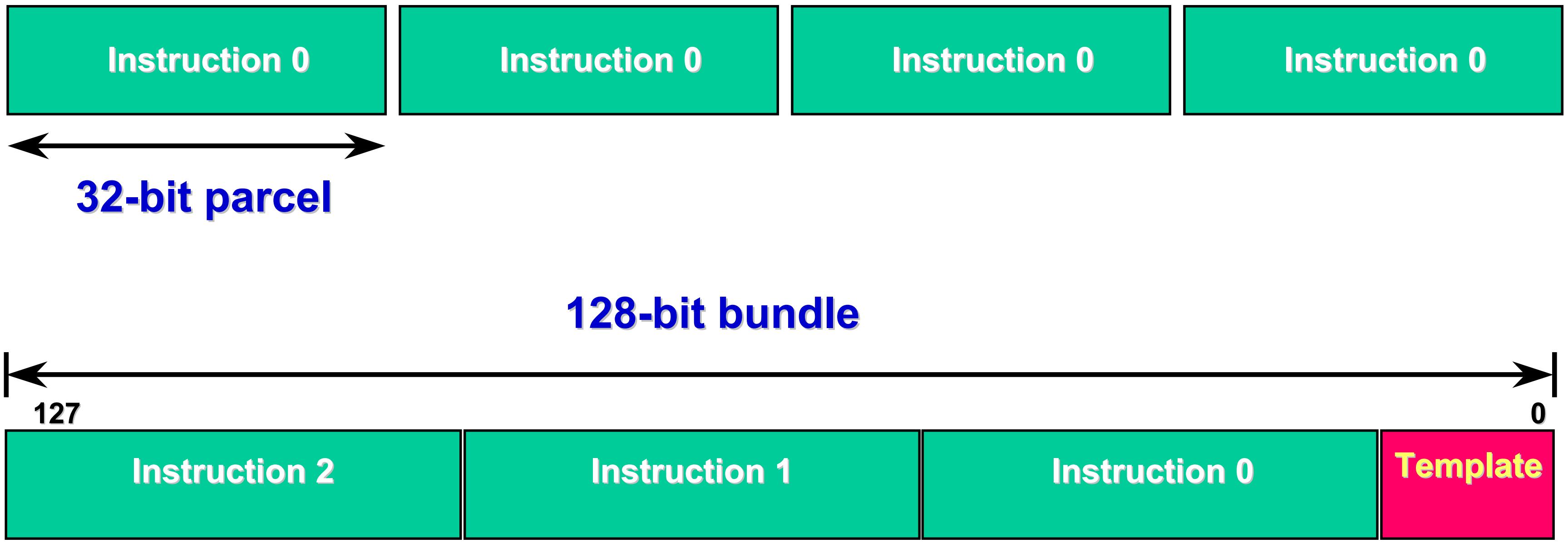
Off-Chip Cache
Latency ~30ns

On-Chip Cache
Latency ~4ns

IA-64 (IPF) Key Features:

- Massive resources: 2* 128 64-bit+ registers.
  n* Integer Units, m*Floating-Point Units, lots of special registers for branches, predication, loop unrolling etc.
- Speculation: The processor can ‘pre-load’ data into the caches even if the access is potentially illegal - so it speculates that the data are valid. Correctness can later be checked in 1 cycle !
- Explicit Parallelization: The compiler ‘tells’ the processor what can be executed in parallel (and what requires sequential processing)
- Predication: The compiler tells the processor to run (for example) both parts of an ‘IF condition’ in parallel and then discard the ‘wrong part’.

RISC vs. EPIC instruction format



Instruction 0

Instruction 0

Instruction 0

Instruction 0

32-bit parcel

128-bit bundle

127

0

Instruction 2

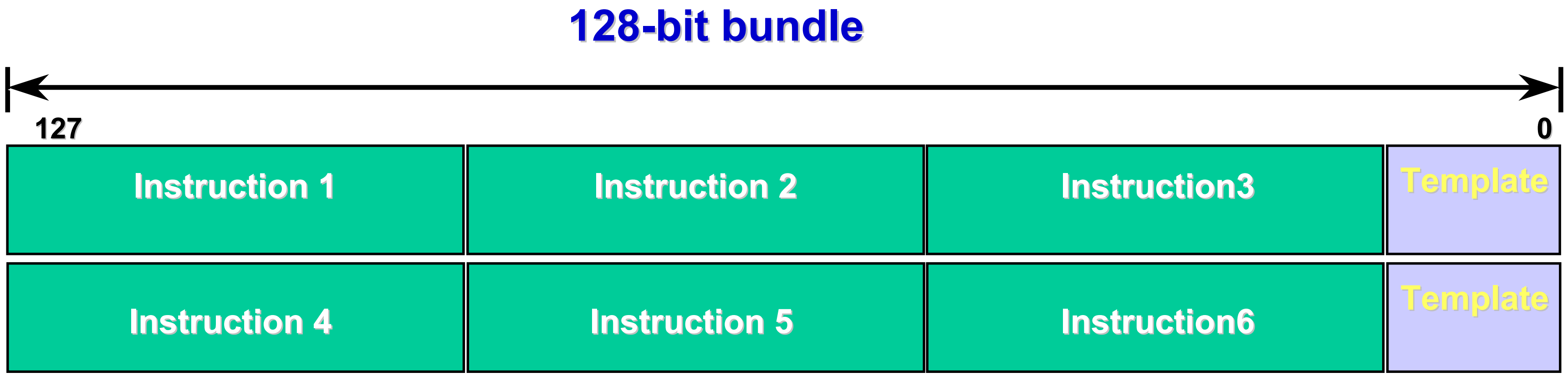
Instruction 1

Instruction 0

Template

The new instruction format enables scalability w/ compatibility

Itanium: explicit parallelism



128-bit bundle

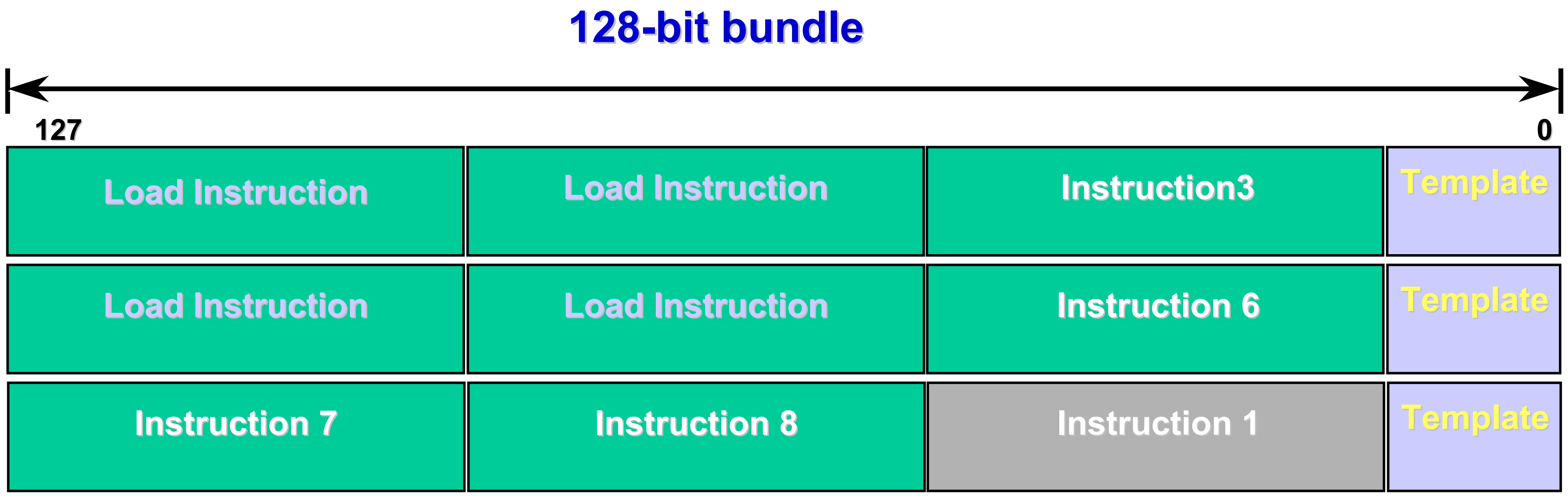
127

0

Instruction 1	Instruction 2	Instruction3	Template
Instruction 4	Instruction 5	Instruction6	Template

6 instructions/cycle on Itanium or McKinley

Itanium: explicit parallelism 2



128-bit bundle

Load Instruction	Load Instruction	Instruction3	Template
Load Instruction	Load Instruction	Instruction 6	Template
Instruction 7	Instruction 8	Instruction 1	Template

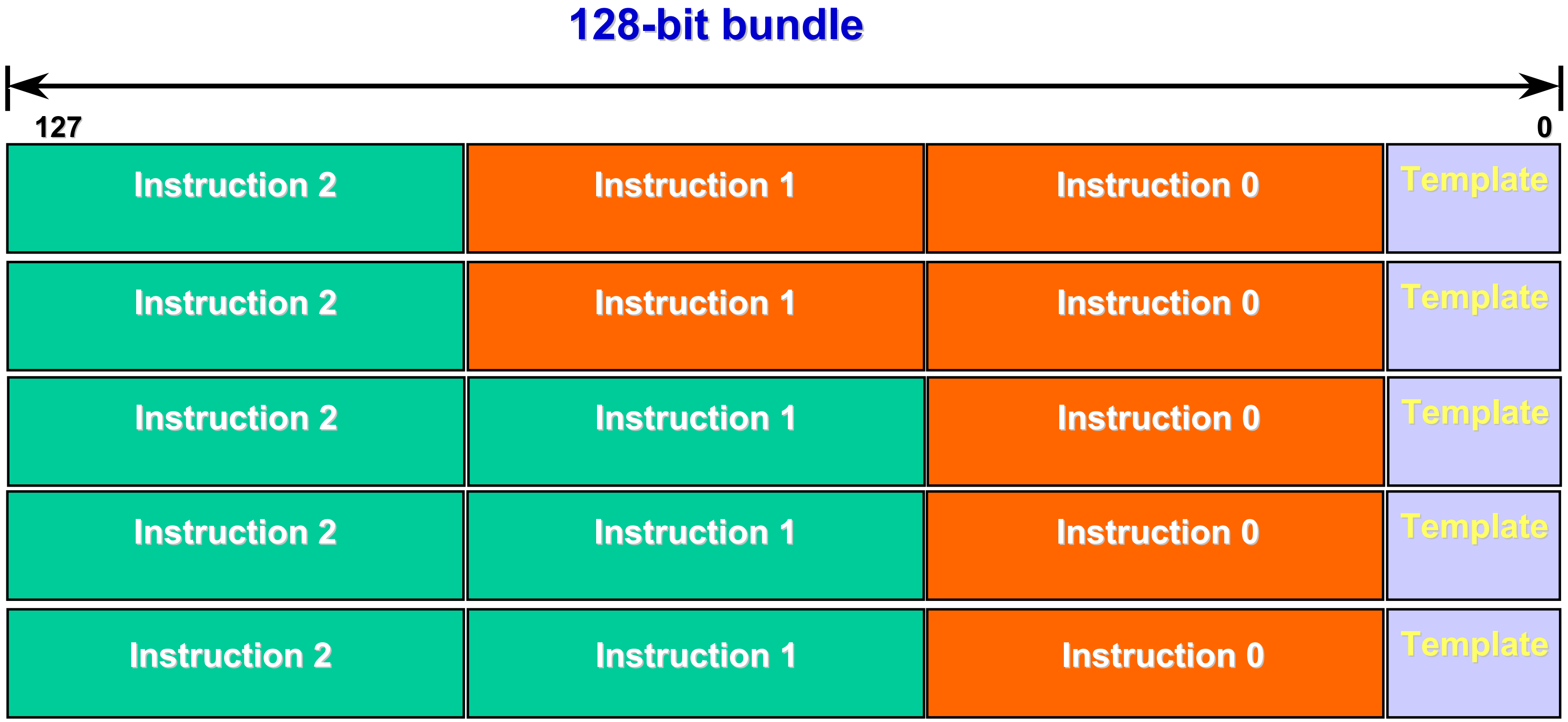
Itanium2 (ex McKinley) and future processors will allow 4 loads/cycle

IA-64 (IPF) Key Features:

- **Massive resources: 2* 128 64-bit+ registers. n* Integer Units, m*Floating-Point Units, lots of special registers for branches, predication, loop unrolling etc.**
- **Speculation: The processor can 'pre-load' data into the caches even if the access is potentially illegal - so it speculates that the data are valid. Correctness can later be checked in 1 cycle !**
- **Explicit Parallelization: The compiler 'tells' the processor what can be executed in parallel (and what requires sequential processing)**
- **Predication: The compiler tells the processor to run (for example) both parts of an 'IF condition' in parallel and then discard the 'wrong part'.**

IA-64: Predication

Run the THEN and ELSE part in parallel !



🟩 Instruction 2	🟧 Instruction 1	🟧 Instruction 0	Template
🟩 Instruction 2	🟧 Instruction 1	🟧 Instruction 0	Template
🟩 Instruction 2	🟩 Instruction 1	🟧 Instruction 0	Template
🟩 Instruction 2	🟩 Instruction 1	🟧 Instruction 0	Template
🟩 Instruction 2	🟩 Instruction 1	🟧 Instruction 0	Template

Predication allows to eliminate branches

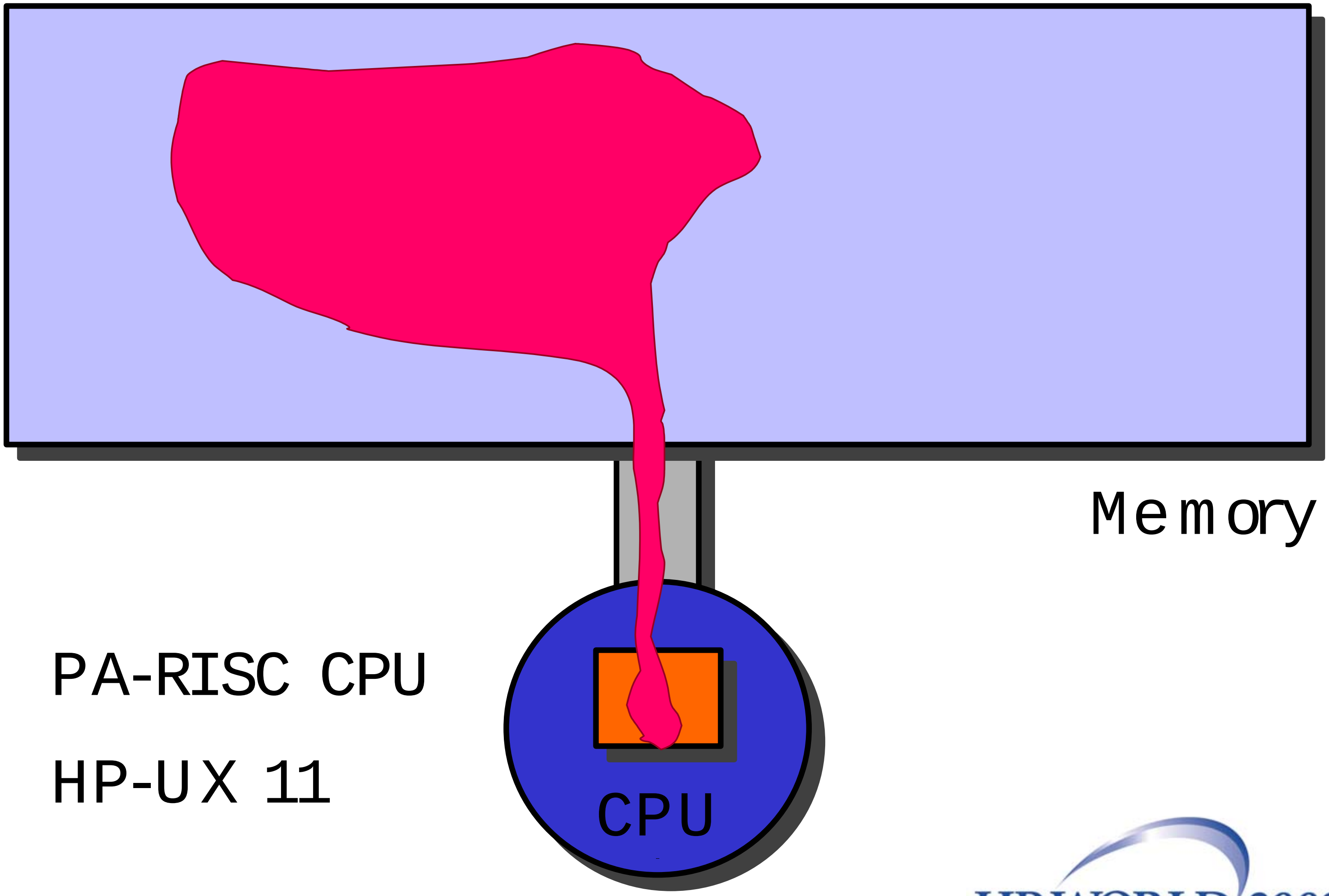
PA-RISC
is the *only*
RISC
architecture
compatible to
IA-64

The IA-64 instructions set has been designed with pa-risc instructions in mind

IA-64 processors can:

- run pa-risc native code (under HP-UX 11)
- run IA-64 native code (hp-ux, linux, …)
- run IA-32 native code (Win64)

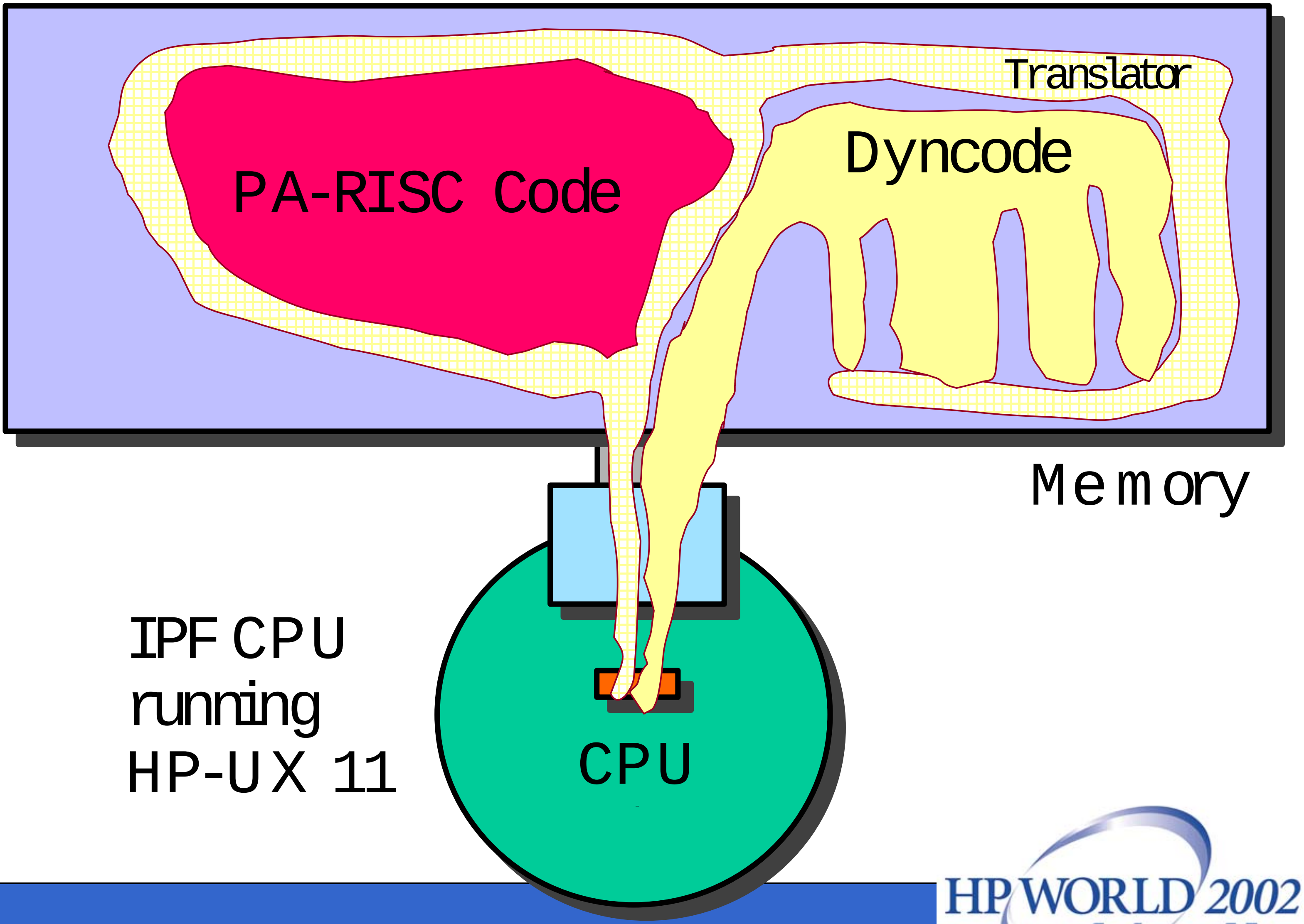
running PA-RISC code on Itanium



PA-RISC CPU

HP-UX 11

running PA-RISC code on IPF



IPF CPU
running
HP-UX 11

trend 2:

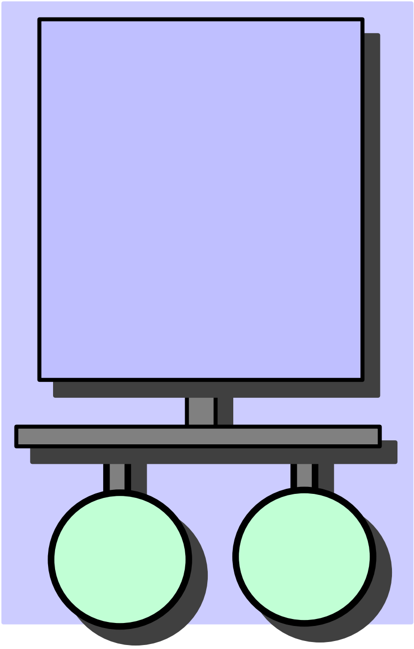
very few instruction set architectures will survive

(no applications on 'exotic' architectures !)

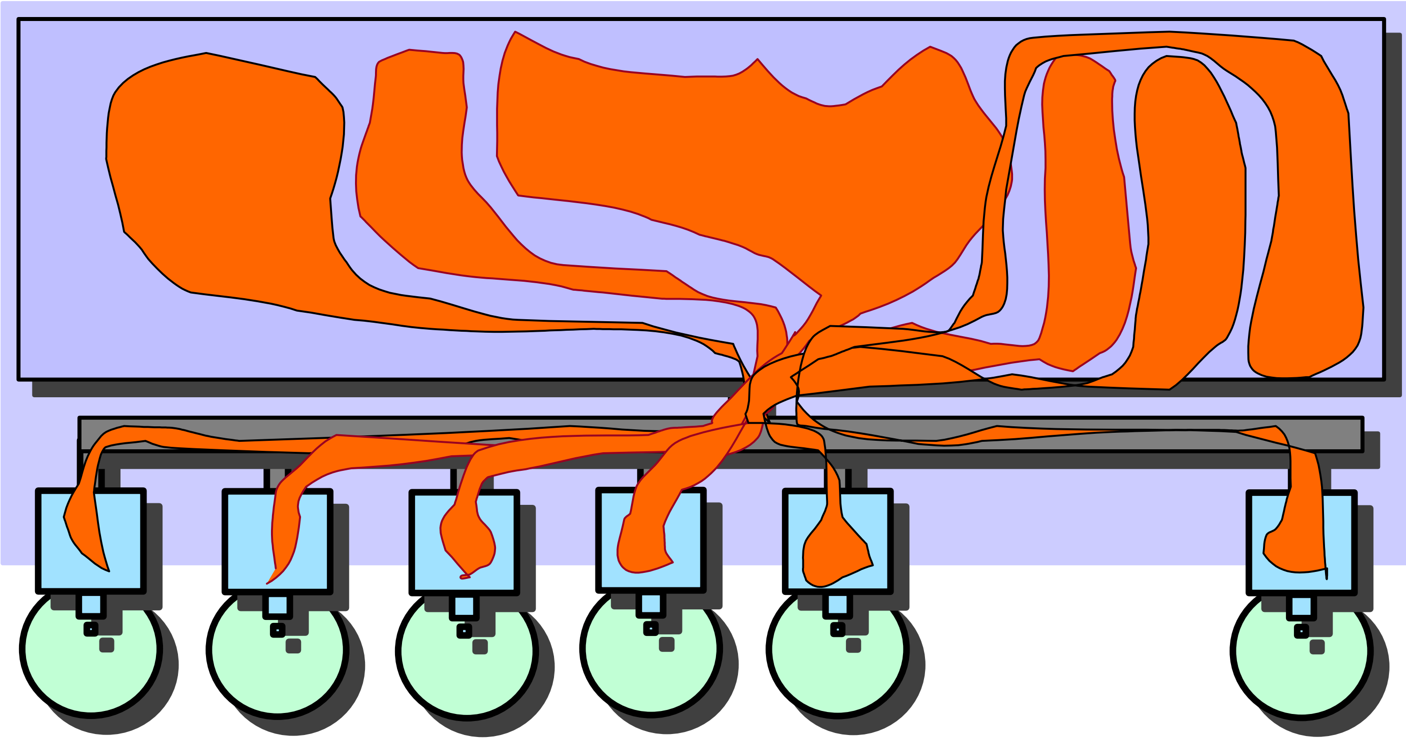
servers

A small server with 2 CPUs

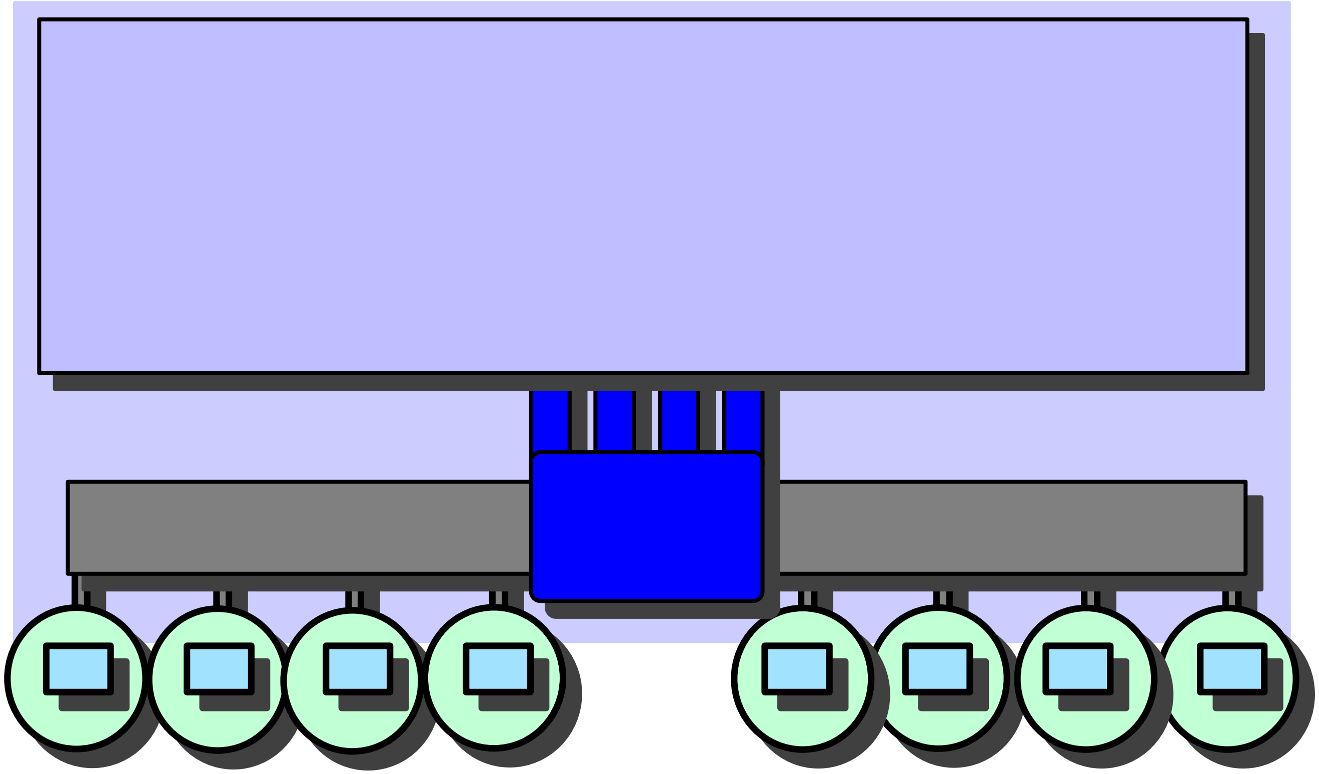
one address space – one copy of HP-UX 11.x (J-Class)



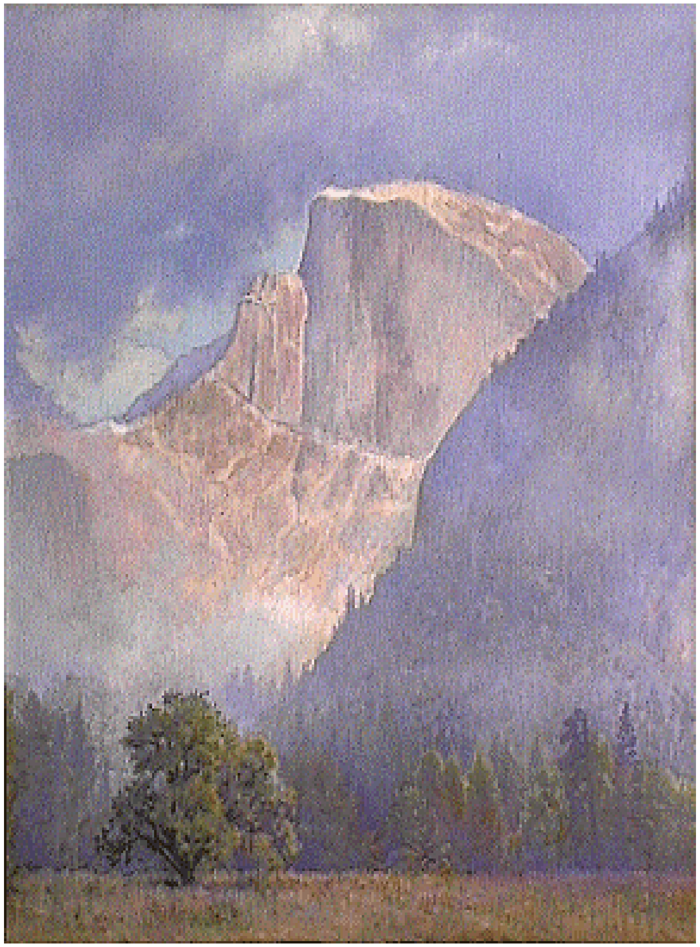
SMP advantage: big & small jobs, but the bus may become a bottleneck ….



rp7400 - a hybrid design (still SMP)



Half Dome / Super Dome

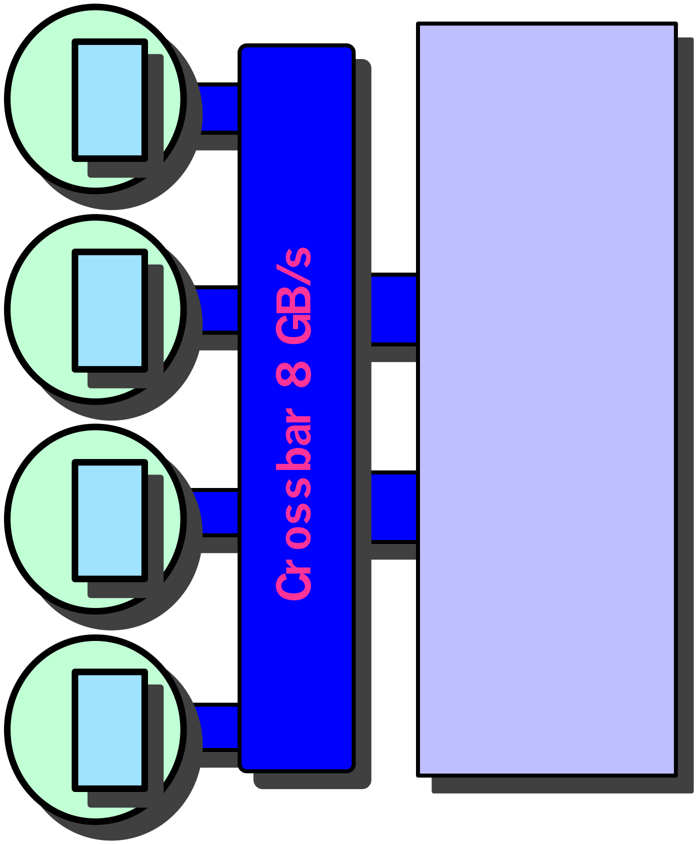


SuperDome

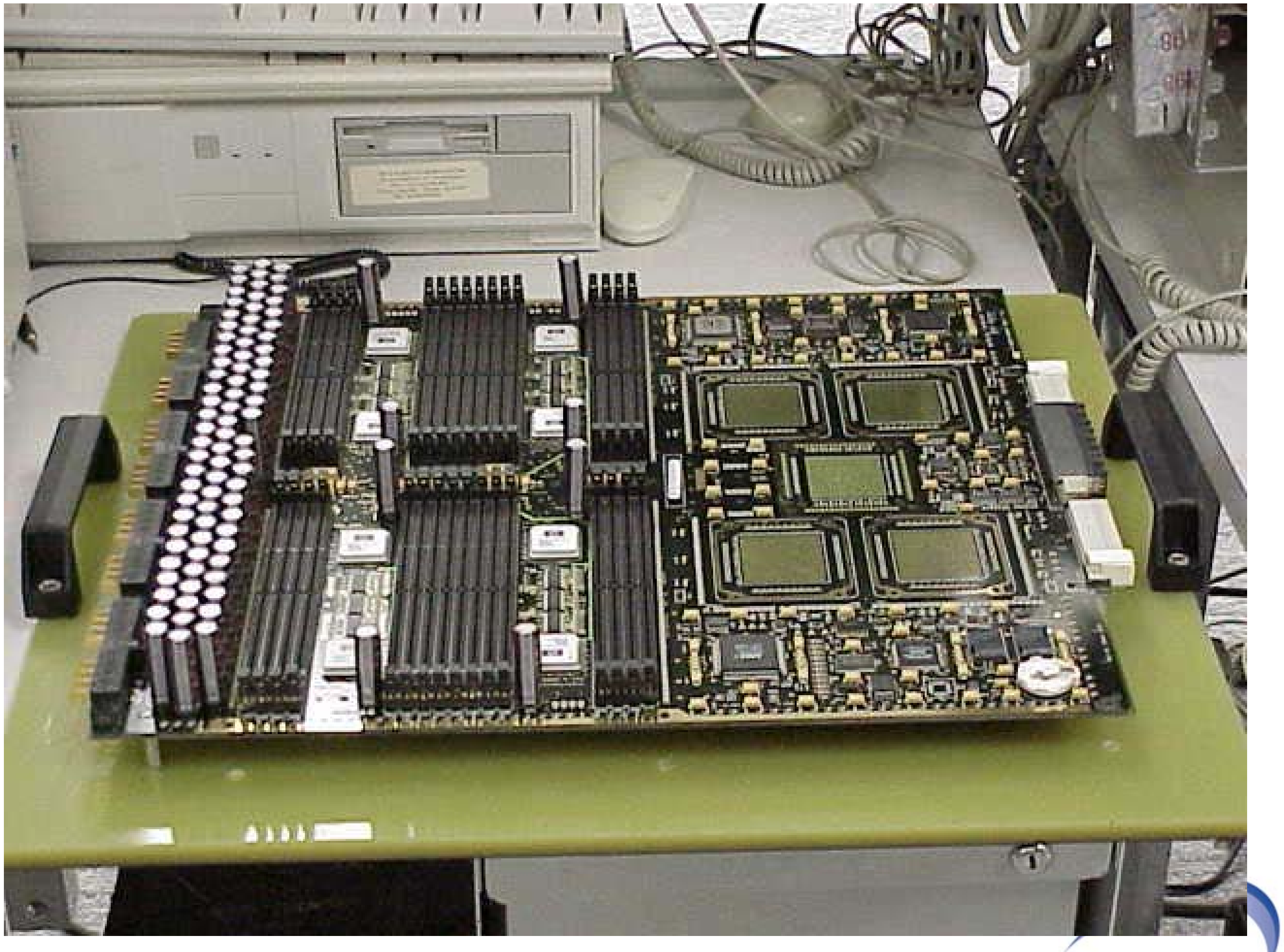
- Scalable Architecture
  - to 64
  - to .5 TB Physical Memory
  - to 192 PCI Busses
- HP-UX 11i
  - resource partitioning
  - near fault-tolerant capabilities
  - on-line add/replace for CPUs, memory and I/O components
- Processor support
  - PA8600 introduction
  - PA-RISC & IA64 roadmap

Cell - A Modular Building Block

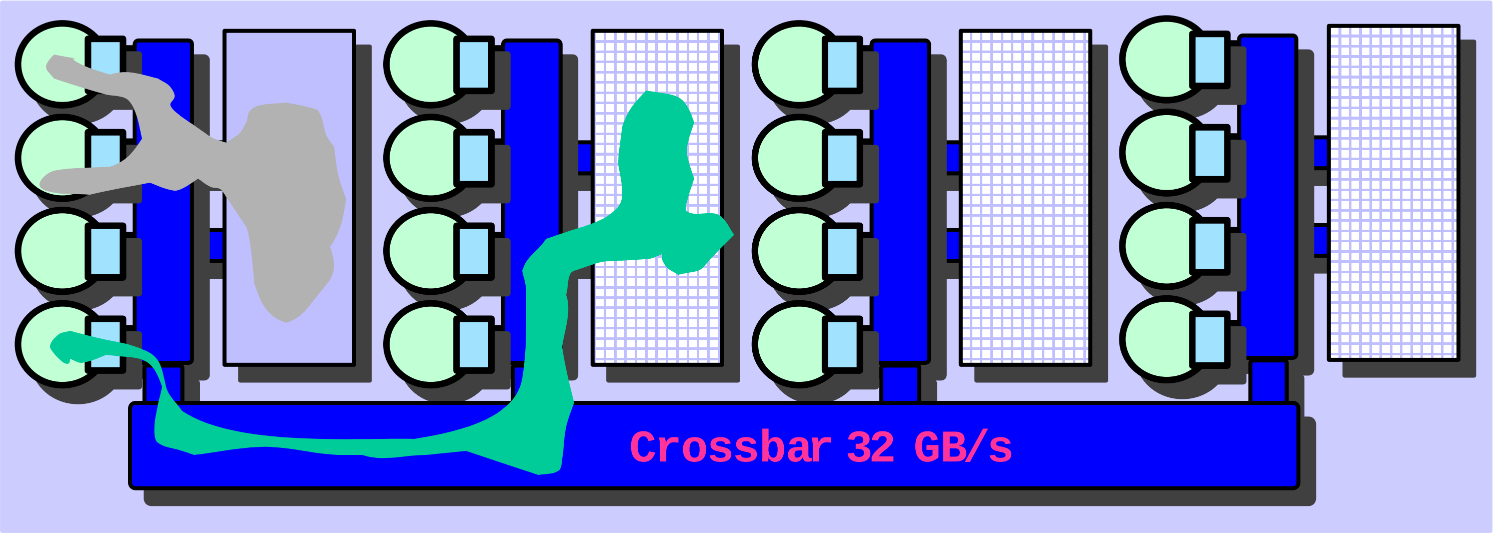
- Up to 4 processors
- Up to 16/32 GB memory (4 carriers x 8 GB ea)
- IO channel supports 16 PCI busses, 16 slots
- Cell crossbar provides 8 GB/s
- Cell I/O bandwidth is 1.8 GB/s



SuperDome Cell Board



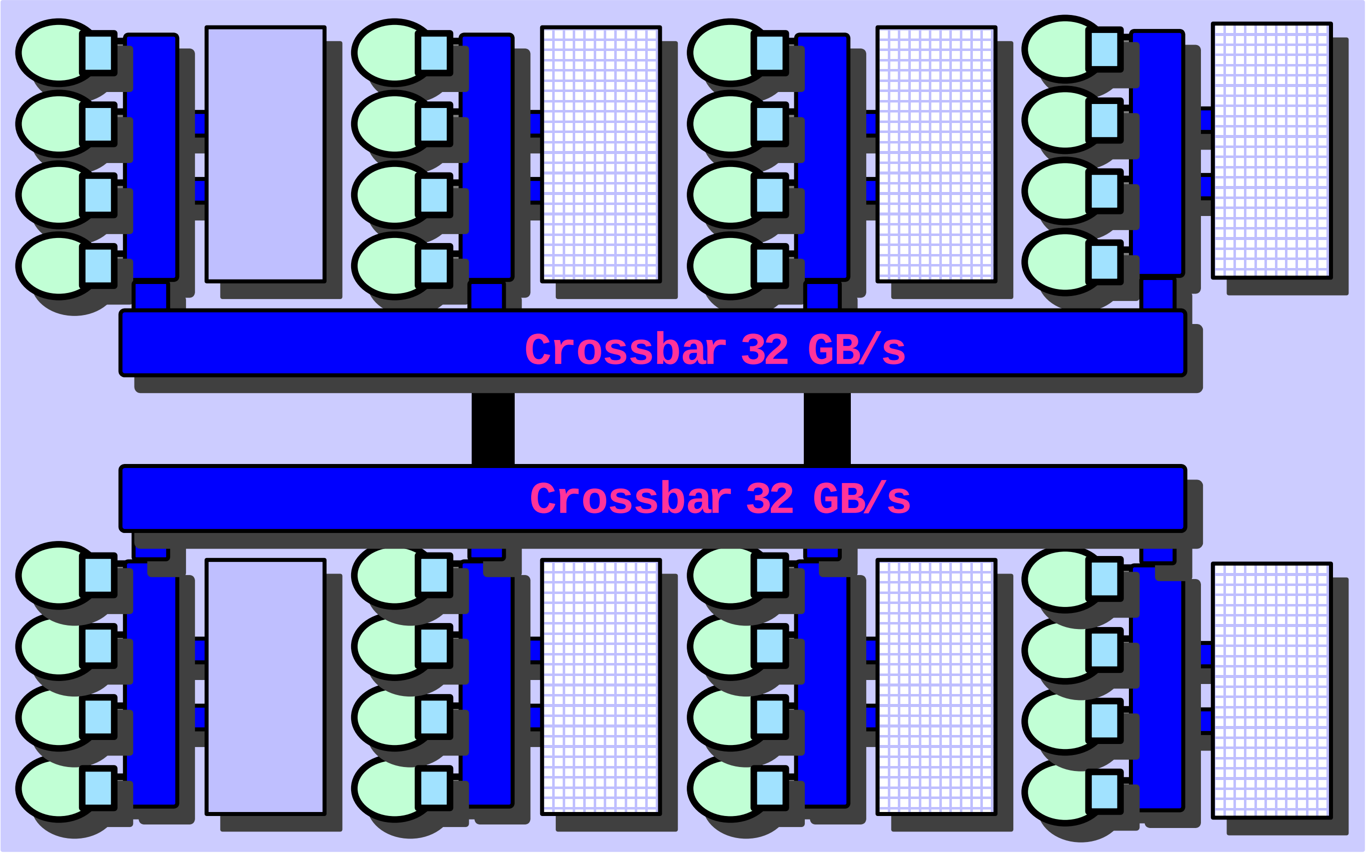
SuperDome /rp8400 (16 CPUs)



server rp8400

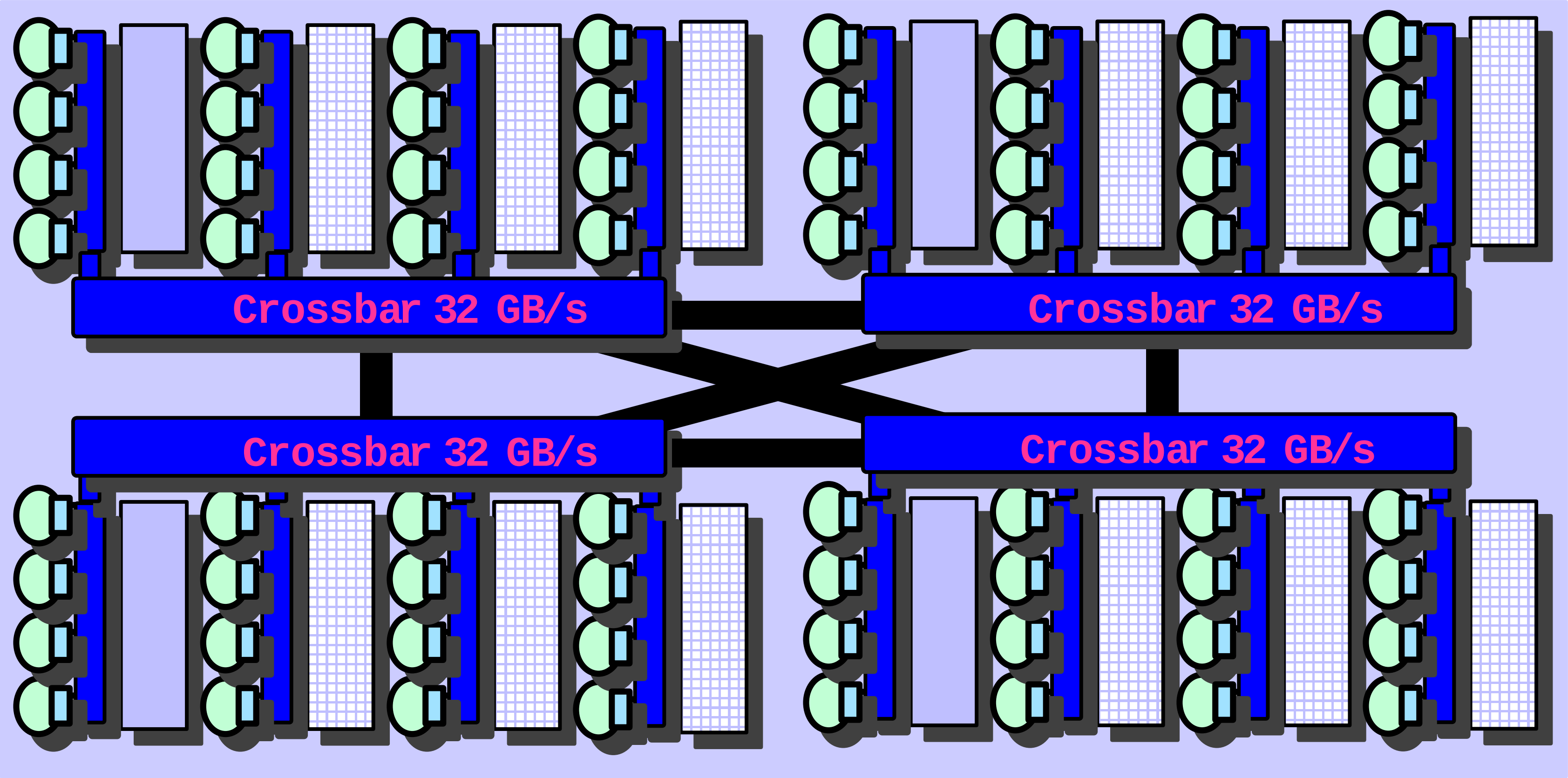
SuperDome (32 CPUs)

Latencies: 208, 301 and 357 ns



SuperDome (64 CPUs)

Latencies: 208, 301 and 357 ns



trend 3:

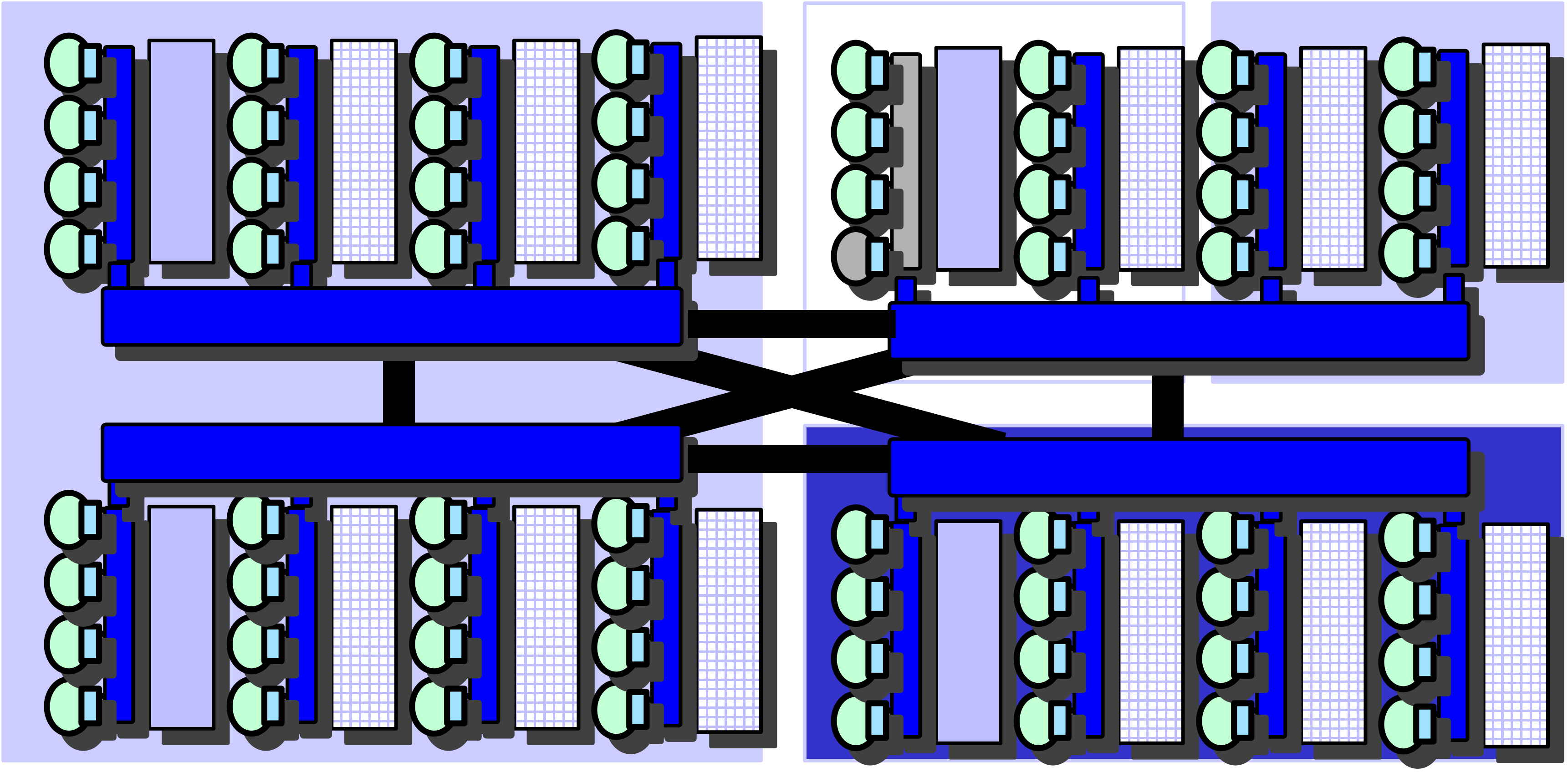
SMPs with crossbars

cells & ccNUMA

inherent upgradability

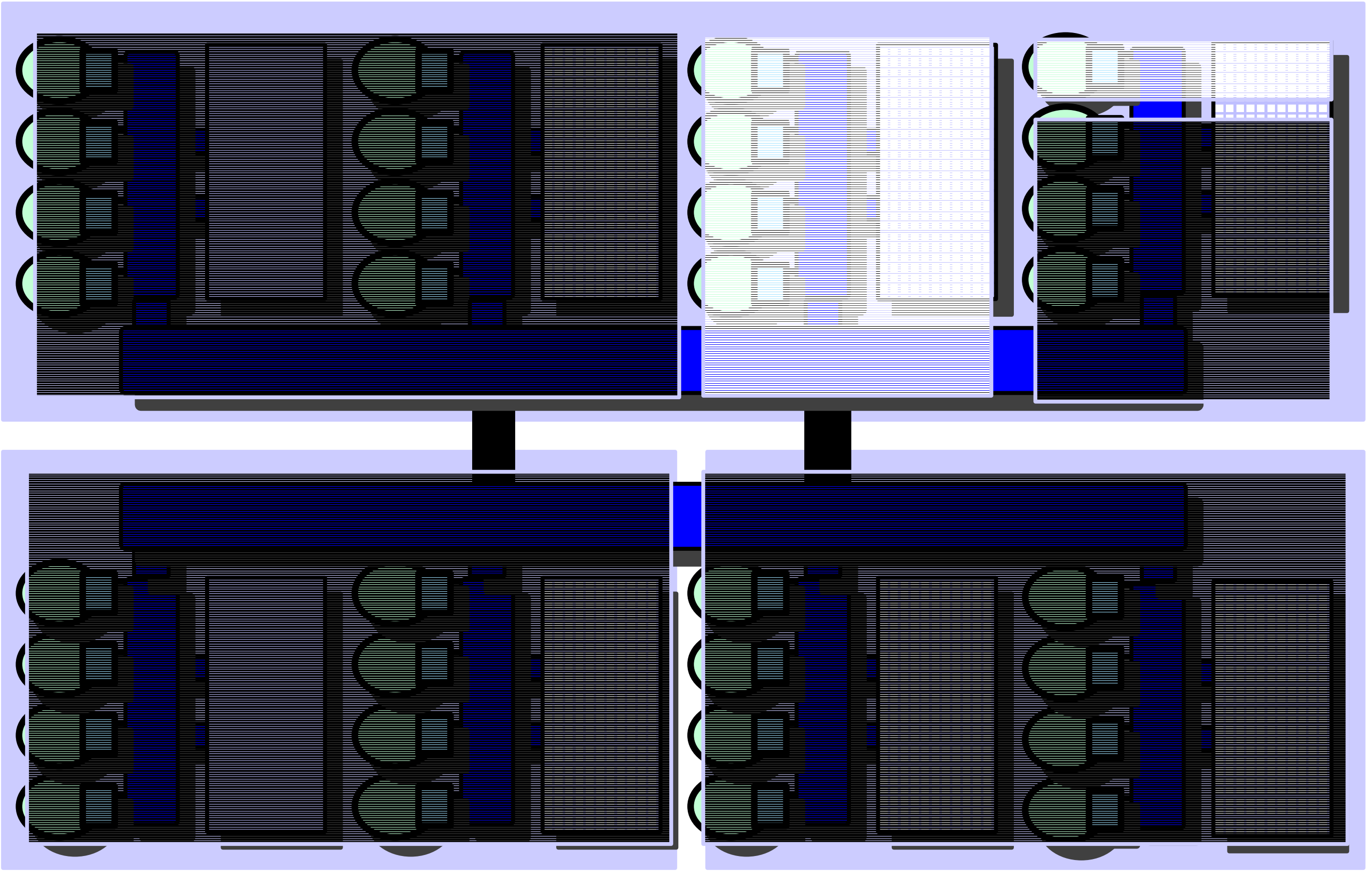
partitions

4 nPartitions (1 x 32, 1 x 16, 2 x 8 CPUs)



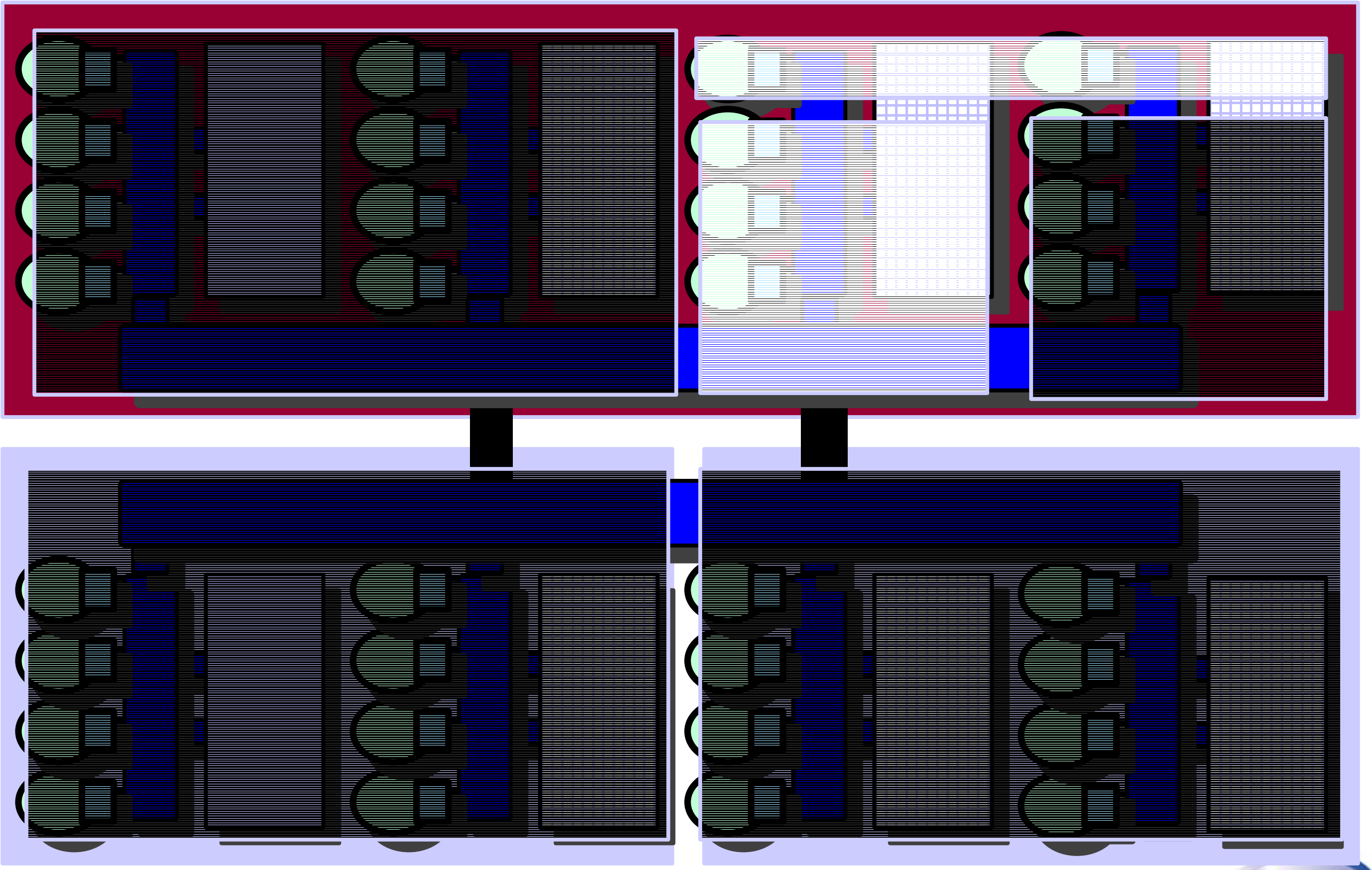
SuperDome (32 CPUs) & virtual partitions

3 nPartitions (16, 8, 8 CPUs), 6 Copies of HP-UX

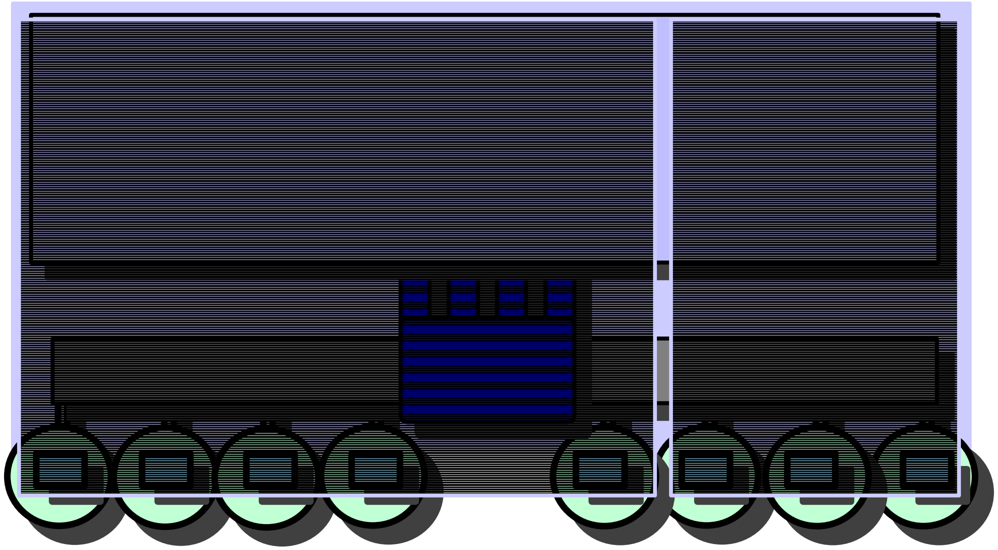


SuperDome (32 CPUs) & virtual partitions

3 nPartitions (16, 8, 8 CPUs), 6 Copies of HP-UX



rp7400 – with virtual partitions (2 x HP-UX)



trend 4:

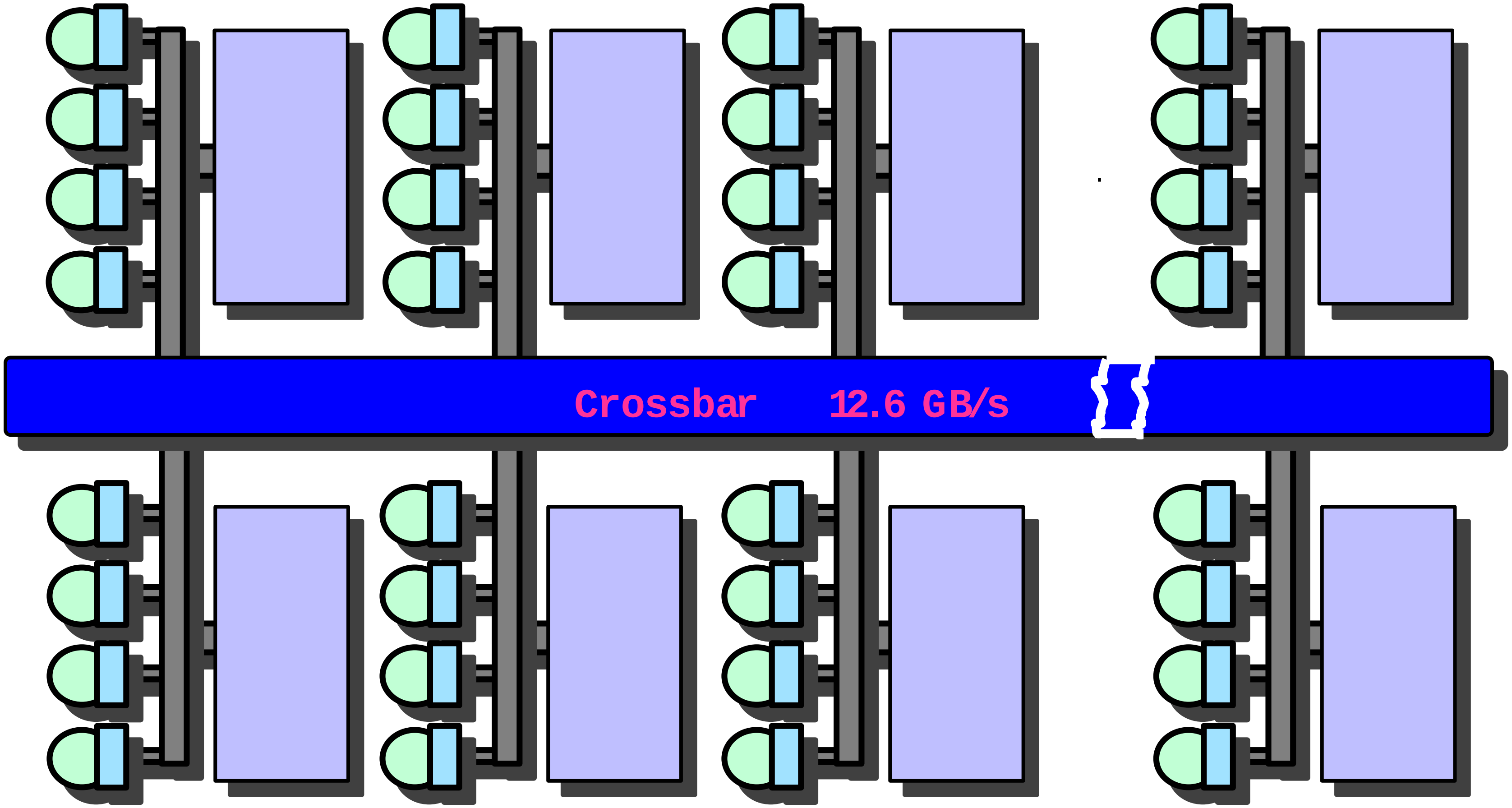
partitioning

- GRIDS
- clusters
- hard partitions
- virtual partitions
- schedulers

Example:
SUN’S UE10000

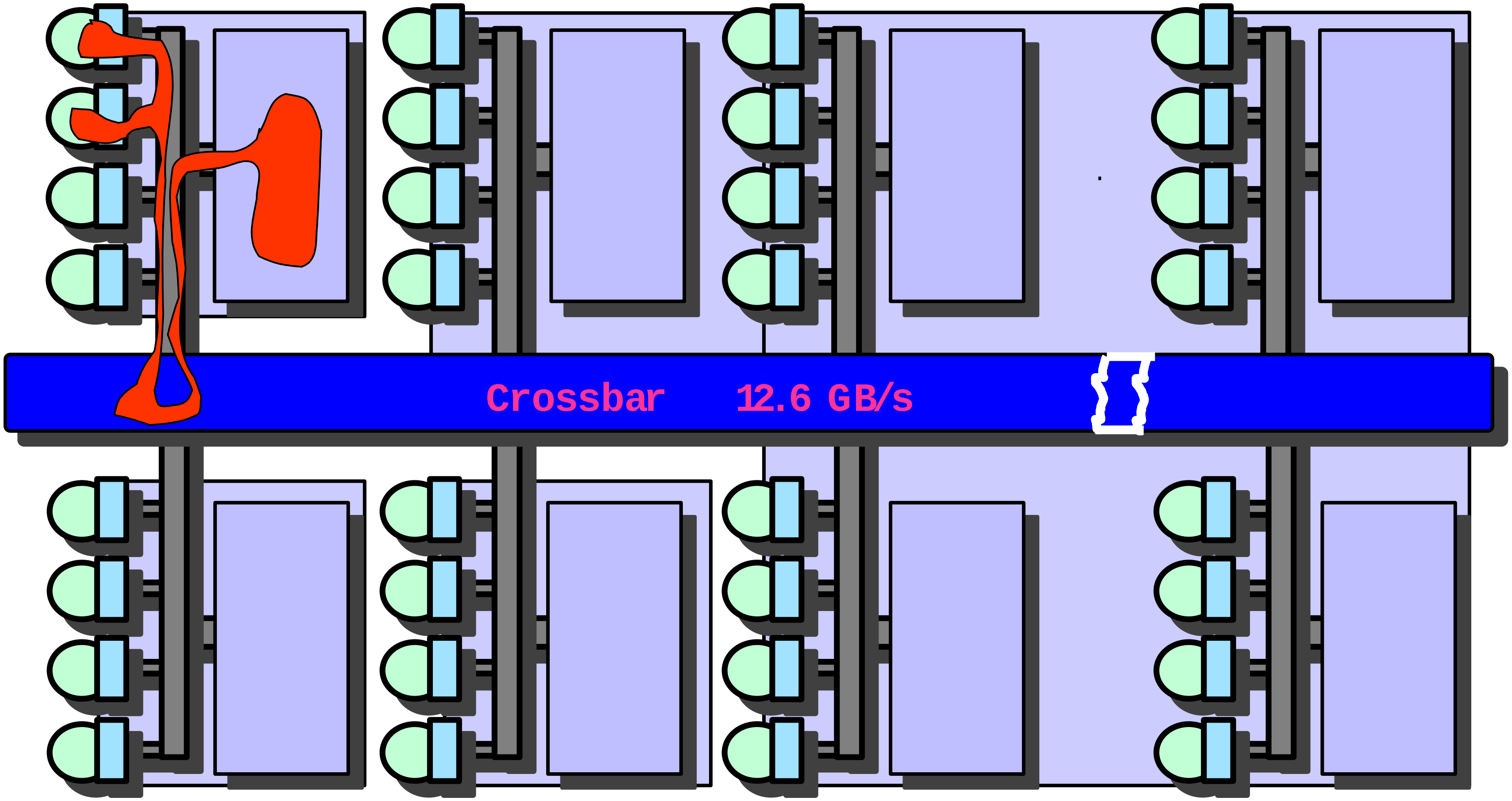
(not a cell-based architecutre)

UE10000: A Special SMP: 560 ns latency



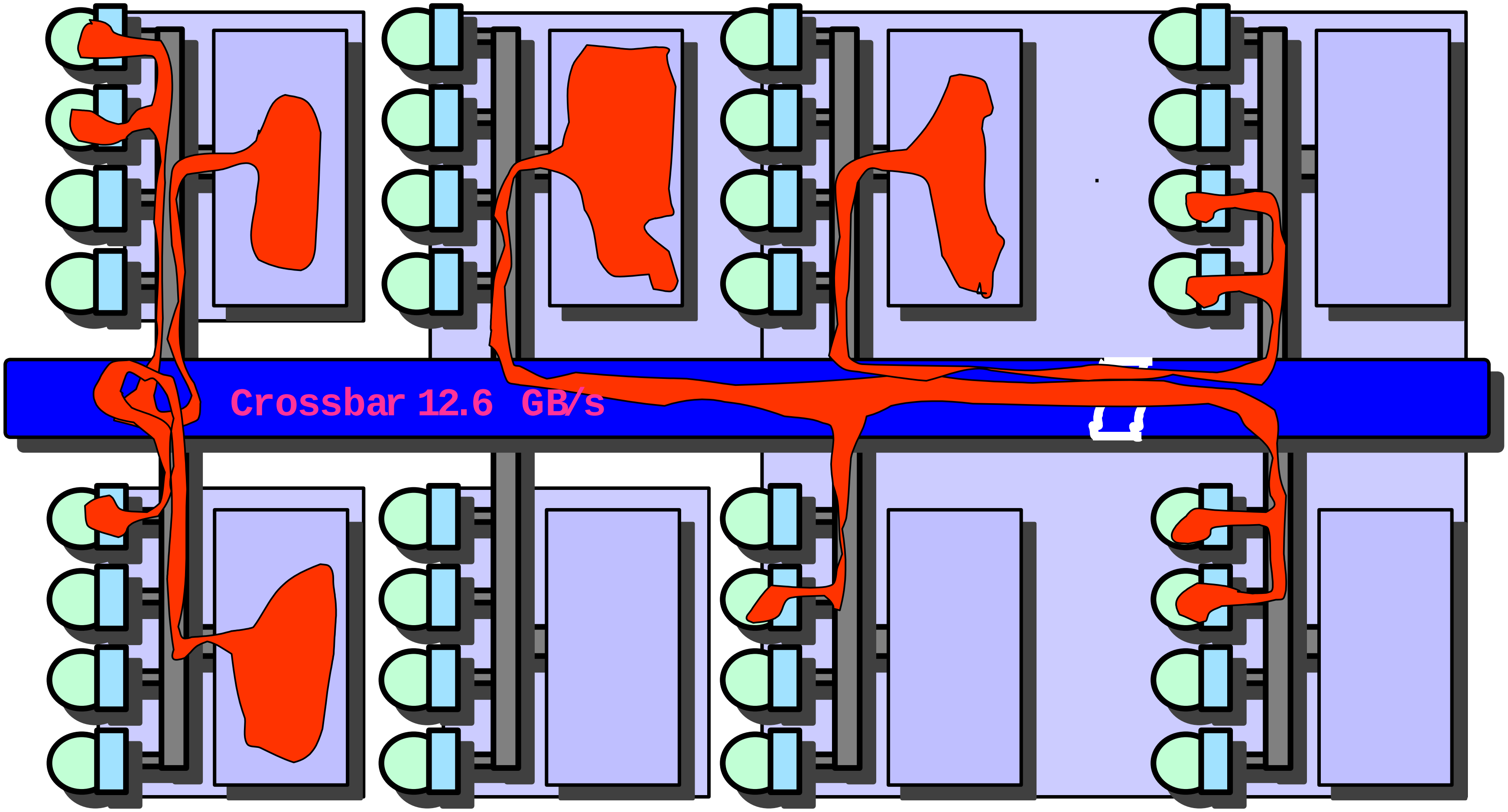
All traffic goes through one crossbar !!!

560 ns latency (SuperDome has 200 – 350ns)



-

Loading multiple domains - one left



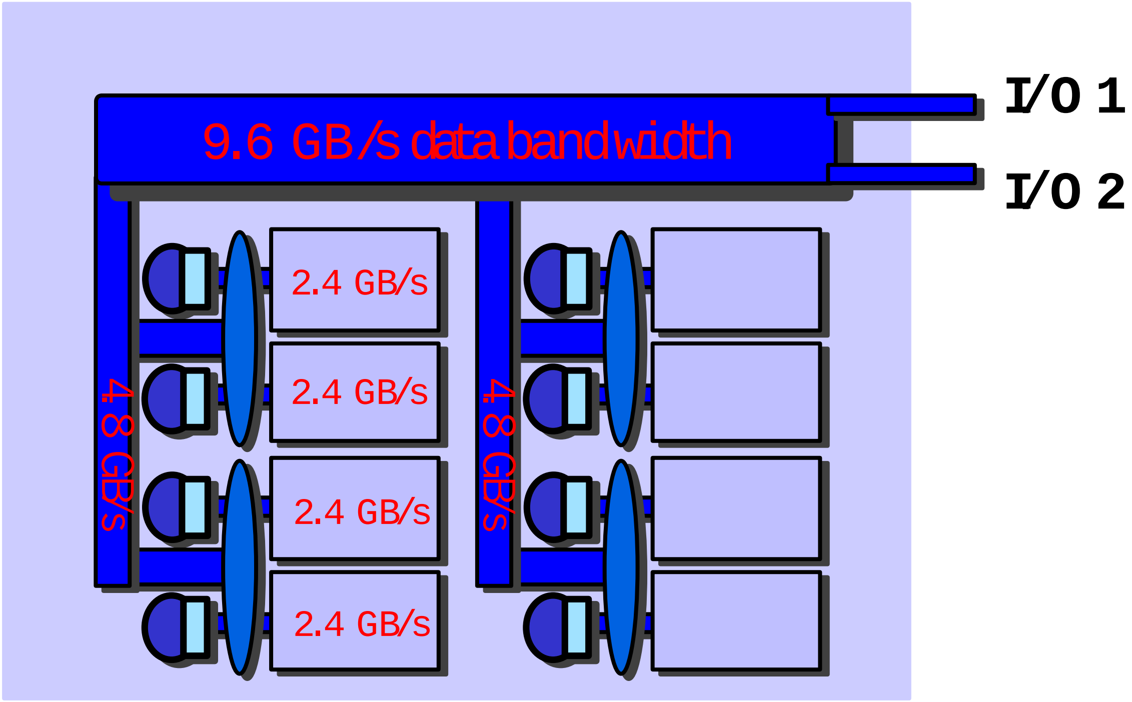
-

Example:

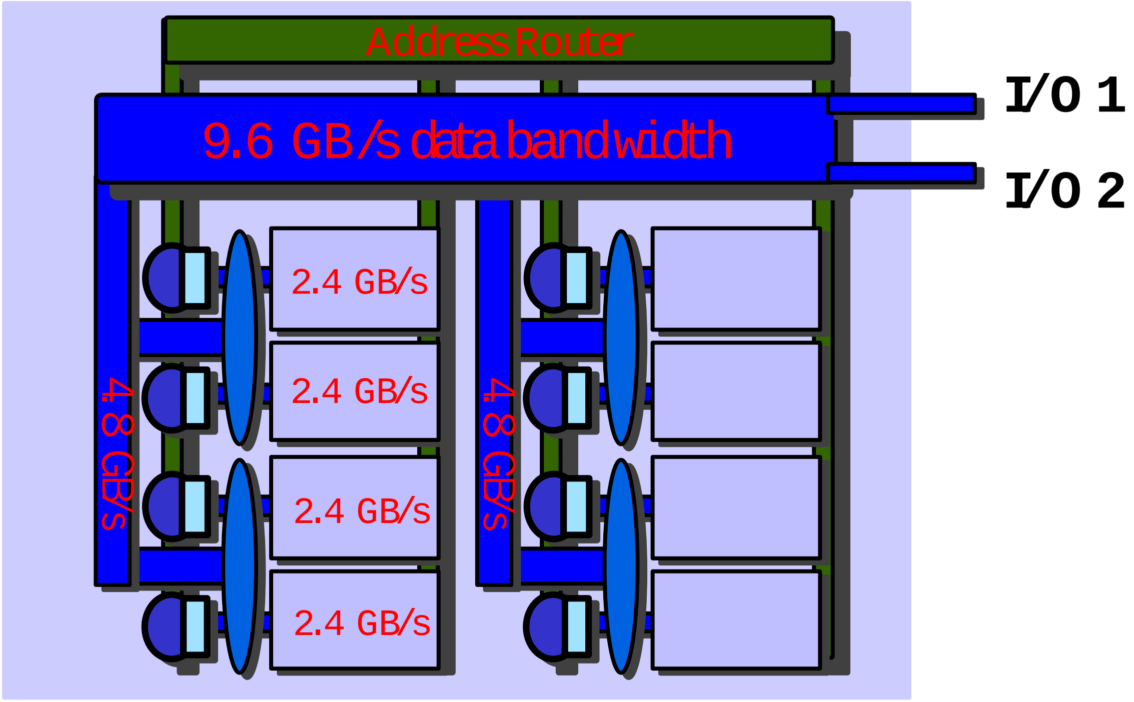
SUN's MidFrames

(cell-based architectures)

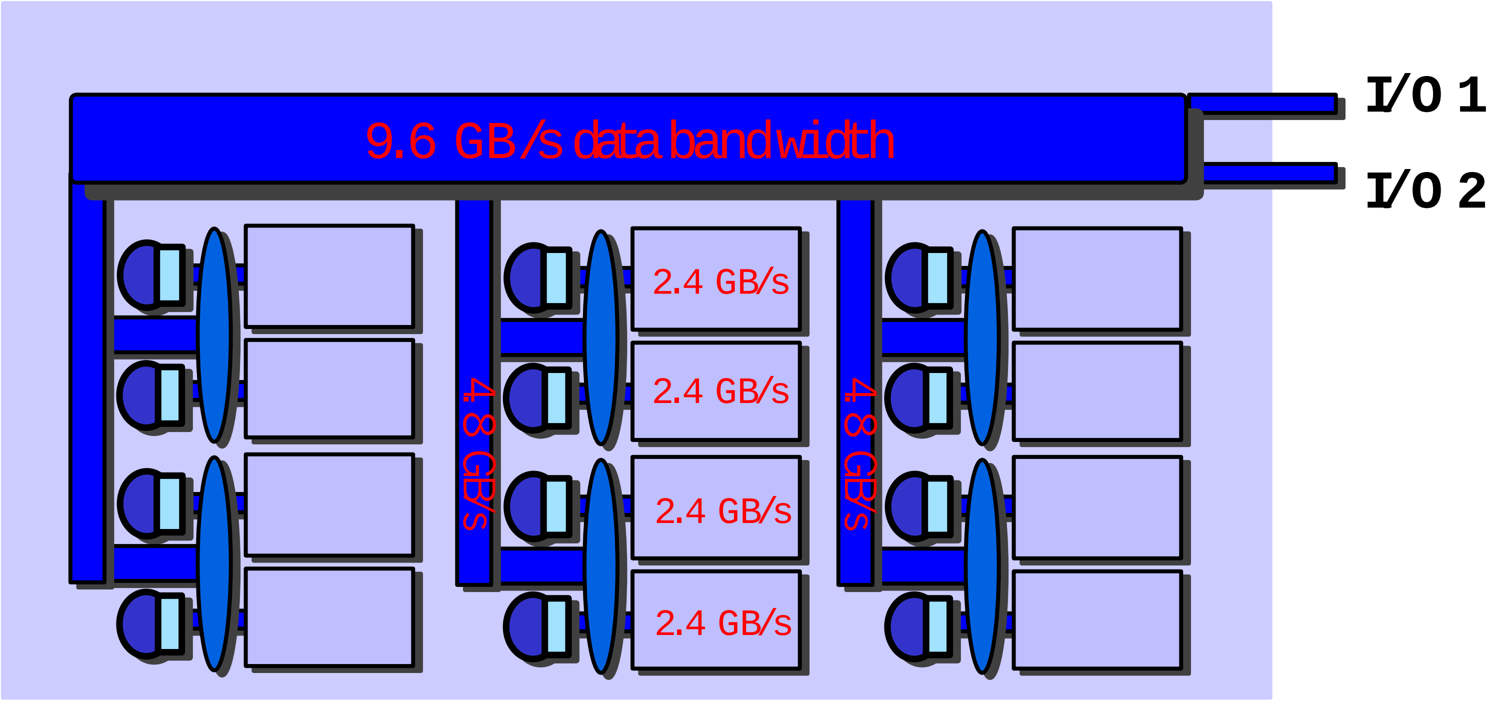
Sun Fire 3800 Server – Basic Design



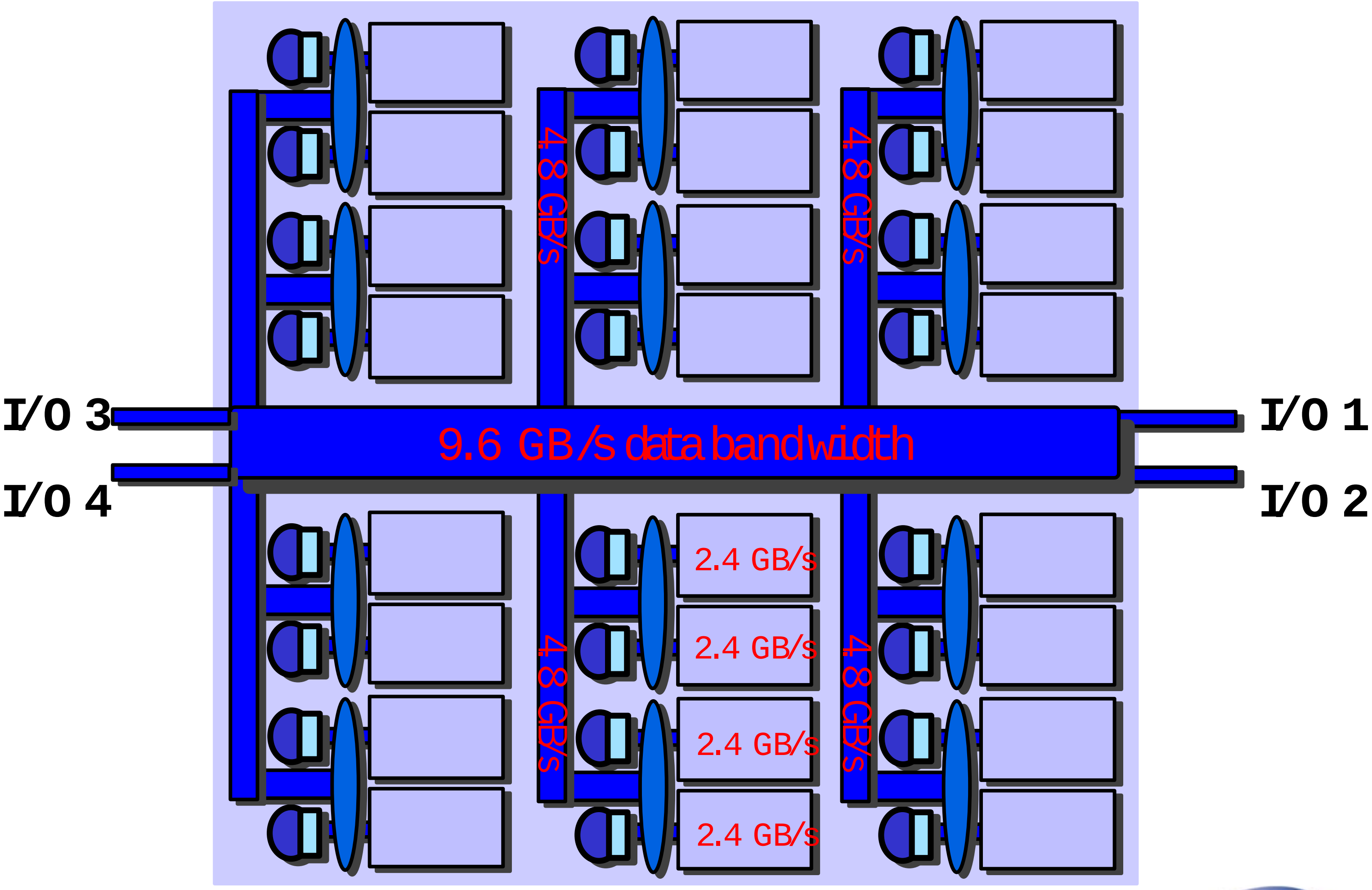
Sun Fire 3800 Server – Basic Design
Now with Address Routers included ... Still 9.6 GB/S



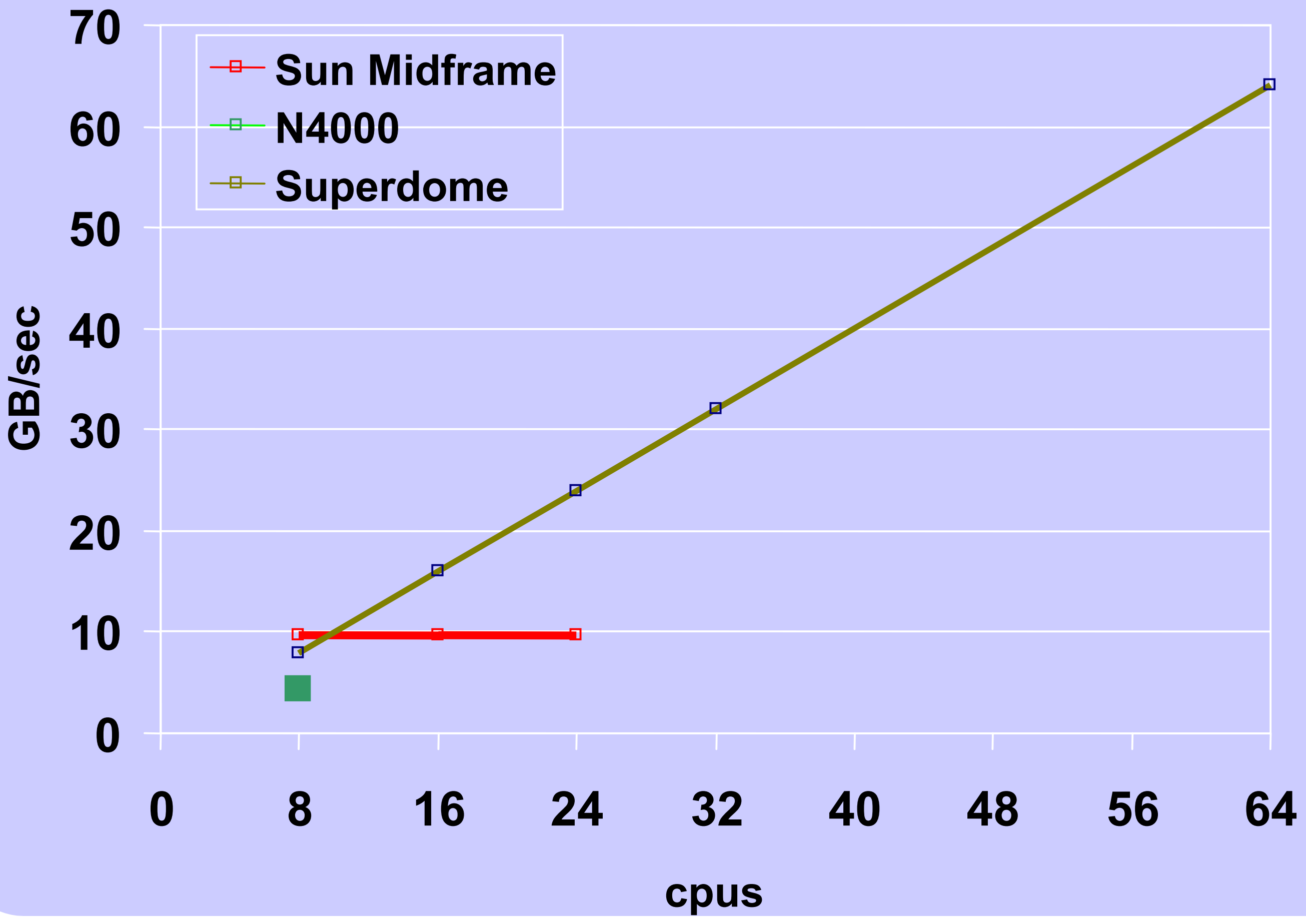
Sun Fire 4800 Server – Basic Design



Sun Fire 6800 Server – Basic Design



System BW Scalability (random access)



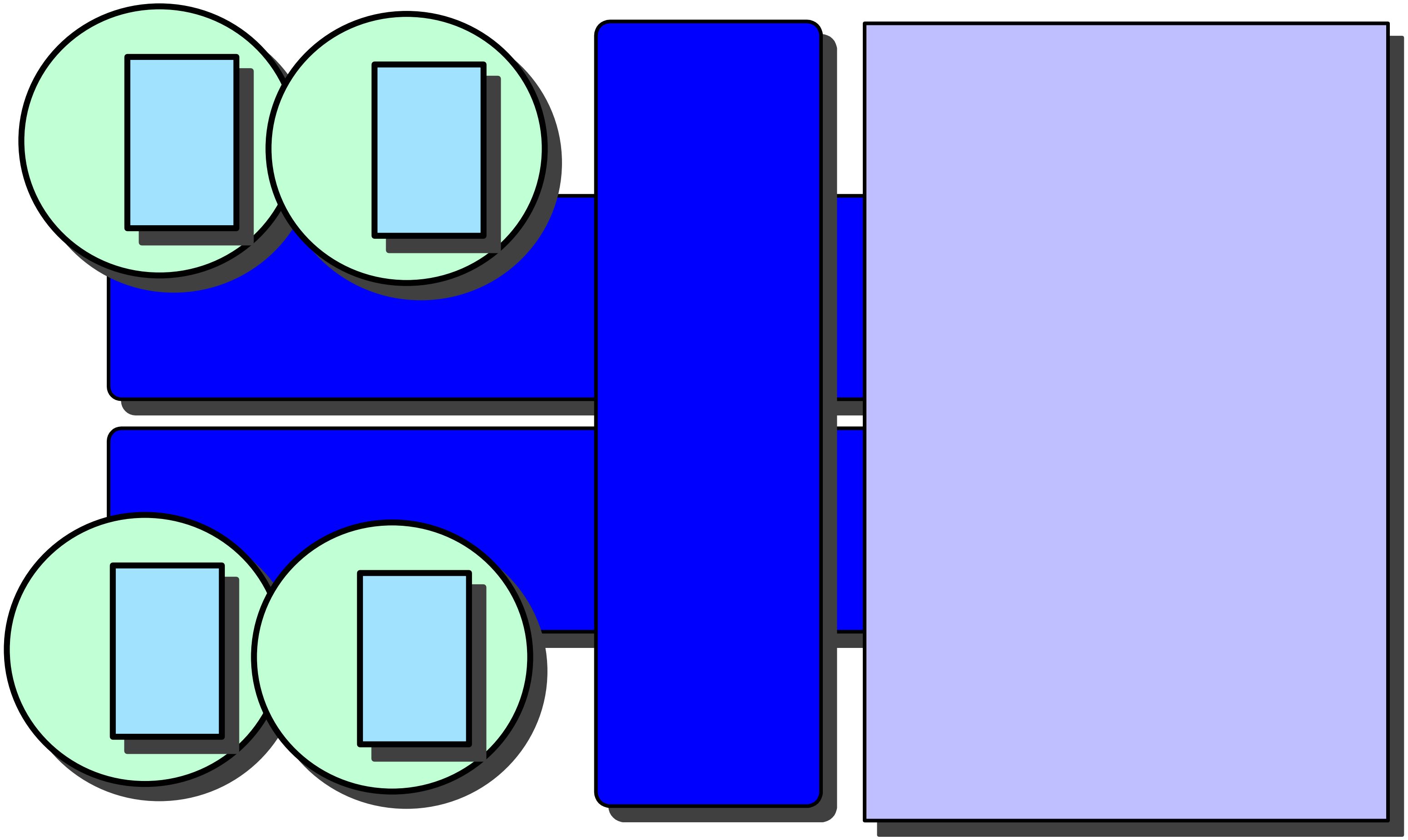
trend 5:

multi-OS environments

Example:

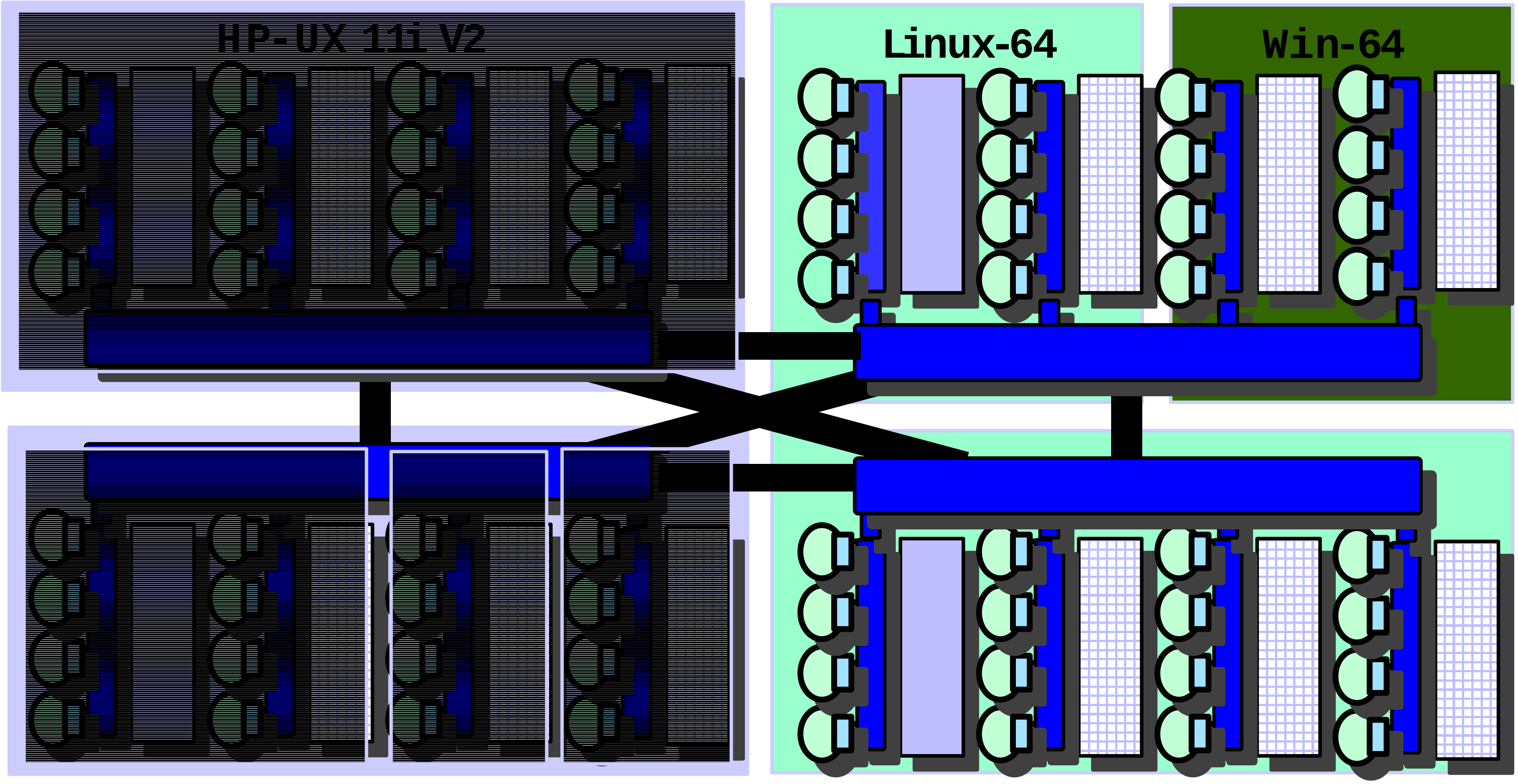
SuperDome after an upgrade to Itanium

Cell2 - Dramatic Improvements

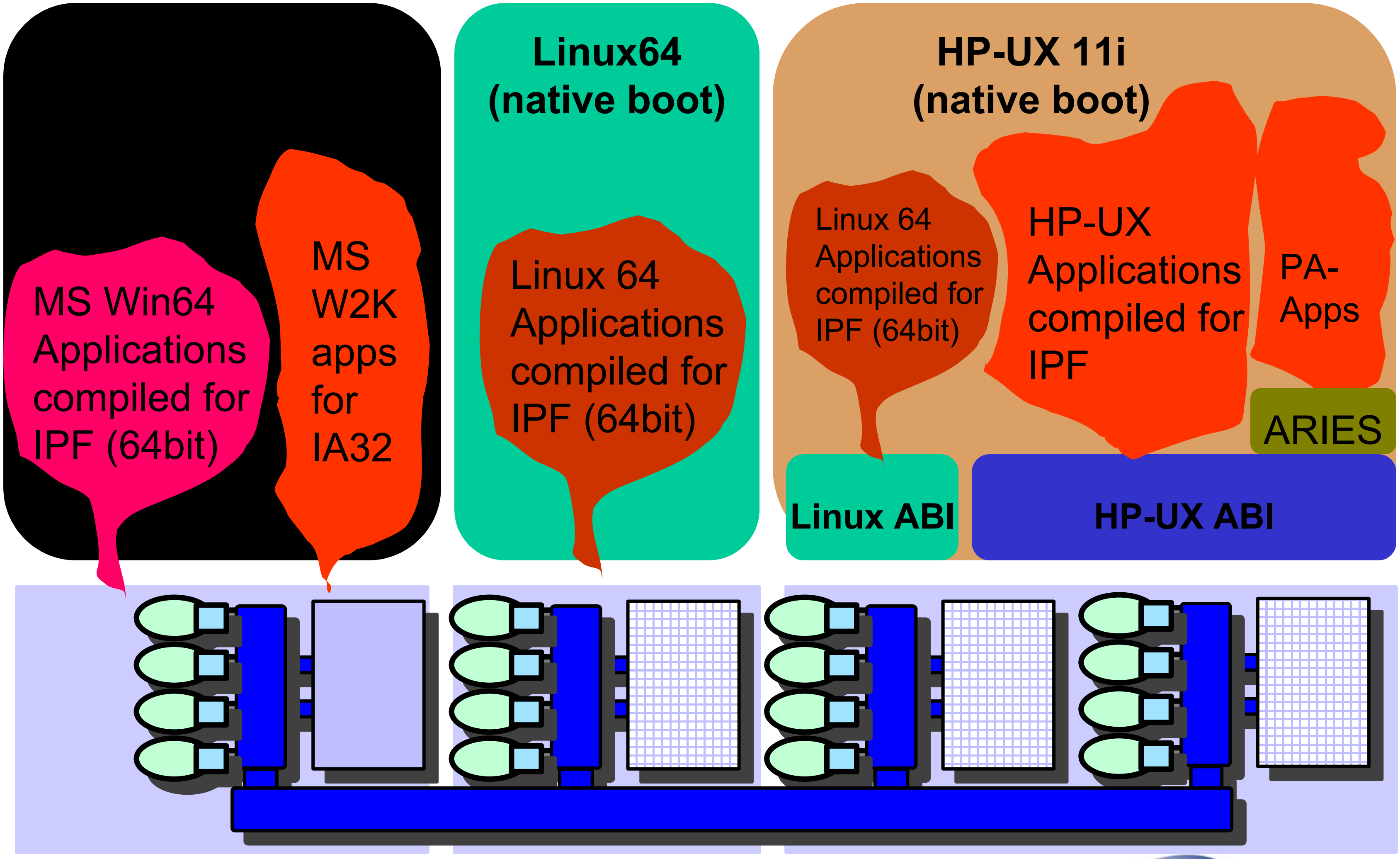


SuperDome-IPF (64 CPUs)

4 nPartitions (1 x 32, 1 x 16, 2 x 8 CPUs), 1 and 3 vPars for HP-UX

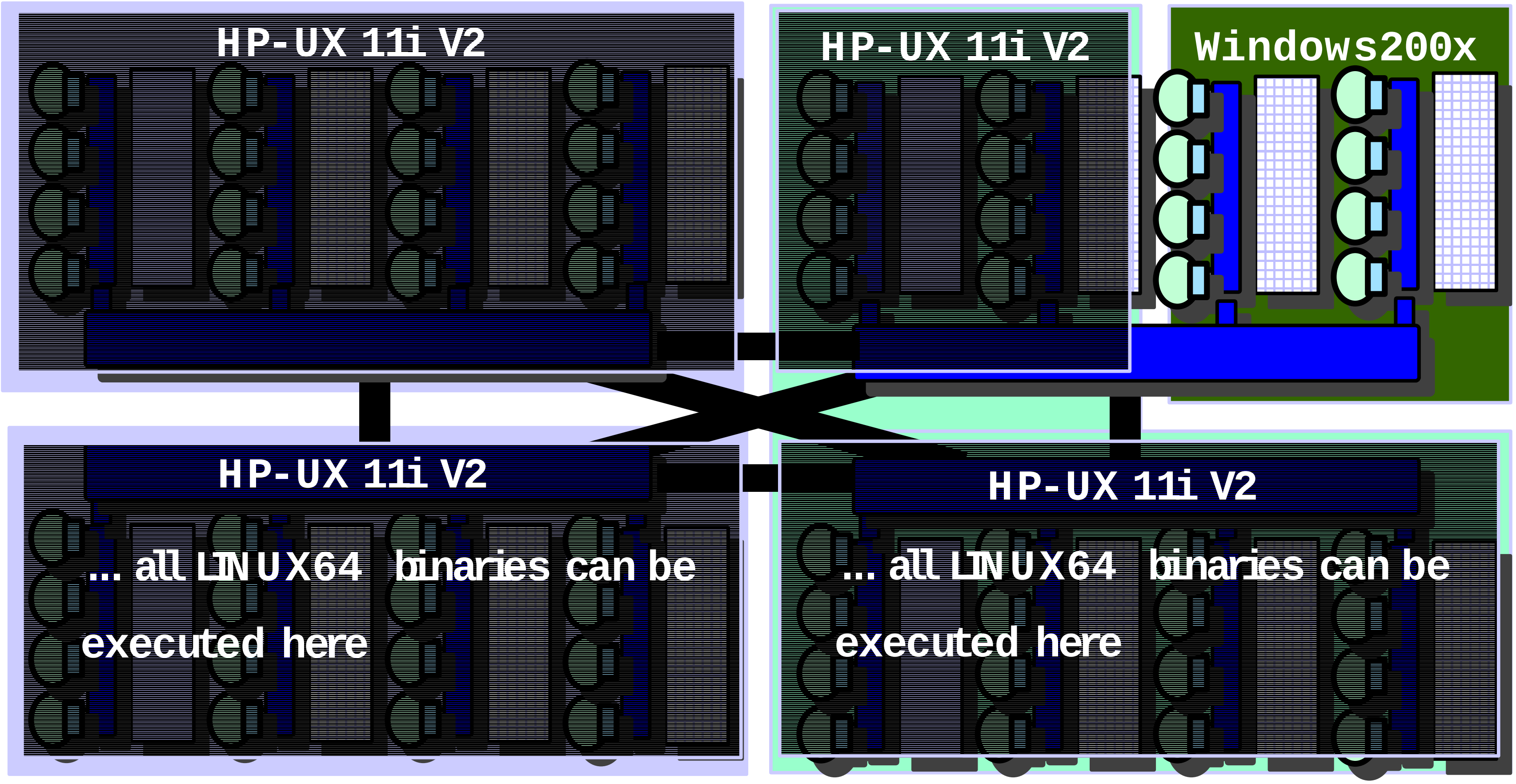


OS/ABI flexibility on HP IPF-Systems


MS Win64 Applications compiled for IPF (64bit)
MS W2K apps for IA32
Linux64 (native boot)
Linux 64 Applications compiled for IPF (64bit)
HP-UX 11i (native boot)
Linux 64 Applications compiled for IPF (64bit)
HP-UX Applications compiled for IPF
PA-Apps
ARIES
Linux ABI
HP-UX ABI

SuperDome 64 (IPF) CPUs

4 nPartitions (1 x 32, 1 x 16, 2 x 8 CPUs)


HP-UX 11i V2
HP-UX 11i V2
Windows200x
HP-UX 11i V2
... all LINUX64 binaries can be executed here
HP-UX 11i V2
... all LINUX64 binaries can be executed here

Thank You

invent

HP WORLD 2002
Conference & Expo