NEC HPC platforms

Introduction and motivation

H. Berger   NEC/ESS
EHPCTC Stuttgart

SX-5
AzusA
NEC   25/11/03

Why are you here?

- Because your simulation requires
  - an extraordinary amount of memory
  - an extraordinary amount of CPU time
  - an extraordinary amount of disk space or I/O performance
- Because you wan't to learn to write parallel code
- Because you wan't to learn to get the maximum out of your code?

SX-5
AzusA
NEC   25/11/03

Where does performance come from?

- 50%: fast systems (from NEC...)
- other 50%: fast code (from you...)
- key performance enablers are
  - ◆ parallelism → pipelining
  - ◆ parallelism → superscalar design
  - ◆ parallelism → multiple CPUs
  - ◆ bandwidth
  - ◆ clock rate

SX-5
AzusA

NEC 25/11/03

How to get performance

- writing fast code is writing parallel code
- writing parallel code does not start with MPI or OpenMP
- single thread performance should be improved first
- your goal is not scalability, but time to solution!
- learn to exploit lower levels of parallelism
- make it visible - the compiler will make the rest

SX-5
AzusA

NEC 25/11/03

Understand and benefit

- By knowing where performance comes from, you can learn where performance disappears
- try to understand your hardwares architecture
- what can you expect?
- What do you get?
- Why is it not the same?
- Next step: improve your algorithms

Clock rate

- Simple: the higher, the better
- but: can memory keep up?
- What to do with several billion operations per second on only 100 million operands?
- Solution: caches
- fast, expensive and small memory
- you are lucky if your data fits in
- otherwise, you are lost. Life is that simple.

Bandwidth

- As mentioned: can not keep up with clockrate growth
- bandwidth and latency are closely related
- latency is even worse, as it is not decreasing
- bandwitdh is determined by
  - ◆ bus clock speed   ↑
  - ◆ bus width   ↑
  - ◆ latency   ↓
  - ◆ number of outstanding transactions   ↑

SX-5
AzusA

NEC   25/11/03

Pipelining

- Very simple and well know approach to speed up tasks consisting of subtasks
- example: automotive industry
  - ◆ move the car
  - ◆ every pipeline stage makes the car more complete
  - ◆ every stage is specialized for one task

time

SX-5
AzusA

NEC   25/11/03

Pipelining 2

- Works only if operations are independant
- it takes the same time to get the result
- but: more results can be computed in the same time
- used for long time in every days work
- used in computers for >30 years
- is used in PCs for ~ 10 years
- NEC SX vector computer: pipelined everything
  - ◆ computation
  - ◆ memory access ➔ latency hiding!

SX-5
AzusA

NEC   25/11/03

Superscalar design

- Simply add arithmetic units
- for example: two multiply-add unions instead of one
- to keep it running: several *independant* operations have to be available
- available for ~ 10 years in PCs
- in SX series: 8 or 16 parallel sets of pipelines
- in Azusa: 2 multiply-add units

SX-5
AzusA

NEC   25/11/03

Several cpus

- Two ways: ,,shared memory" or ,,distributed memory"
- shared memory offers high comfort
- incremental parallelization is possible
- drawback: higher costs
- solutions:
  - ◆ distributed shared memory
  - ◆ non uniform access shared memory

SX-5
AzusA

NEC   25/11/03

Small vs. Big

- Last 10 years: trend away from single strong CPU towards many weak CPU
- Promise: cheaper and as fast as vector
- Problem: Amdahls law
- Just adding hardware does not solve the problem
- Software has to improve as well
- Can software improve enough?
- Can YOU improve your software enough?

SX-5
AzusA

NEC   25/11/03

Why strong single CPU?

- Amdahls law

	98	99	99.90
8	7.02	7.48	7.94
16	12.31	13.91	15.76
512	45.63	83.80	338.85
1024	47.72	91.18	506.18

- Might be a good idea to operate in the „nice"
  area of amdahls law...

SX-5
AzusA

NEC

Distributed shared memory

- Example: NEC SX series
- nodes with up to 16 CPUs with up to 128 GB of
  shared memory
- can be coupled to a cluster using IXS crossbar
- programming model:
  - ◆ Thread parallelism inside node
  - ◆ message passing between nodes over IXS link

SX-5
AzusA

NEC

Non uniform shared memory

- Example: NEC AzusA
- up to 16 CPUs on 64 GB shared memory
- system consists of 4 cells with 4 CPUs each
- cells are connected by crossbar
- cache coherency is done by hardware
- remote latency is very low
- feels like uniform shared memory
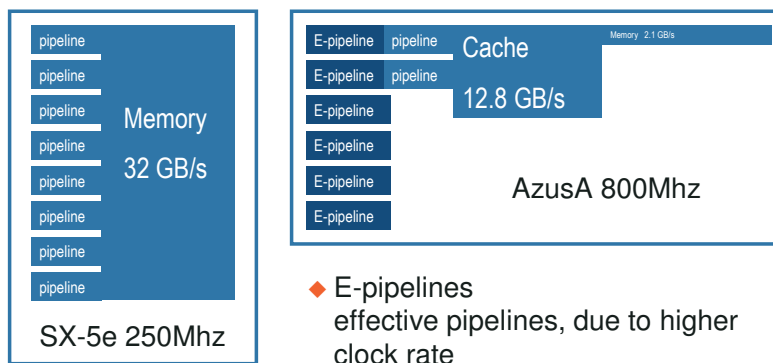
SX-5
AzusA

NEC
25/11/03

Scalar vs vector

- Modern RISC CPU
  - pipelining
  - superscalar
  - software pipelining
  - high bandwidth caches

- Modern vector CPU
  - vector pipelines
  - several pipe sets
  - chaining
  - vector data registers
  - high bandwidth memory
  - pipelined memory access

- RISC learned a lot from vector computers
- But they suffer from bandwith due to non pipelined memory access

SX-5
AzusA

NEC
25/11/03

Scalar vs vector

pipeline	
pipeline	
pipeline	**Memory**
pipeline	
pipeline	**32 GB/s**
pipeline	
pipeline	
pipeline	

SX-5e 250Mhz

E-pipeline	pipeline	Cache	Memory 2.1 GB/s
E-pipeline	pipeline	12.8 GB/s	
E-pipeline			
E-pipeline			
E-pipeline		AzusA 800Mhz	
E-pipeline			

◆ E-pipelines
effective pipelines, due to higher clock rate

NEC   25/11/03   *SX-5 AzusA*

Why NEC?

- All key technologies for HPC inside NEC
  - ◆ Semiconductor Devices
  - ◆ Packaging
  - ◆ HW Design
  - ◆ Interconnections and Network
  - ◆ Operating Systems Software
  - ◆ Languages and Tools
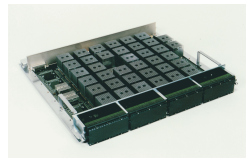 - ◆ Applications Tuning and Support
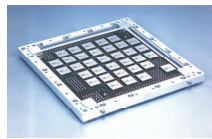
NEC   25/11/03   *SX-5 AzusA*

NEC products for HPC

Products Enhancements Released
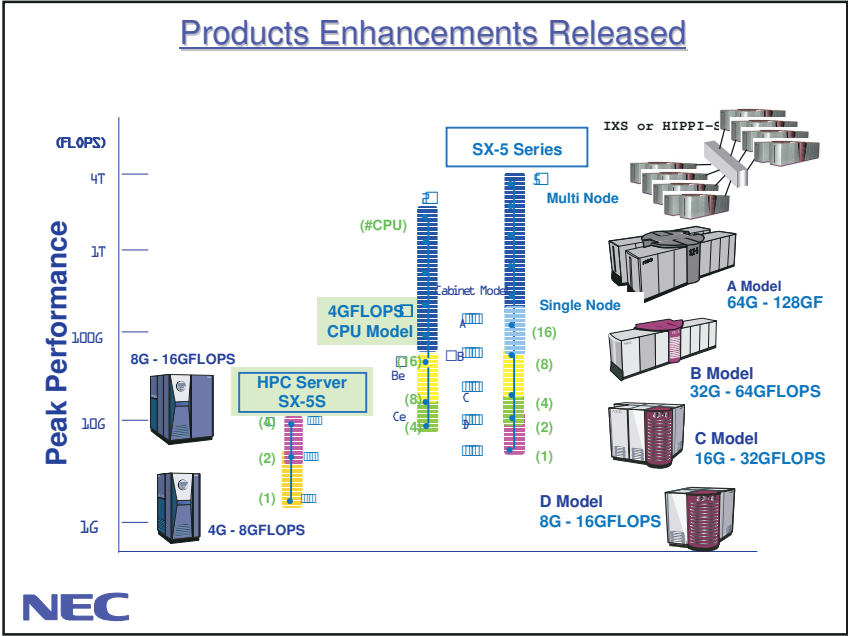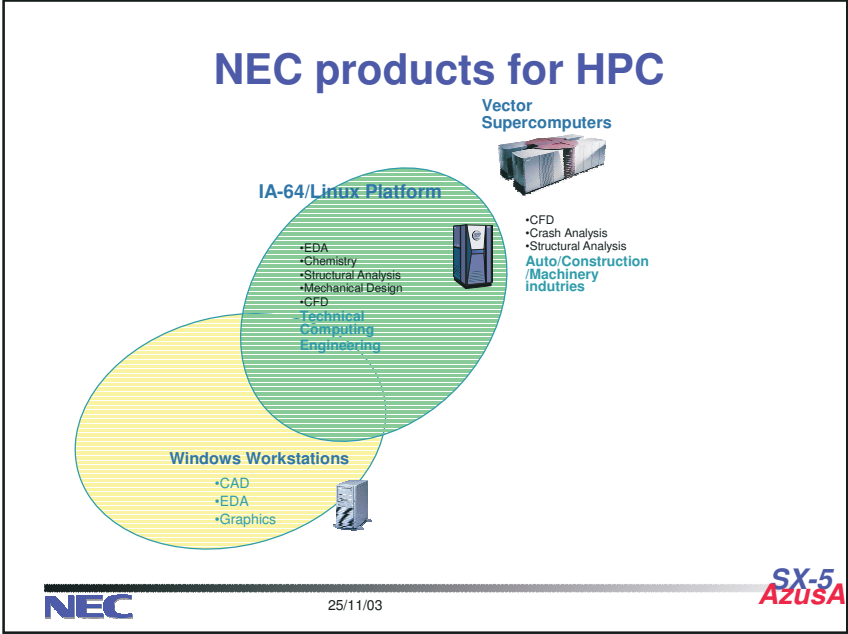
Processor Card

457 x 386mm	225 x 225 mm	20 x 20mm
SX - 4	SX- 5	SX-5X

Performance	2 GFlops 8 nS	8 GFlops 4 nS	8GFlops 2 nS
LSI	0.35μ m CMOS	0.25μ mCMOS	0.15μ m CMOS
	37 Chips	32 Chips	1 Chip Processor

SX-5
AzusA
NEC   25/11/03

Memory Card
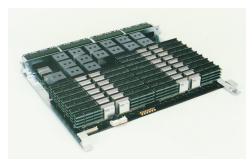
457 x 386mm	457 x 386	105 x 120mm
SX - 4	SX- 5	SX-5X
Capacity : 256MB / Card	4GB / Card	1GB / Card
Memory Chip : 4Mb SSRAM	64Mb SDRAM	256Mb SDRAM
32Mb SDRAM		
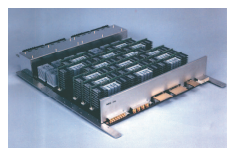
SX-5
AzusA
NEC   25/11/03
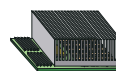
Questions?

- Support desk at HWW:

hwwsupport@ess.nec.de