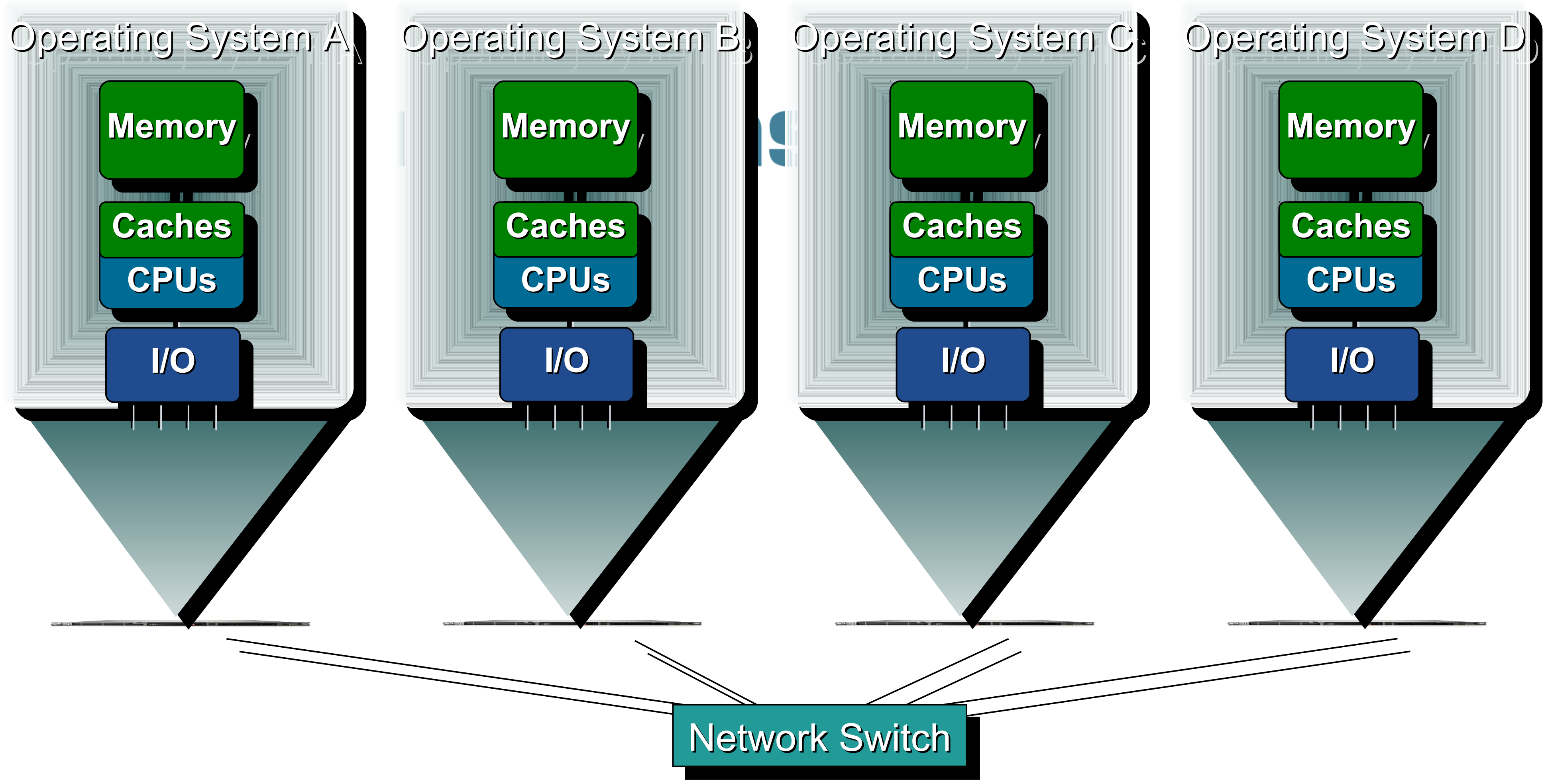
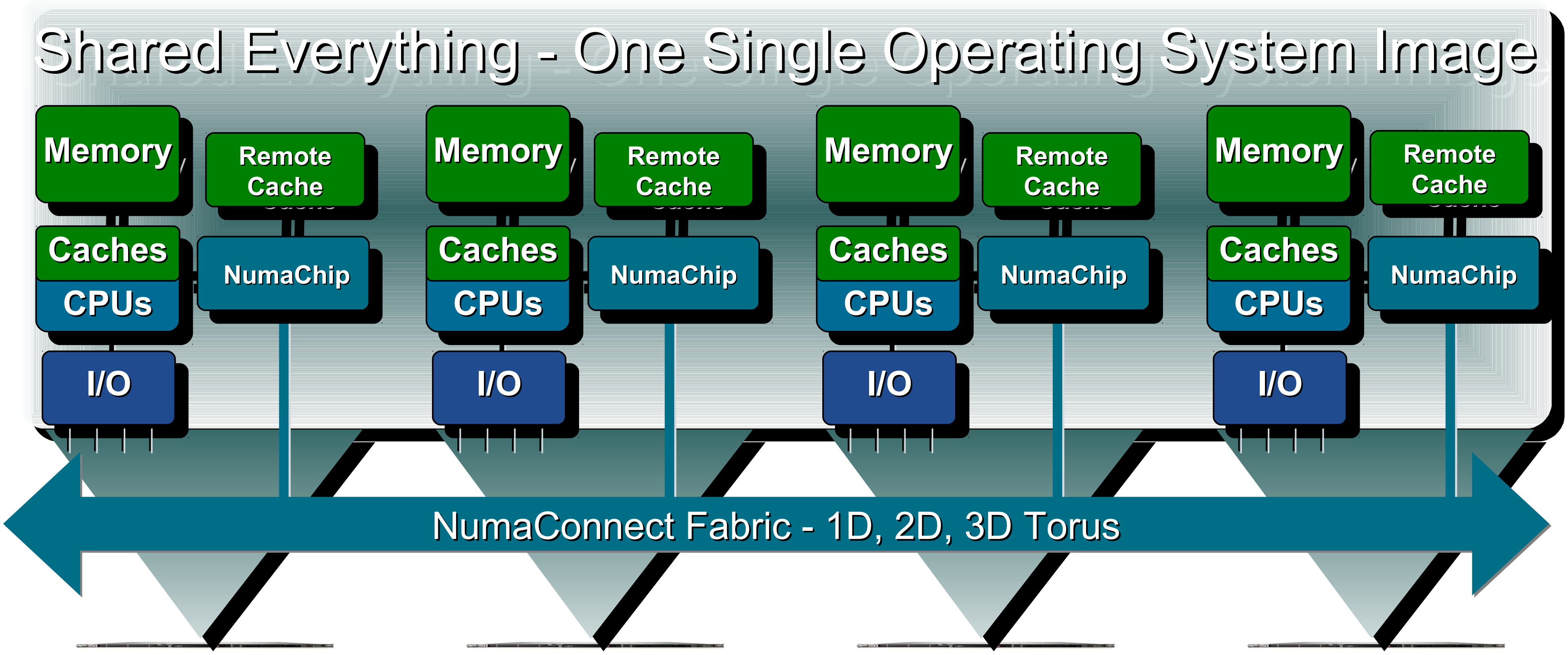
Clusters - NO Shared Resources

Individual Instances of the Operating System



Cache Coherent Shared Memory

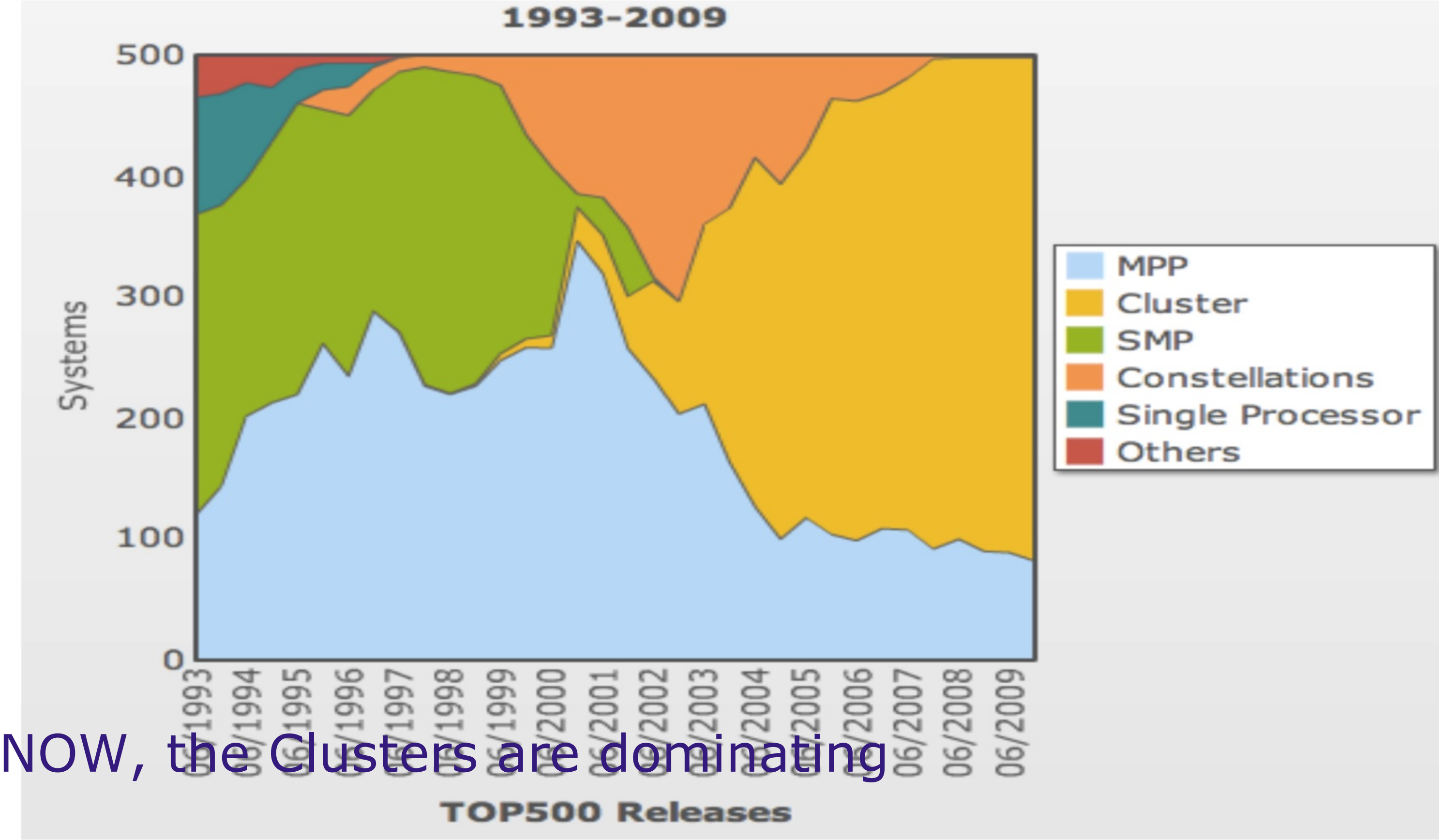


Capabilities like Mainframe - Price like Cluster

Top500 History

numascale

- The expensive SMPs used to rule:
  - Cray XMP, Convex Exemplar, Sun ES



- NOW, the Clusters are dominating

Technology Background

- Convex Exemplar (Acquired by HP)
  - First implementation of the ccNUMA architecture from Dolphin in 1994
- Data General Aviion (Acquired by EMC)
  - Designed in 1996 with deliveries from 1997 - 2002
  - Used Dolphin's chips with 3 generations of processor/memory buses
- I/O Attached Products for Clustering OEMs
  - Sun Microsystems (SunCluster)
  - Siemens RM600 Server (IO Expansion)
  - Siemens Medical (3D CT)
  - Philips Medical (3D Ultra Sound)
  - Dassault/Thales Rafale
- HPC Clusters (WulfKit w. Scali)
  - First Low Latency Cluster Interconnect

Dolphin's Cache Chip

Dolphin's Low Latency Clustering HW

Company Background

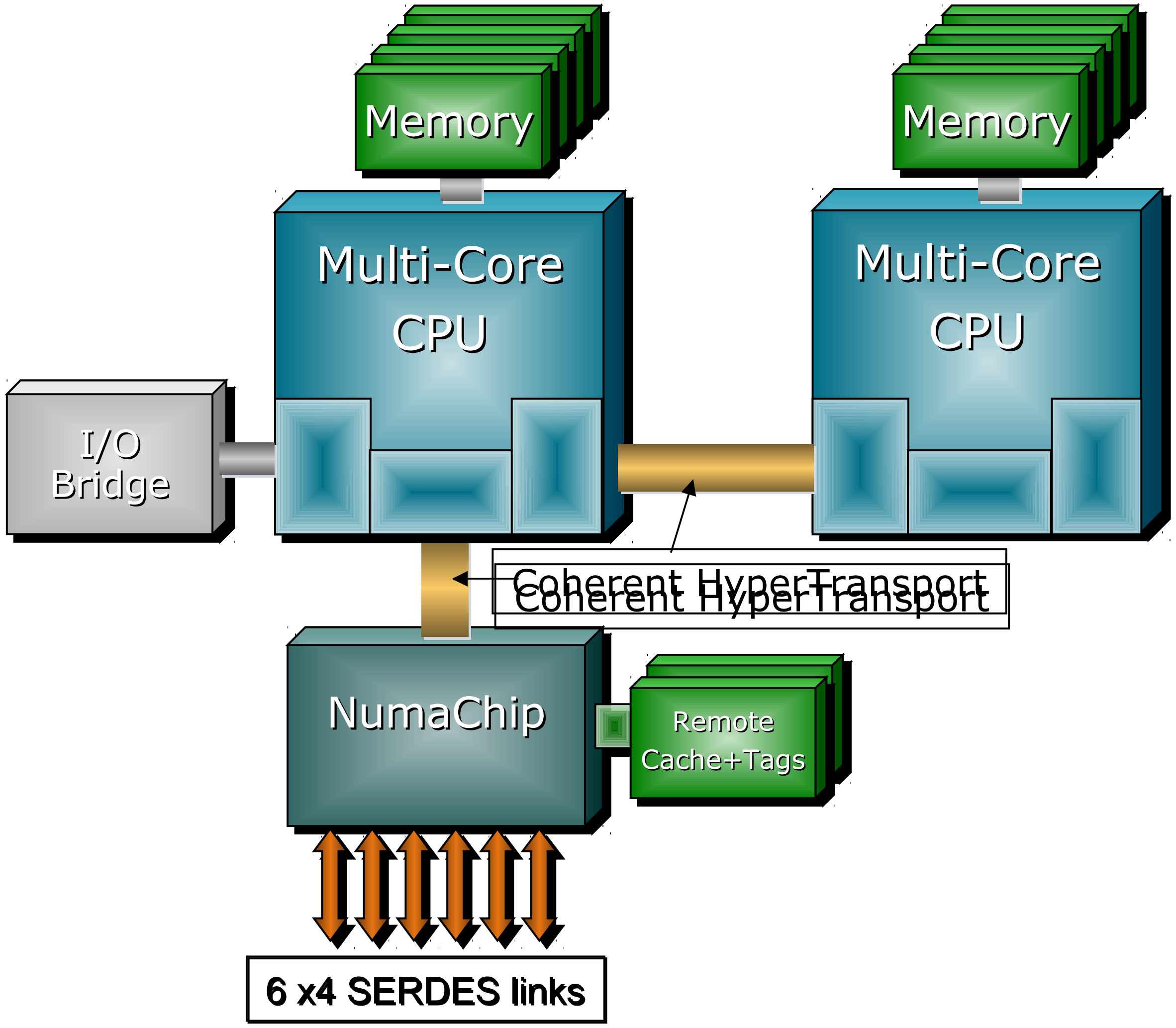
numascale

- Founded in Oslo in 2008 as a spin-out from Dolphin Interconnect Solutions
- Technology from Norsk Data 1987→
  - Dolphin Interconnect 1992
- 24 Experienced Staff Members
  - Interconnects
  - Processor Architecture
  - Supercomputing
  - Data Acquisition
- Main Owners:
  - Investinor (32.1%)
  - ProVenture Seed (20.8%)
  - Statoil ASA (17.5%)
  - Helge B. Risnes (Ex. InfoCare) (7.3%)
  - Svein A. Tunheim (Ex. ChipCon) (7.3%)

NumaConnect Main Features

- 12 bits Node ID = 4,096 nodes, (lots and lost of cores), >19,000 cores
- 48 bits node physical address space = 256TBytes
- Scalable, directory based cache coherency protocol
- Scalable On-Chip switch fabric (2-D, 3-D Torus)
- Configurable Cache for remote data (1 - 16GB/node)
- System-wide cache coherency in hardware
- 64Byte cache line granularity same as x86 CPUs

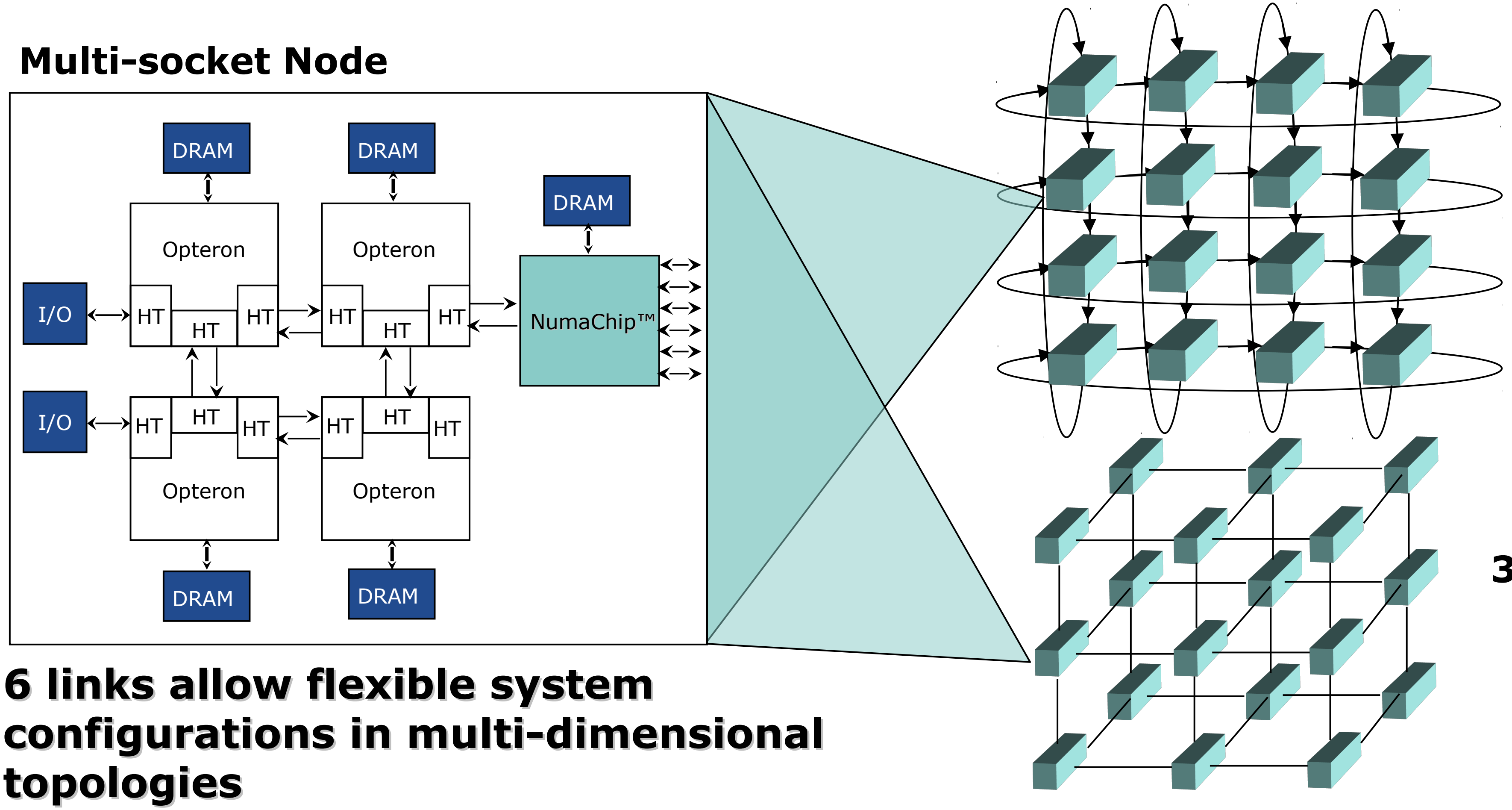
NumaChip™ Server Configuration



NumaChip™ System Architecture

numascale

Multi-socket Node

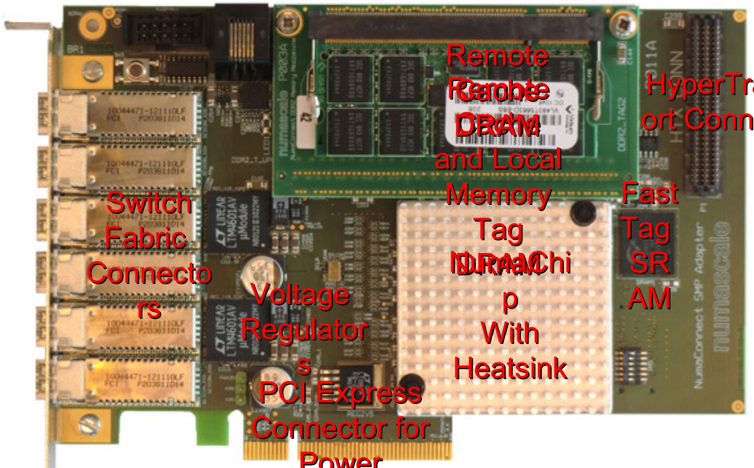


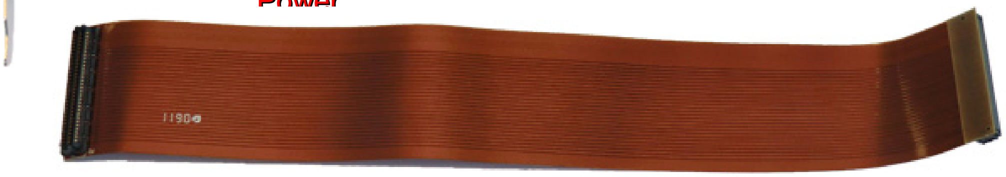
2-D Torus

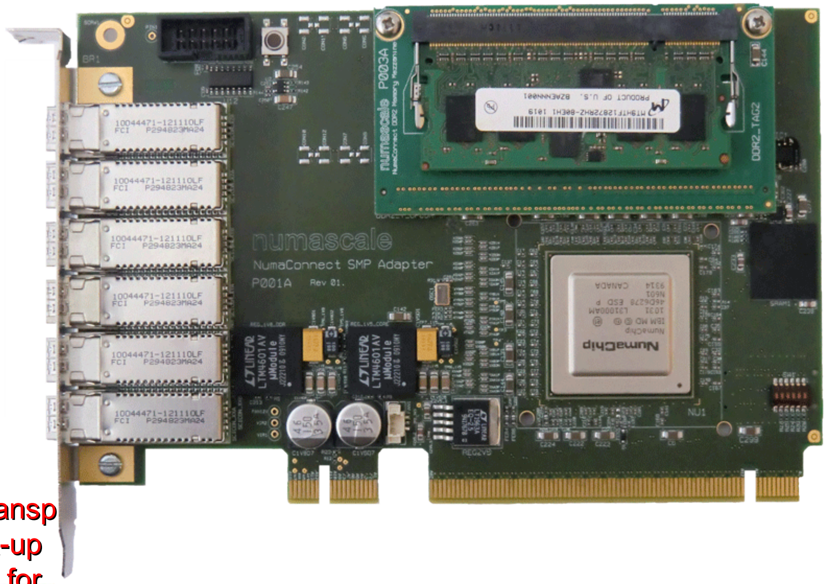
3-D Torus

6 links allow flexible system configurations in multi-dimensional topologies

NumaConnect Cards HTX & PCIe





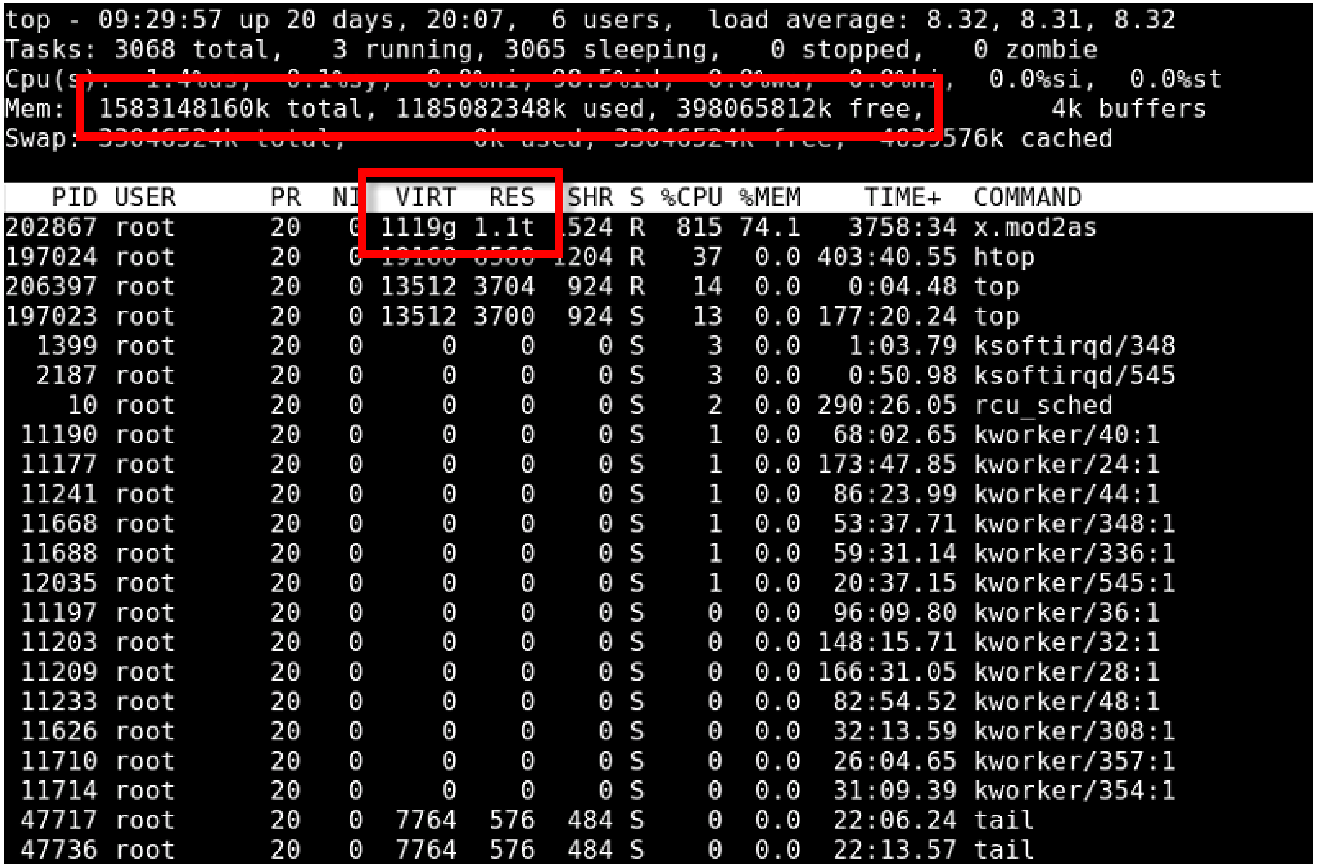


Shorter Time to Performance

A NumaConnect system can be programmed just as an "ordinary" computer!

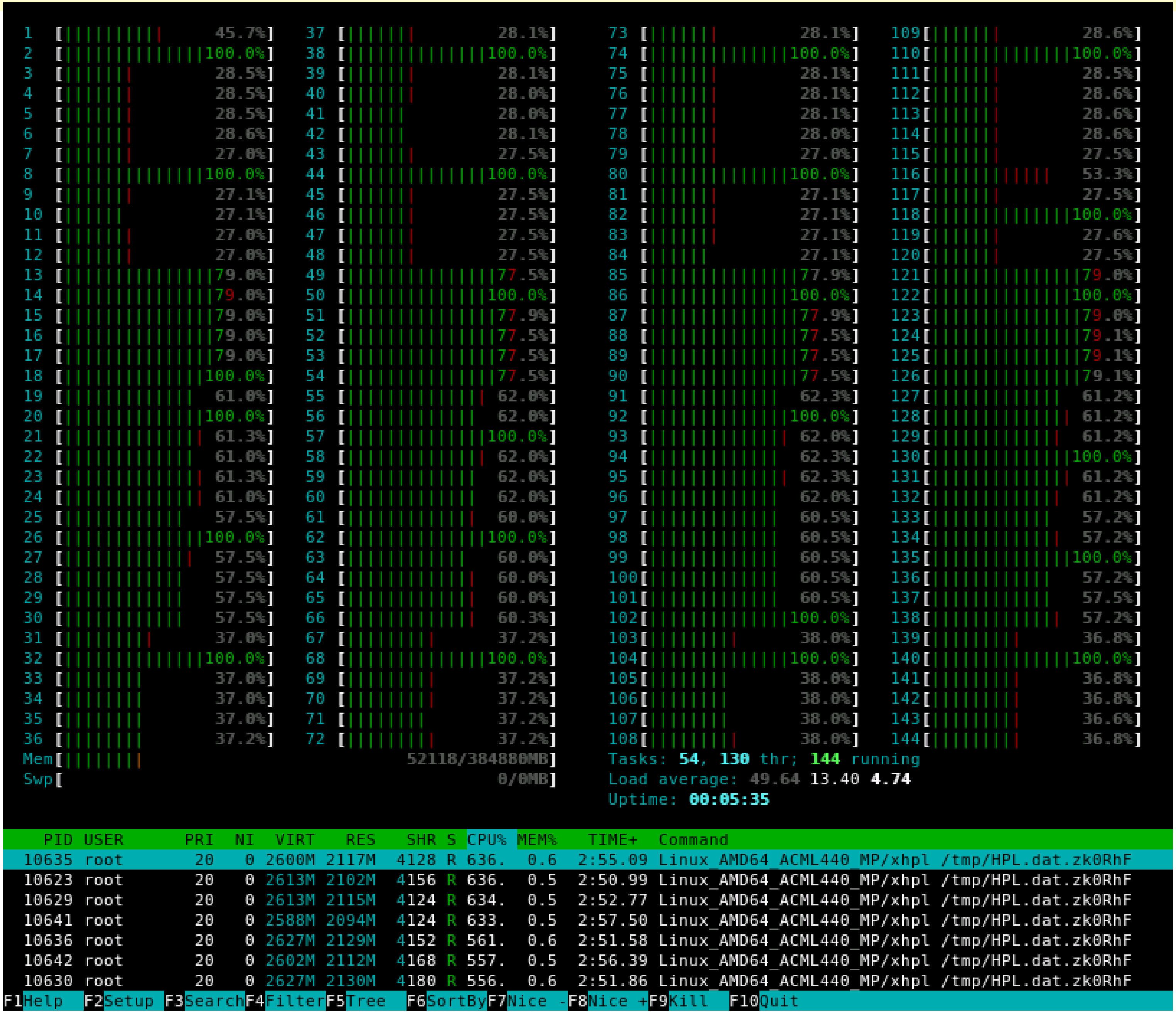
The full memory range is available to all applications

You can run "top" on a 1.5TB NumaConnect system



```
top - 09:29:57 up 20 days, 20:07,  6 users,  load average: 8.32, 8.31, 8.32
Tasks: 3068 total,   3 running, 3065 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.4%us,  0.1%sy,  0.0%ni, 98.5%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  1583148160k total, 1185082348k used, 398065812k free,        4k buffers
Swap: 33046524k total,        0k used, 33046524k free,  4035576k cached

   PID USER       PR  NI  VIRT  RES  SHR S %CPU %MEM     TIME+  COMMAND
202867 root       20   0 1119g 1.1t 1524 R  815 74.1   3758:34 x.mod2as
197024 root       20   0 19160 6560 1204 R   37  0.0 403:40.55 htop
206397 root       20   0 13512 3704  924 R   14  0.0   0:04.48 top
197023 root       20   0 13512 3700  924 S   13  0.0 177:20.24 top
  1399 root       20   0     0    0    0 S    3  0.0   1:03.79 ksoftirqd/348
  2187 root       20   0     0    0    0 S    3  0.0   0:50.98 ksoftirqd/545
    10 root       20   0     0    0    0 S    2  0.0 290:26.05 rcu_sched
 11190 root       20   0     0    0    0 S    1  0.0  68:02.65 kworker/40:1
 11177 root       20   0     0    0    0 S    1  0.0 173:47.85 kworker/24:1
 11241 root       20   0     0    0    0 S    1  0.0  86:23.99 kworker/44:1
 11668 root       20   0     0    0    0 S    1  0.0  53:37.71 kworker/348:1
 11688 root       20   0     0    0    0 S    1  0.0  59:31.14 kworker/336:1
 12035 root       20   0     0    0    0 S    1  0.0  20:37.15 kworker/545:1
 11197 root       20   0     0    0    0 S    0  0.0  96:09.80 kworker/36:1
 11203 root       20   0     0    0    0 S    0  0.0 148:15.71 kworker/32:1
 11209 root       20   0     0    0    0 S    0  0.0 166:31.05 kworker/28:1
 11233 root       20   0     0    0    0 S    0  0.0  82:54.52 kworker/48:1
 11626 root       20   0     0    0    0 S    0  0.0  32:13.59 kworker/308:1
 11710 root       20   0     0    0    0 S    0  0.0  26:04.65 kworker/357:1
 11714 root       20   0     0    0    0 S    0  0.0  31:09.39 kworker/354:1
 47717 root       20   0  7764  576  484 S    0  0.0  22:06.24 tail
 47736 root       20   0  7764  576  484 S    0  0.0  22:13.57 tail
```

Large Shared Memory, all threads available



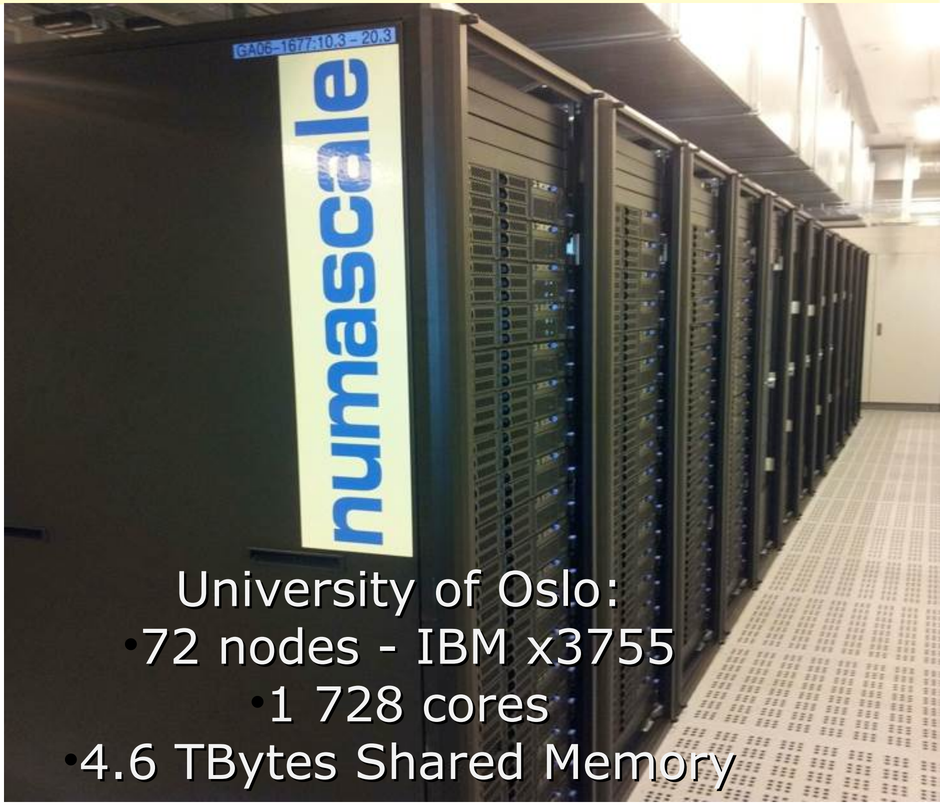
```
1  [||||||||||     45.7%]   37 [|||||||        28.1%]   73 [|||||||        28.1%]   109[|||||||        28.6%]
2  [||||||||||||||100.0%]   38 [||||||||||||||100.0%]   74 [||||||||||||||100.0%]   110[||||||||||||||100.0%]
3  [|||||||        28.5%]   39 [|||||||        28.1%]   75 [|||||||        28.1%]   111[|||||||        28.5%]
4  [|||||||        28.5%]   40 [|||||||        28.0%]   76 [|||||||        28.1%]   112[|||||||        28.6%]
5  [|||||||        28.5%]   41 [||||||         28.0%]   77 [|||||||        28.1%]   113[|||||||        28.6%]
6  [|||||||        28.6%]   42 [||||||         28.1%]   78 [|||||||        28.0%]   114[|||||||        28.6%]
7  [|||||||        27.0%]   43 [|||||||        27.5%]   79 [|||||||        27.0%]   115[|||||||        27.5%]
8  [||||||||||||||100.0%]   44 [||||||||||||||100.0%]   80 [||||||||||||||100.0%]   116[||||||||||||   53.3%]
9  [|||||||        27.1%]   45 [|||||||        27.5%]   81 [|||||||        27.1%]   117[|||||||        27.5%]
10 [||||||         27.1%]   46 [|||||||        27.5%]   82 [|||||||        27.1%]   118[||||||||||||||100.0%]
11 [|||||||        27.0%]   47 [|||||||        27.5%]   83 [|||||||        27.1%]   119[|||||||        27.6%]
12 [|||||||        27.0%]   48 [|||||||        27.5%]   84 [||||||         27.1%]   120[|||||||        27.5%]
13 [|||||||||||||||79.0%]   49 [|||||||||||||||77.5%]   85 [|||||||||||||||77.9%]   121[|||||||||||||||79.0%]
14 [|||||||||||||||79.0%]   50 [||||||||||||||100.0%]   86 [||||||||||||||100.0%]   122[||||||||||||||100.0%]
15 [|||||||||||||||79.0%]   51 [|||||||||||||||77.9%]   87 [|||||||||||||||77.9%]   123[|||||||||||||||79.0%]
16 [|||||||||||||||79.0%]   52 [|||||||||||||||77.5%]   88 [|||||||||||||||77.5%]   124[|||||||||||||||79.1%]
17 [|||||||||||||||79.0%]   53 [|||||||||||||||77.5%]   89 [|||||||||||||||77.5%]   125[|||||||||||||||79.1%]
18 [||||||||||||||100.0%]   54 [|||||||||||||||77.5%]   90 [|||||||||||||||77.5%]   126[|||||||||||||||79.1%]
19 [|||||||||||||  61.0%]   55 [|||||||||||||| 62.0%]   91 [|||||||||||||  62.3%]   127[|||||||||||||  61.2%]
20 [||||||||||||||100.0%]   56 [|||||||||||||  62.0%]   92 [||||||||||||||100.0%]   128[|||||||||||||| 61.2%]
21 [|||||||||||||| 61.3%]   57 [||||||||||||||100.0%]   93 [|||||||||||||| 62.0%]   129[|||||||||||||  61.2%]
22 [|||||||||||||  61.0%]   58 [|||||||||||||| 62.0%]   94 [|||||||||||||  62.3%]   130[||||||||||||||100.0%]
23 [|||||||||||||| 61.3%]   59 [|||||||||||||  62.0%]   95 [|||||||||||||| 62.3%]   131[|||||||||||||| 61.2%]
24 [|||||||||||||| 61.0%]   60 [|||||||||||||  62.0%]   96 [|||||||||||||  62.0%]   132[|||||||||||||  61.2%]
25 [||||||||||||   57.5%]   61 [|||||||||||||  60.0%]   97 [|||||||||||||  60.5%]   133[||||||||||||   57.2%]
26 [||||||||||||||100.0%]   62 [||||||||||||||100.0%]   98 [|||||||||||||  60.5%]   134[|||||||||||||  57.2%]
27 [|||||||||||||  57.5%]   63 [||||||||||||   60.0%]   99 [|||||||||||||  60.5%]   135[||||||||||||||100.0%]
28 [||||||||||||   57.5%]   64 [|||||||||||||  60.0%]   100[|||||||||||||  60.5%]   136[||||||||||||   57.2%]
29 [||||||||||||   57.5%]   65 [|||||||||||||  60.0%]   101[|||||||||||||  60.5%]   137[||||||||||||   57.5%]
30 [||||||||||||   57.5%]   66 [|||||||||||||  60.3%]   102[||||||||||||||100.0%]   138[|||||||||||||  57.2%]
31 [|||||||||      37.0%]   67 [|||||||||      37.2%]   103[|||||||||      38.0%]   139[||||||||       36.8%]
32 [||||||||||||||100.0%]   68 [||||||||||||||100.0%]   104[||||||||||||||100.0%]   140[||||||||||||||100.0%]
33 [||||||||       37.0%]   69 [|||||||||      37.2%]   105[||||||||       38.0%]   141[||||||||       36.8%]
34 [||||||||       37.0%]   70 [|||||||||      37.2%]   106[||||||||       38.0%]   142[||||||||       36.8%]
35 [||||||||       37.0%]   71 [||||||||       37.2%]   107[||||||||       38.0%]   143[||||||||       36.6%]
36 [||||||||       37.2%]   72 [|||||||||      37.2%]   108[|||||||||      38.0%]   144[||||||||       36.8%]
Mem[||||||||                          52118/384880MB]   Tasks: 54, 130 thr; 144 running
Swp[                                           0/0MB]   Load average: 49.64 13.40 4.74
                                                        Uptime: 00:05:35

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
10635 root       20   0 2600M 2117M  4128 R 636.  0.6  2:55.09 Linux_AMD64_ACML440_MP/xhpl /tmp/HPL.dat.zk0RhF
10623 root       20   0 2613M 2102M  4156 R 636.  0.5  2:50.99 Linux_AMD64_ACML440_MP/xhpl /tmp/HPL.dat.zk0RhF
10629 root       20   0 2613M 2115M  4124 R 634.  0.5  2:52.77 Linux_AMD64_ACML440_MP/xhpl /tmp/HPL.dat.zk0RhF
10641 root       20   0 2588M 2094M  4124 R 633.  0.5  2:57.50 Linux_AMD64_ACML440_MP/xhpl /tmp/HPL.dat.zk0RhF
10636 root       20   0 2627M 2129M  4152 R 561.  0.6  2:51.58 Linux_AMD64_ACML440_MP/xhpl /tmp/HPL.dat.zk0RhF
10642 root       20   0 2602M 2112M  4168 R 557.  0.5  2:56.39 Linux_AMD64_ACML440_MP/xhpl /tmp/HPL.dat.zk0RhF
10630 root       20   0 2627M 2130M  4180 R 556.  0.6  2:51.86 Linux_AMD64_ACML440_MP/xhpl /tmp/HPL.dat.zk0RhF
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Linux

- NumaConnect Architecture Supported in Linux kernel
  - Interprocessor Interrupt (APIC extension HW)
- Runs with standard kernel
- Tuned kernel recommended
  - Especially for large systems >8 servers
  - “Custom Kernel” with recommended options
- Patches
  - Queue-Based Spin Locks (Scalability)
  - Optimized Timing Framework for NumaConnect Fabric

Memory Bandwidth-Stream

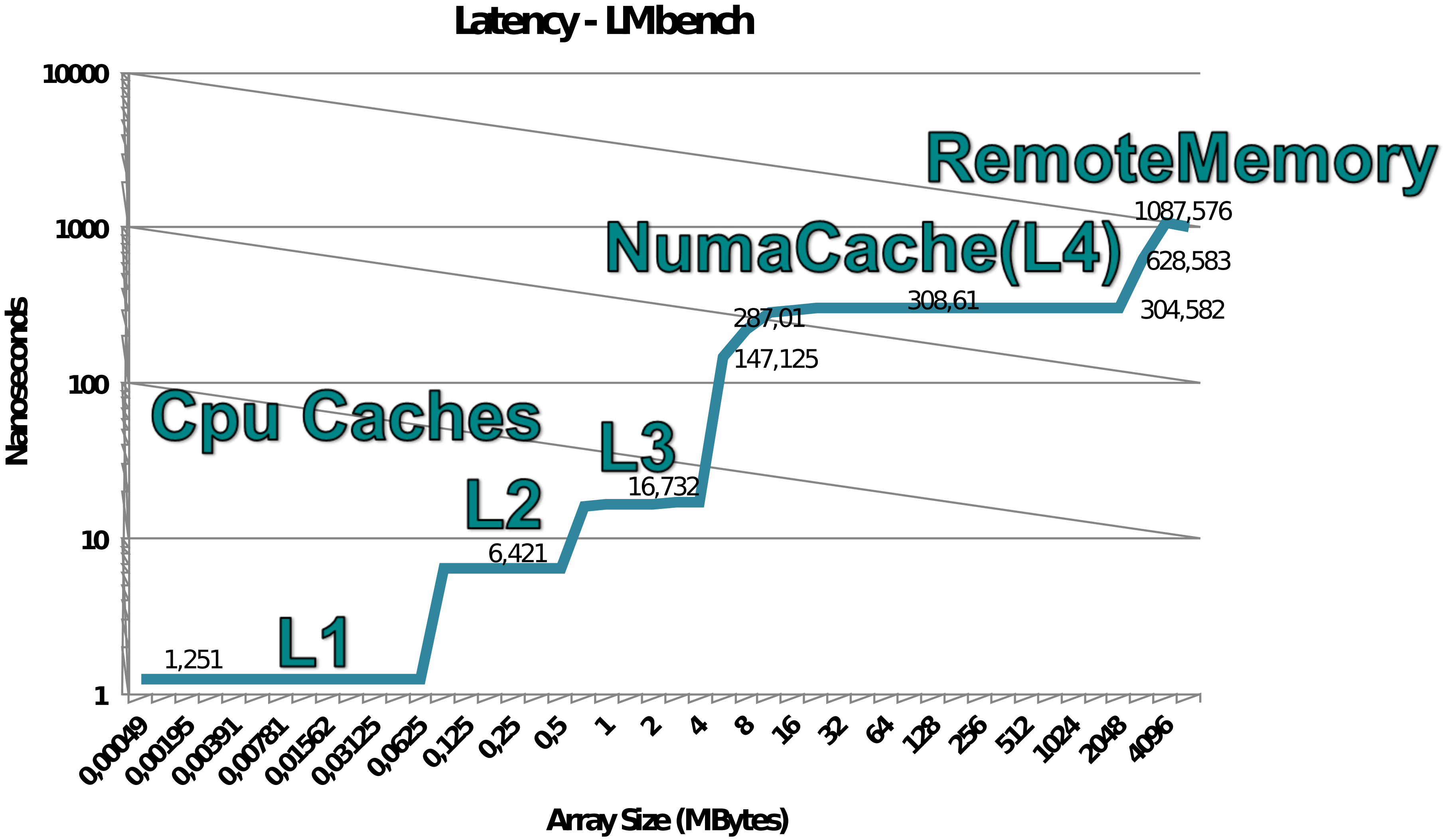
```
This system uses 8 bytes per DOUBLE PRECISION word.

-------------------------------------------------------------

Array size = 180000000000, Offset = 0

Total memory required = 4119873.0 MB.

Each test is run 10 times, but only

the *best* time for each is used.

-------------------------------------------------------------

Number of Threads requested = 828

-------------------------------------------------------------

Function      Rate (MB/s)   Avg time     Min time     Max time

Copy:       1599317.5028       1.9224       1.8008       2.1393

Scale:      1468219.1643       2.0954       1.9616       2.2290

Add:        1664455.1221       2.8375       2.5954       3.0947

Triad:      1492414.0721       3.0478       2.8946       3.3267
```



University of Oslo:
- 72 nodes - IBM x3755
- 1 728 cores
- 4.6 TBytes Shared Memory

Human readable numbers:
Array size = 180 GB
Copy = 1.6 TB/s
Scale =1.5 TB/s
Add = 1.7 TB/s
Triad = 1.5 TB/s

LMbench - Chart



RWTH TrajSearch

Code performance and scaling results

Dipl.-Inform. Dirk Schmidl

- High Performance Computing
- RWTH Aachen
- Center for Computing and Communication

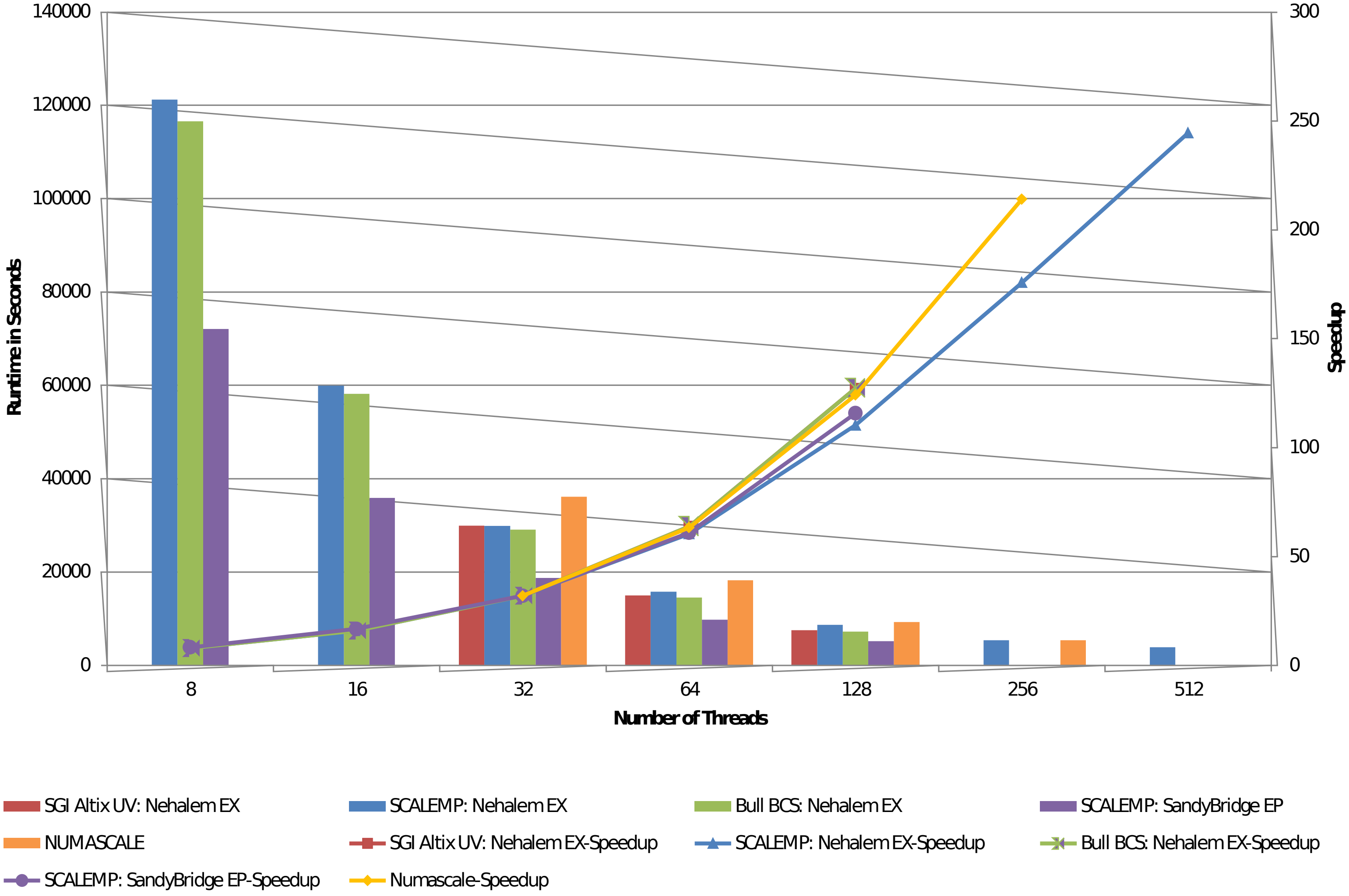
Numascale Demo System

- 8 Supermicro 1042G-LTF+ Servers with NumaConnect
- Each Server
  - 128GB RAM
  - Two 16 cores AMD 6380 processors
  - 512GB SSD per node
- Total 256 cores sharing 1TB

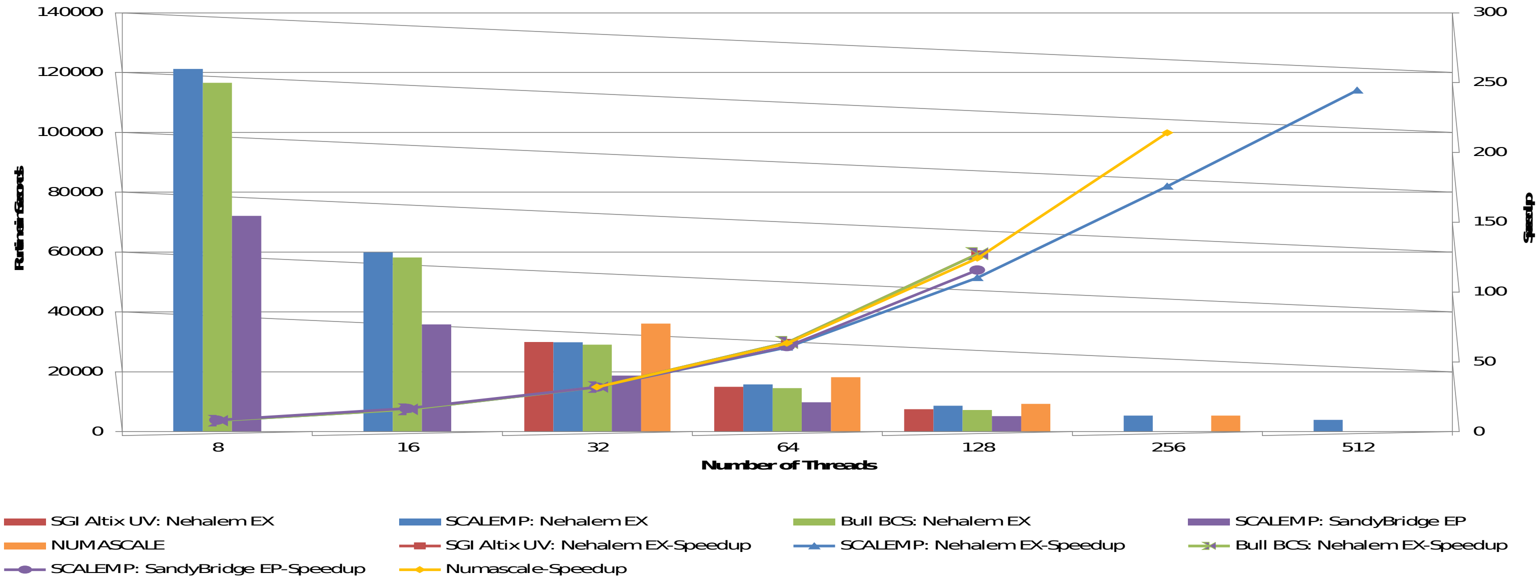
RWTH TrajSearch code

numascale

The graph shows that the application has the **most speedup on NumaConnect**, even if it was originally adapted to ScaleMP.

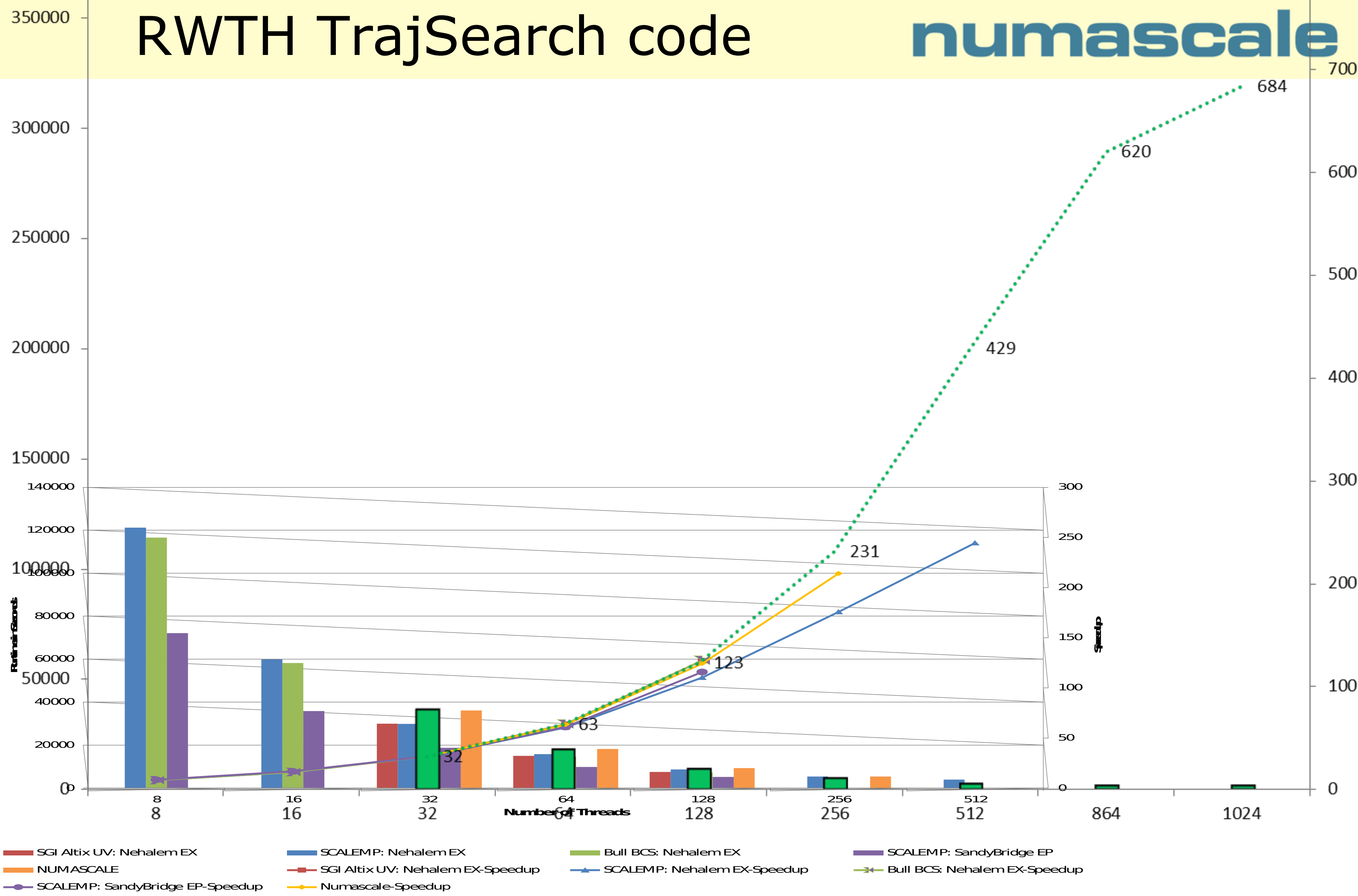


RWTH TrajSearch code



RWTH TrajSearch code

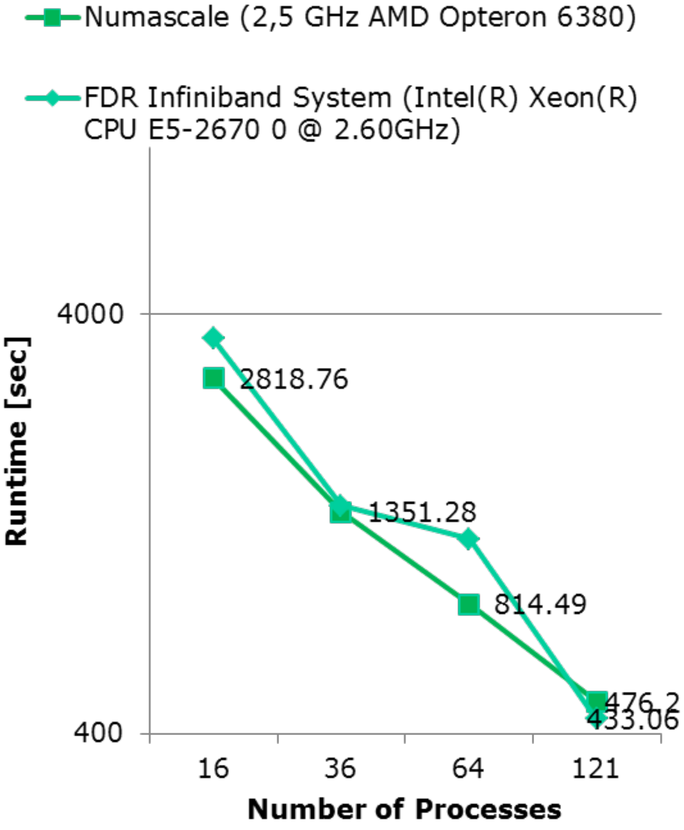
numascale



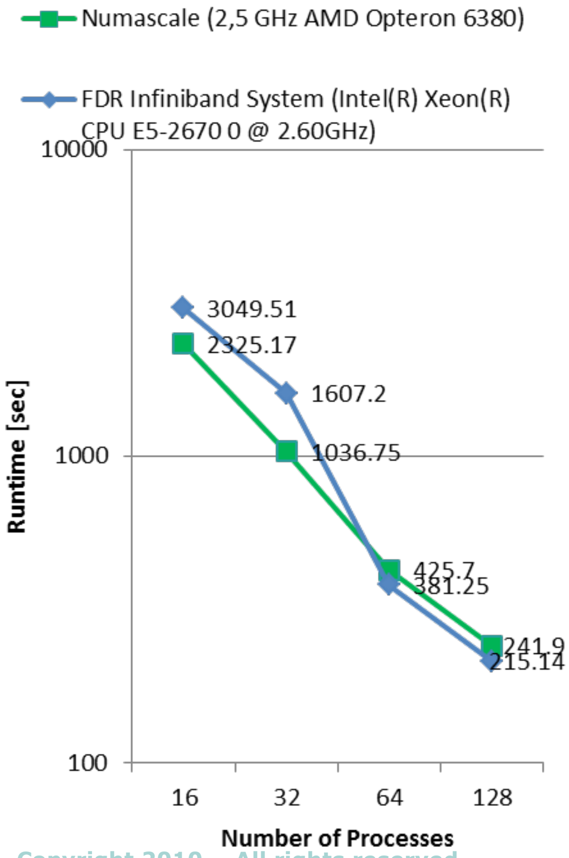
NAS Parallel benchmarks MPI

If you can get scalable OpenMP and MPI performance, ease of programming and ease of administration at commodity cluster price points, why limit yourself to an MPI cluster?

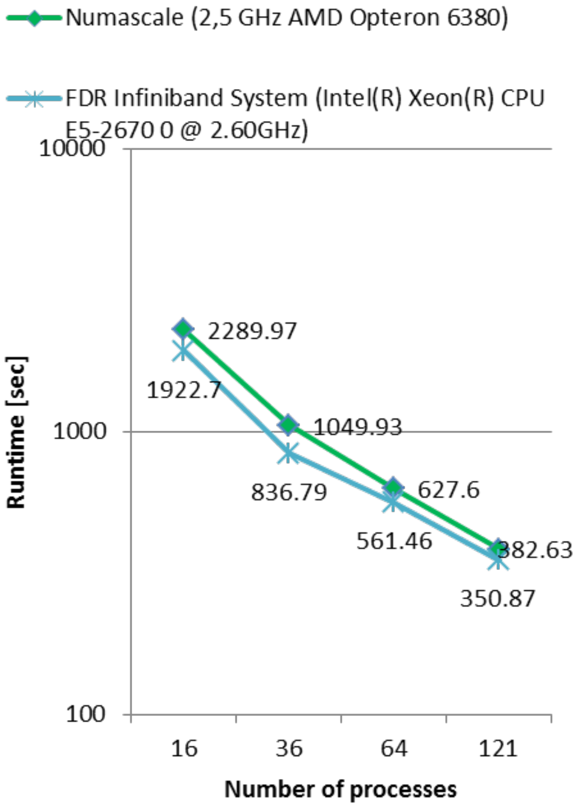
NPB-SP MPI CLASS D Time in seconds



NPB-LU MPI CLASS D Time in seconds



NPB-BT MPI CLASS=D Time in seconds

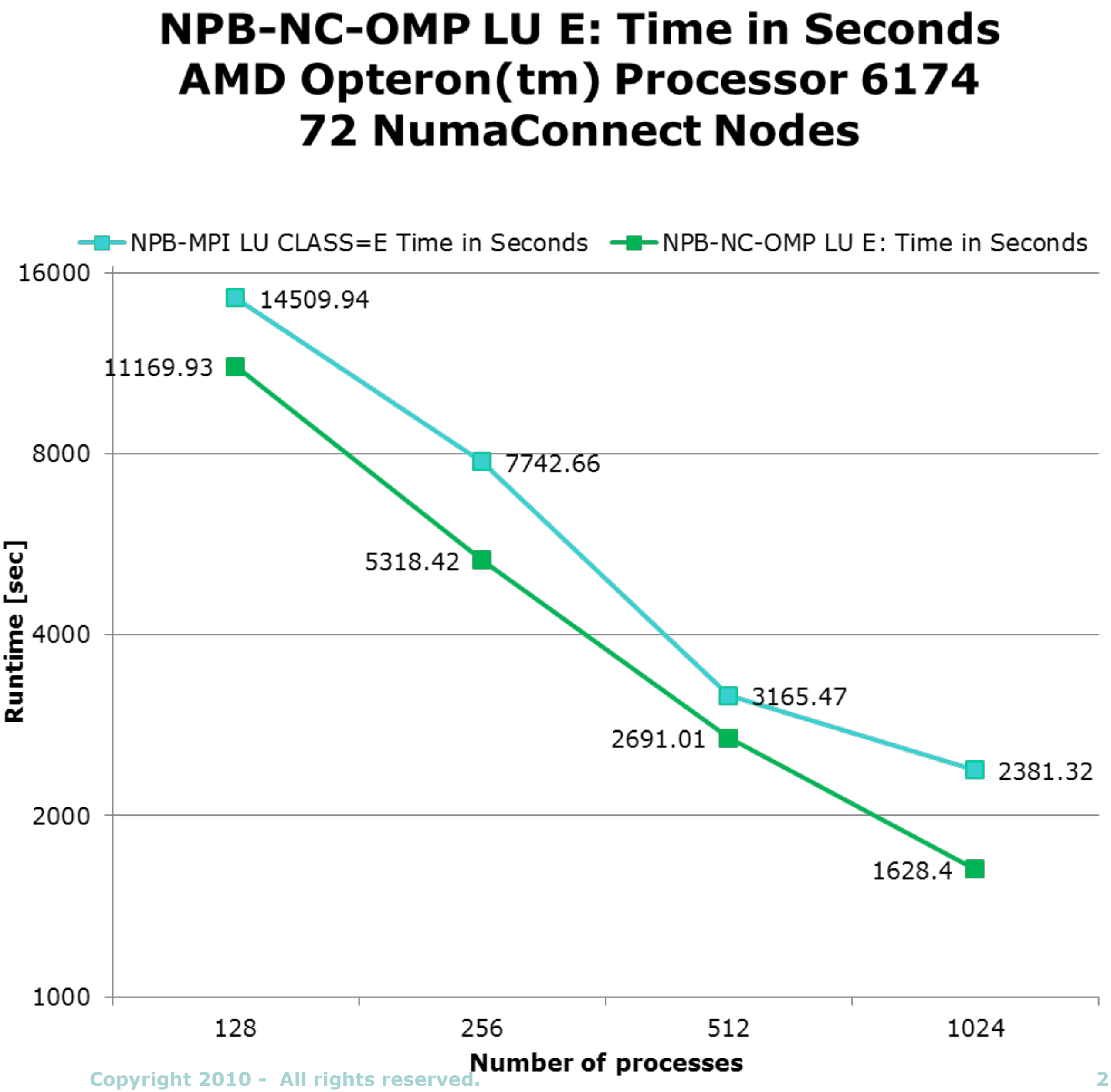


NAS Parallel benchmarks OpenMP runtime

numascale

The **overhead** introduced by MPI is not needed when we are running on a **Shared Memory System**

- ➢The NAS Parallel Benchmarks (NPB)
  - ➢ evaluate the performance of parallel supercomputers
  - ➢ derived from computational fluid dynamics (CFD) applications
  - ➢ LU is a simulated uses symmetric successive over-relaxation (SSOR) method to solve a seven-block-diagonal system resulting from finite-difference discretization of the Navier-Stokes

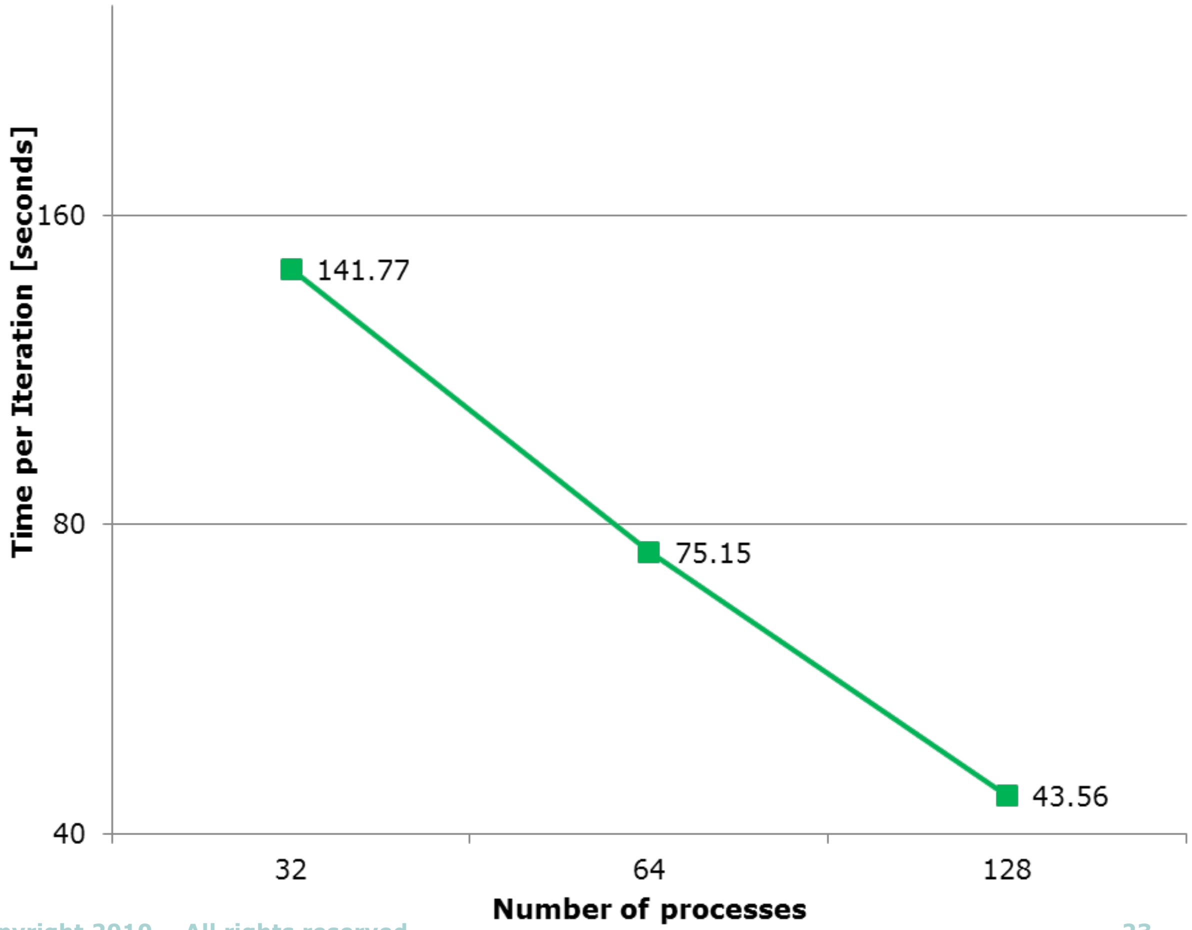


CD-adapco STAR-CCM+ (MPI)

- STAR-CCM+ is a technology leading Computational Fluid Dynamics (CFD) package unrivalled in its ability to tackle problems involving multi-physics and complex geometries, http://www.cd-adapco.com/products/star-ccm®

- The NumaConnect Shared Memory test system used to conduct the tests has:
- 1TB of memory
- 256 cores
- It utilizes 8 servers each equipped with:
  - 2 x AMD Opteron 2,5 GHz 6380 CPUs
  - 16 cores in each CPU
    -

STAR-CCM+
"Time per Iteration [seconds]"
Lower is better

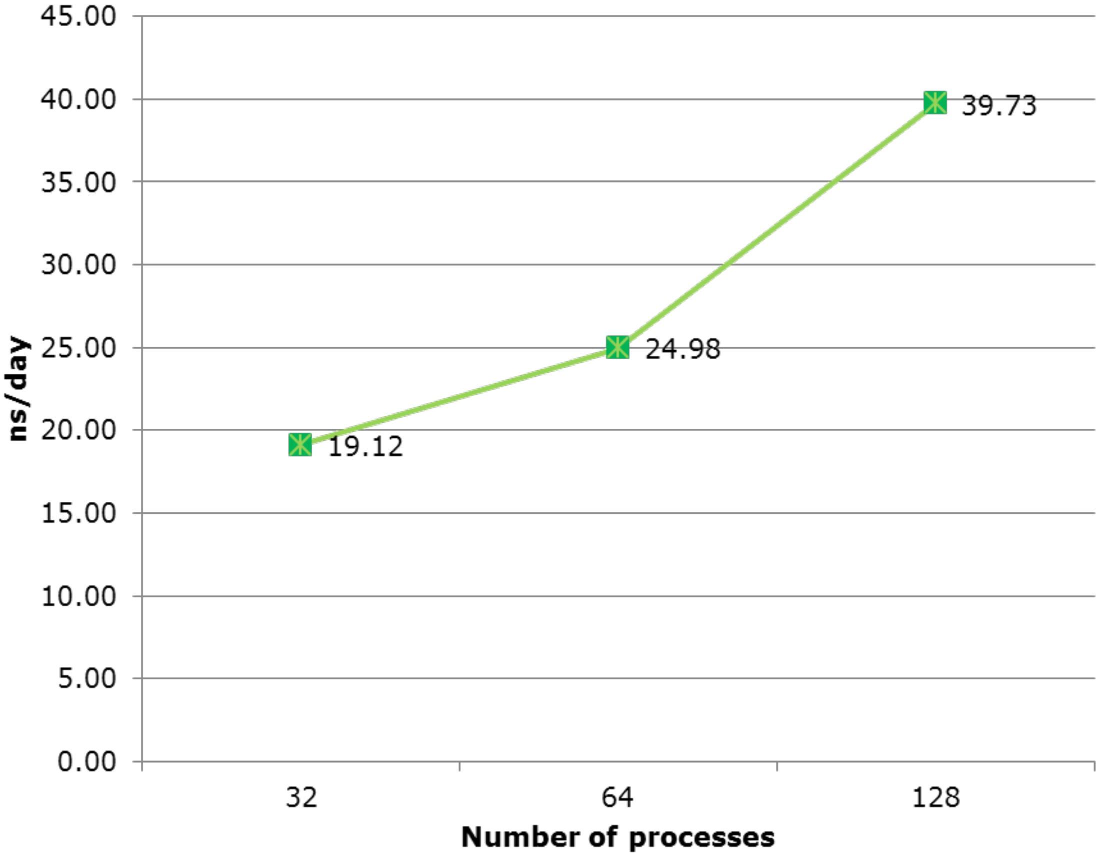


GROMACS

- **GROMACS** is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles. It is primarily designed for **biochemical molecules** like proteins, lipids and nucleic acids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the nonbonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers.

- The NumaConnect Shared Memory test system used to conduct the tests has:
- 1TB of memory
- 256 cores
- It utilizes 8 servers each equipped with:
  - 2 x AMD Opteron 2,5 GHz 6380 CPUs
  - 16 cores in each CPU
  - 128GB

GROMACS with NC-OpenMPI [ns/day] (higher is better) case: Test-performance_protein-water-membrane.tpr



Shared GPU, GPGPU

3 AIC Octans

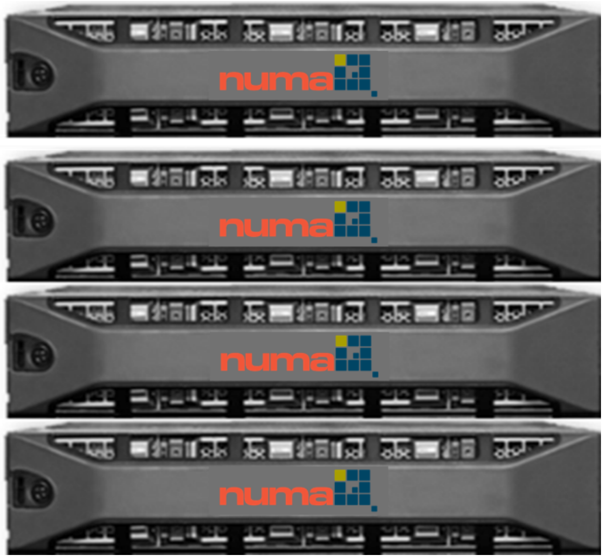
3 NVIDIA GeForce GT 640 2GB

Numascale Shared Memory System

- Cache Coherent Global Shared Memory and Shared IO
- All GPUs providing aggregated TFLOPS
- Running N-body CUDA application

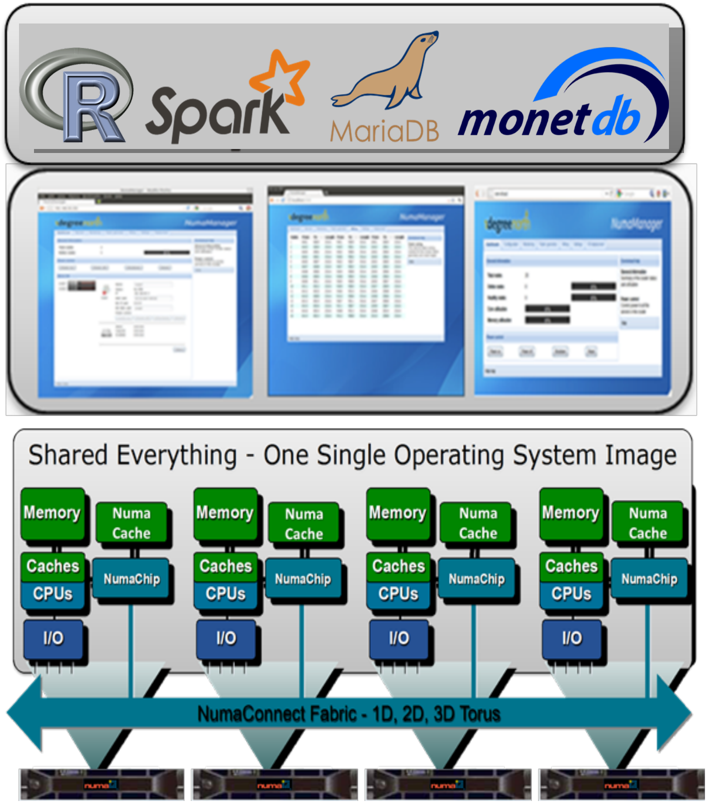
Numa-Q

In-memory analytics appliance



4 x Dell R815s

Running 1Billion row, Spark regression benchmark, 4X gain over cluster

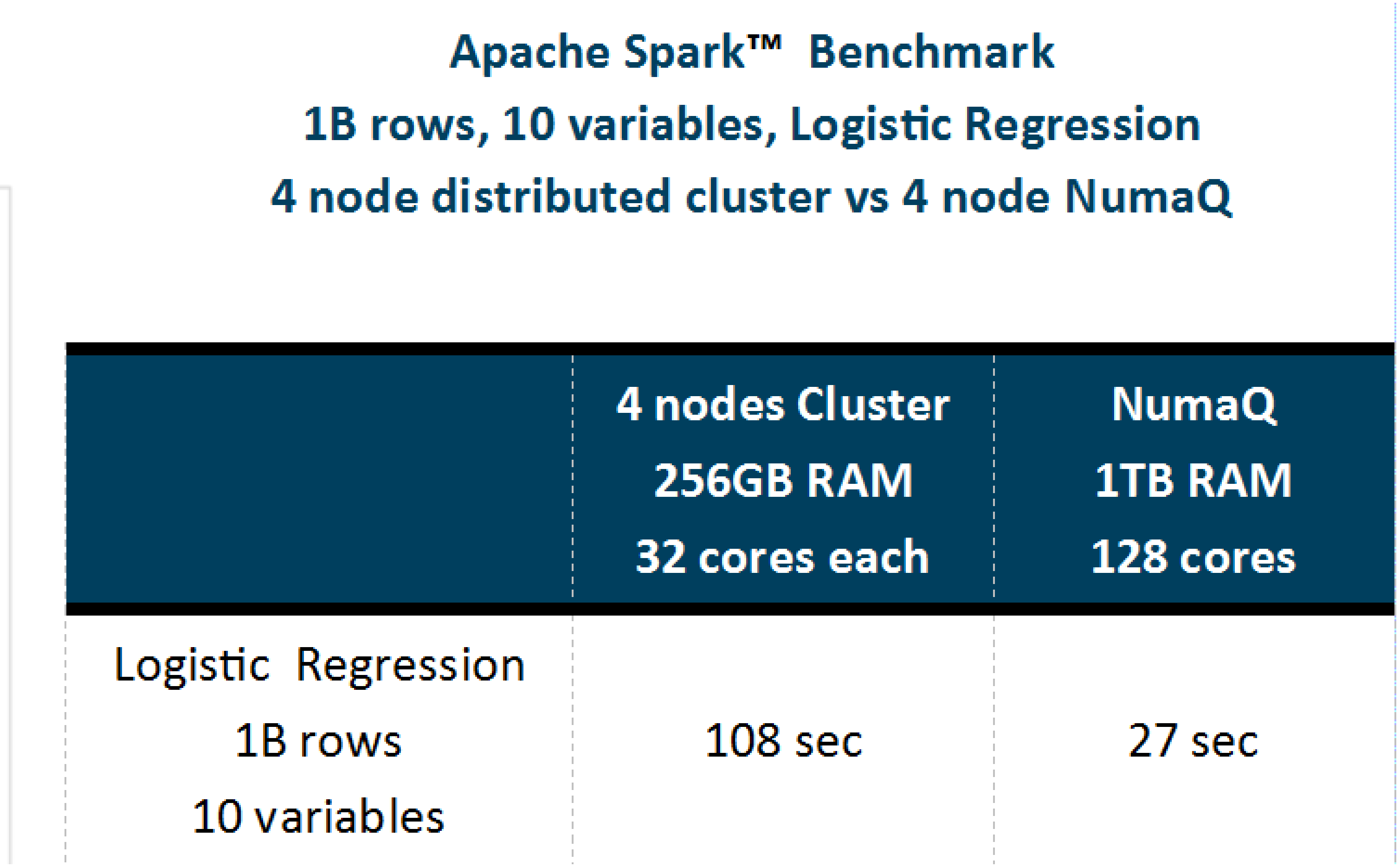


Advanced Analytics & Visualization
Simplified Management With NumaManager
Terabytes of Memory

Thousands of Cores

Single Linux Instance

Spark Benchmark



Apache Spark™ Benchmark

1B rows, 10 variables, Logistic Regression

4 node distributed cluster vs 4 node NumaQ

	4 nodes Cluster 256GB RAM 32 cores each	NumaQ 1TB RAM 128 cores
Logistic Regression 1B rows 10 variables	108 sec	27 sec

Competition - differentiation

	Perfor-mance	Shared Memory	Price	Comments
Software solutions with InfiniBand or 10Gbe (ScaleMP)	?	✓	2X	Software emulation Non-standard Operating System - Virtualization Layer
Mainframes (SGI, HP, IBM, Oracle (Sun))	High	✓	10-30X	• ”Max” performance – shared memory • 50TB limit for SGI • Limited Scalability
High-end interconnect for clusters (InfiniBand)	High		1X	• Pure message passing only
YarcData - uRiKA	High	✓	10X?	• Complete system solution – Big Data Appliance, Proprietary architecture
Numascale	High	✓	1X+	• Independent hardware vendor • Commodity server hardware

Shared Memory returning - Why?

- Compelling programming model
  - Less code
  - Large memories - less effort, no data domain decomposition
  - Flexibility
- System Utilization
  - More efficient utilization of Resources, up to 90% Hitachi mainframe Cambridge University, versus 50% cluster University of Oslo.
  - Reduced sysadmin’, single OS
  - Data Center Fabric
- NumaConnect
  - Turns COTs servers into SSI - ccNUMA , CHEAPLY!

With ASIC transistor density, and HT, it can now be done cheaply!

What’s best for scientific research?

- Cluster
  - Look at the size of my Linpack!
  - # nodes/cores
  - Grand Challenge benchmarks
  - Interesting Computer Science
  - Poor ROI
- Shared Memory
  - High system utilization
  - Easier sysadmin’
  - Easier programming
  - More research papers produced
  - Better science

NumaScale Futures

numascale

- Intel
  - XEON based solution
- AMD
  - Higher Link Speeds
  - Larger node memory
  - Higher core count
  - Hybrid CPU/GPU