Petascale Computing

Trends and Challenges

Herbert Huber, Department High Performance Systems, LRZ

www.lrz.de

System Architecture of Future PetaFlop Systems

- Several 100.000 processor cores in
  - large CC-NUMA partitions with more than 1000 physical cores and/or inhomogenous processor architecture as well as even more logical cores (SMT)
  - or thousands of small SMP-nodes
- N-dimensional (N ≥3) network torus with
  - position dependent communication latencies of the communication partners
  - position dependent per core network bandwidth
- Several MW of electrical power consumption

www.lrz.de

Challenges

Challenges of Petascale Computing

- Application scalability to several 100.000 processor cores
  - new programming models and languages are needed which exploit parallelism at all levels of a program;
  - in many cases users and ISVs have to redesign or completely rewrite a large fraction of their codes;
- Scalability of system management software
  - The OS as well as all other system management software components, e.g.,
    - resource managers,
    - batch schedulers,
    - system installation, configuration and monitoring software
    - must be capable to serve up to 1.000.000 processor cores in 10.000 compute nodes
    - the uncoordinated interruption of parallel processes by means of OS tasks (“OS noise”) has to be drastically minimized

www.lrz.de

Challenges

Challenges of Petascale Computing

- Network scalability in terms of number of clients, performance and costs
- I/O and disk storage in terms of number of clients connected to a single file system, file system performance, file system size as well as number of files in one file system (multi PByte & Millions of files) and time needed to recover from file system errors
- Reliability of hardware and software components
  - The MTBF of hardware components must be greatly enhanced;
  - The redundancy of critical hardware components like network and I/O adapters must be ensured;

www.lrz.de

Challenges

Challenges of Petascale Computing

- Hard- and software support for hybrid or heterogeneous architectures, i.e., systems containing GPUs or FPGAs as code accelerators
  - SDKs, Fortran as well as C/C++ compilers and optimized libraries have to be available for these architectures;
  - Batch scheduler and resource monitors have to support hybrid system architectures;
- In many cases the programming model of applications must be changed to optimally utilize accelerator hardware

www.lrz.de

How can Batch Scheduling Systems cope with these future PetaFlop Systems?

How can Batch Scheduling Systems help to reduce the electrical power consumption of these Systems?

www.lrz.de

HLRB II & Lessons learned

ORNL Symposium 2008

www.lrz.de

HLRB II: Altix 4700

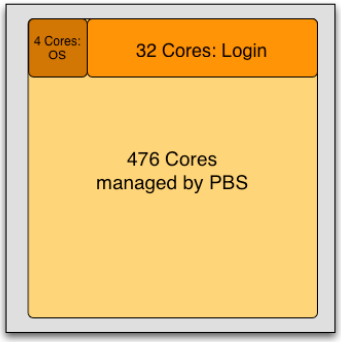
Overall characteristics for both installation phases:	Phase 1 (until 03/2007)	Phase 2 (since 04/2007)
Total number of cores	4096	9728
Peak Performance (entire system)	26.3 TFlops	62.3 TFlops
Linpack Performance	24.5 TFlops	56.5 TFlops
LRZ-Benchmark Performance	8.2 TFlops	16.2 TFlops
Size of memory (entire system)	17.5 TByte	39 TByte
Direct Attached Disks	300 TByte	600 TByte
Network Attached Disks	40 TByte	60 TByte
Processor type	Intel Itanium2 Madison 9M	Intel Itanium2 Montecito Dual Core
Clock rate	1.6 GHz	1.6 GHz
L3 Cache (per core)	6 MByte	9 MByte
Memory per core	4 GByte	4 GByte
Clock rate of frontside bus (FSB)	533 MHz	533 MHz
Peak bandwidth to local memory	8.5 GByte/s	8.5 GByte/s (shared between 2 or 4 cores)

HLRB II: Partition Layout

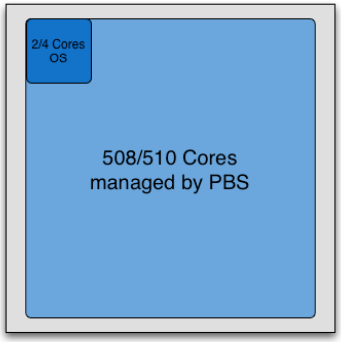
www.lrz.de

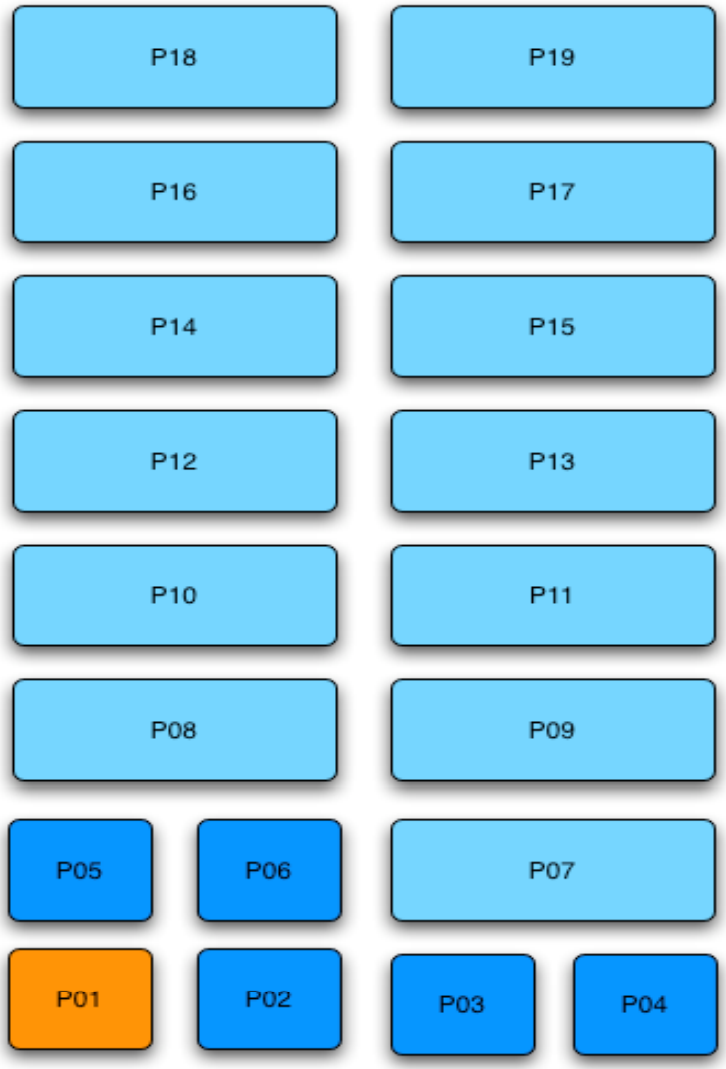
- 512 Cores per Partition
- 19 Partitions managed by PBSPro
- 13 High-Bandwidth partitions (light blue)
  - 2 cores share memory channel
  - 2 cores reserved for OS
- 6 High-Density Partitions (aqua + orange)
  - 4 cores share memory channel
  - 4 cores reserved for OS
  - 32 cores reserved for login in partition1
- Interaktive Jobs for compilation, tests and performance measuerments via PBS
- Advanced reservation possible for LRZ staff

Login Partition



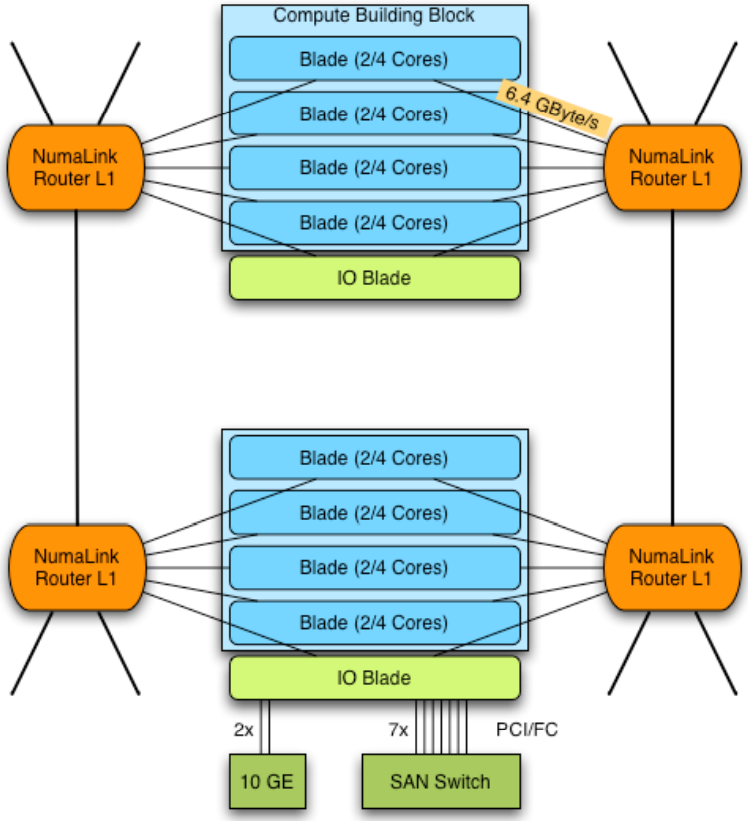
Regular Batch Partition

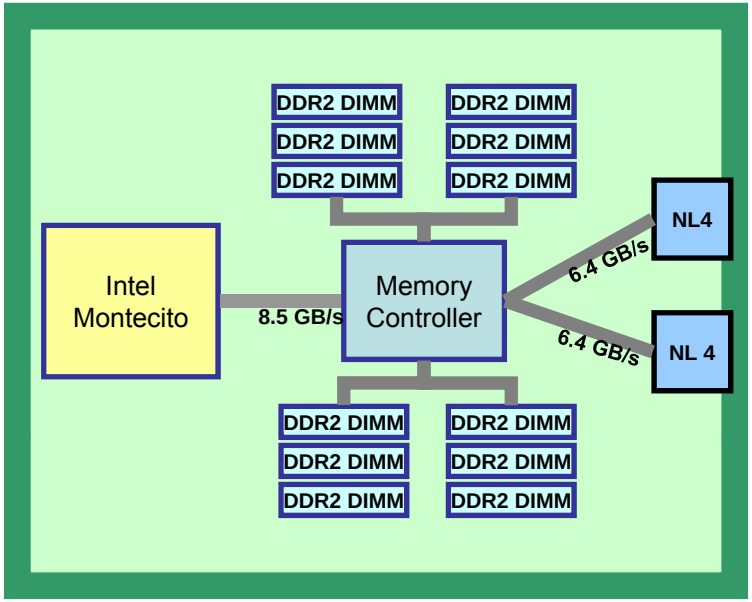


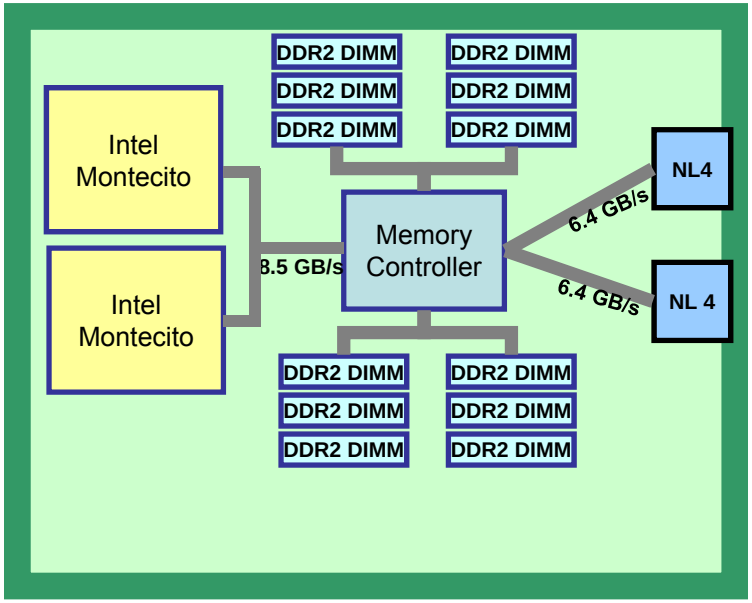


www.lrz.de

HLRB II: NUMALink Building Block

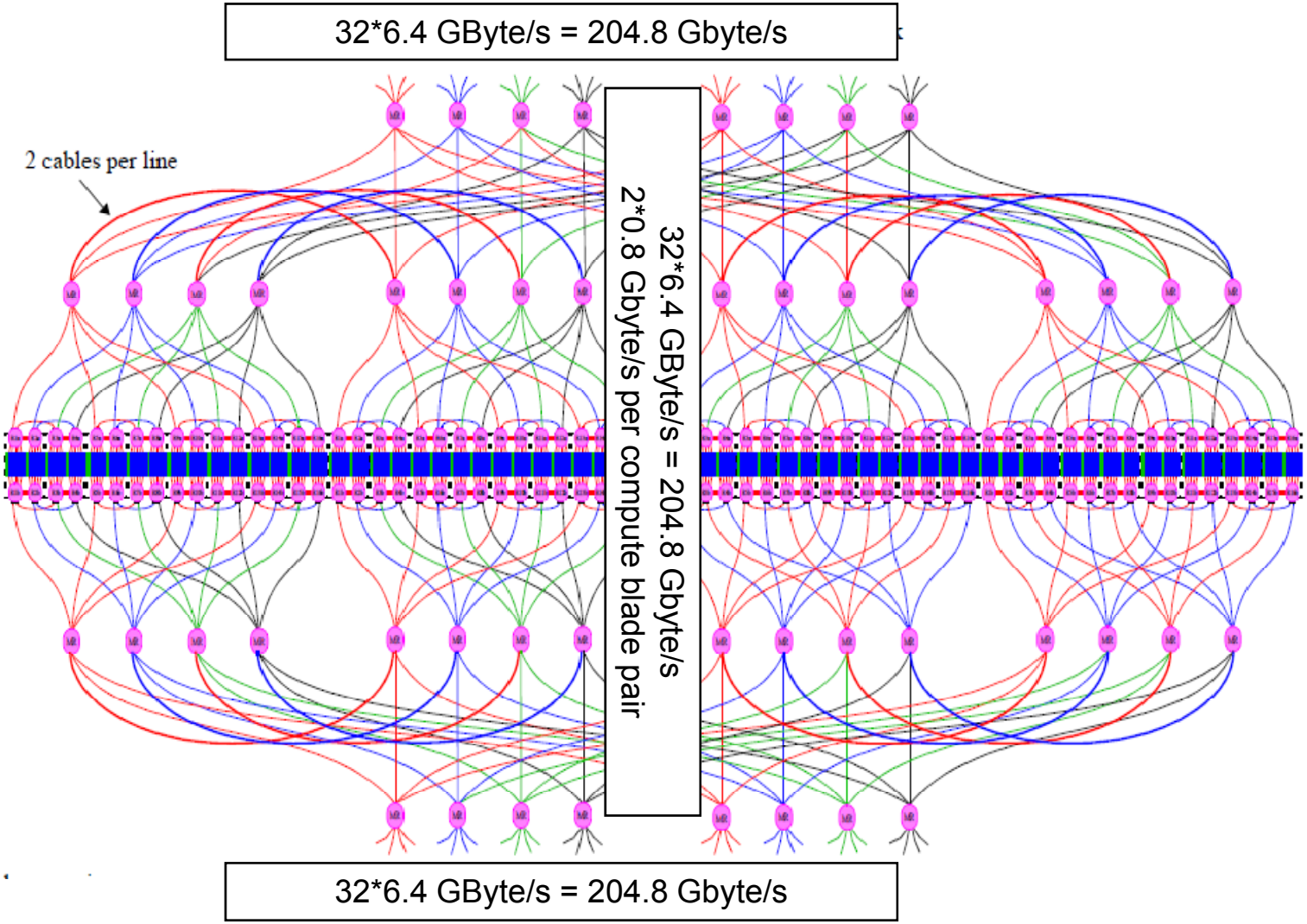






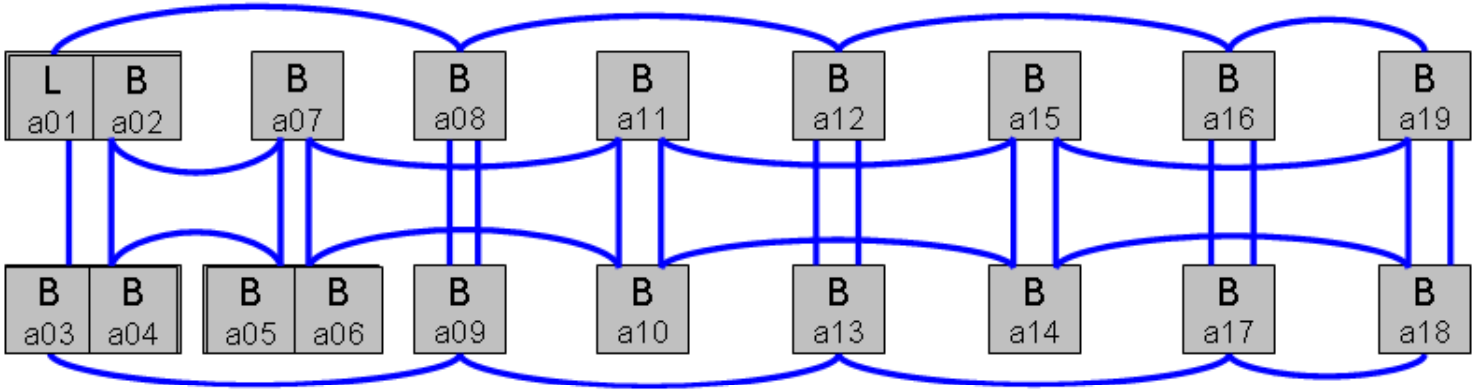
HLRB II: Fat Tree Building Block (Partition)

www.lrz.de



HLRB II: Inter-Partition NUMAlink configuration

Inter-Partition NUMAlink configuration



Each grey box:
- 512 core partition (SSI)
- L login, B batch

Each line represents:
- 2 NUMAlink4 planes with 16 cables (total)
- each cable: 2 * 3.2 GB/s
- high density partitions share cables in each line

Bisection Bandwidths per **blade** pair:
- intra-partition: 2 * 0.8 GB/s
- any 2 vertical partitions: 2 * 0.4 GB/s
- 4 partitions: 2 * 0.2 GB/s
  assumes only 32 cables used (shortest paths)
  many hops for bad choice → less bandwidth
- total system: 2 * 0.1 GB/s

(factor 2 indicates "per direction")

Site specific Job Placement Strategies

HLRB II example:

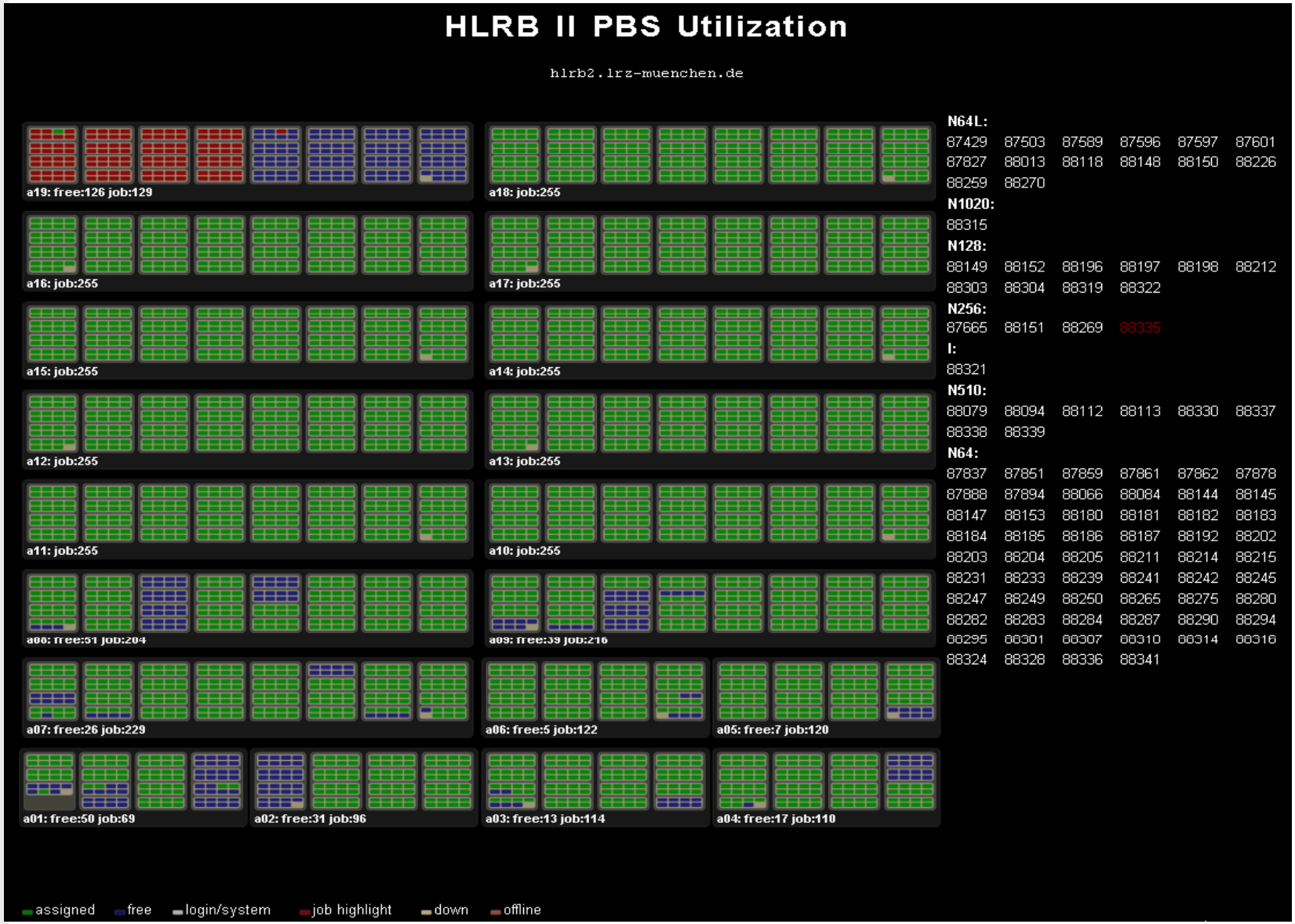
- Definition of so called placement sets for an optimal placement of jobs
  - Group tightly coupled cpu and memory resources (vnodes)
  - Placement sets are determined by system topology
  - Placement sets defined: SHub, IRU, IRUpair, rack, rackpair, host, rank, rank2, rank4, universe
- Definition via resource for each vnode
- Vnode grouping on partition, queue and job level
- Cpusets as job containers
  - Job isolation (nodes allocated to one job exclusively)
  - Ideal placement

Scheduling Tools: HLRB II & Lessons learned

- **GUI to display actual job placements**
- **Standardized API to add site specific scheduling functionality**
  - Site specific job placement strategies
  - Site specific scheduling strategies
    - Optimize scheduling logic for large jobs in an environment where a large number of small and medium size jobs are waiting in the scheduling queue
    - Tunables for temporary acceptable unoccupied processor capacity
    - Backfill strategy taking into account real runtimes of user jobs in the past

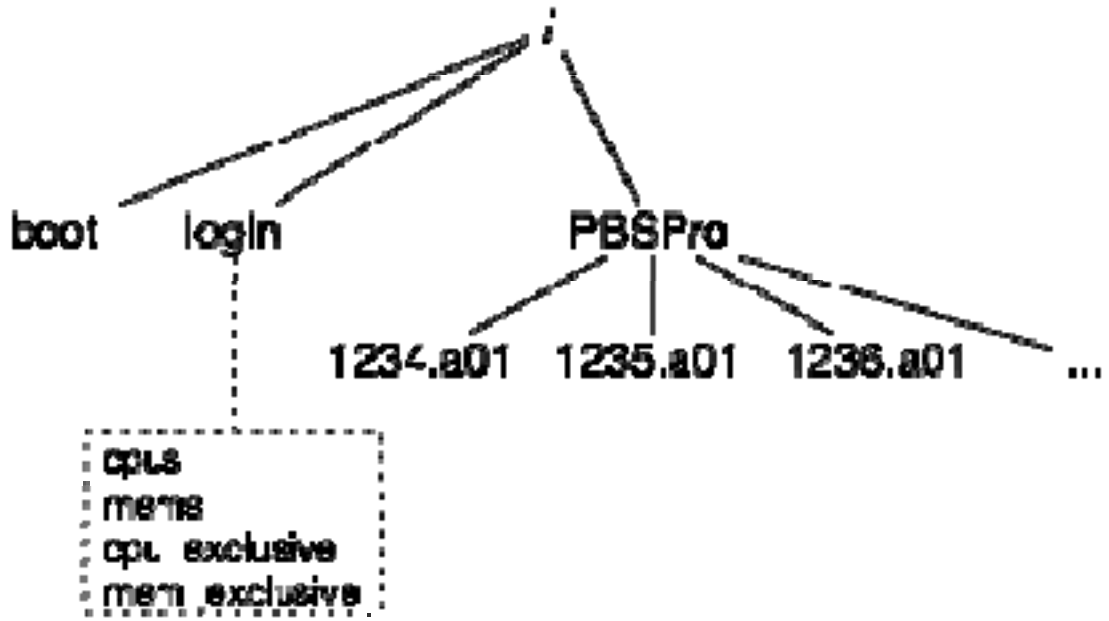
Site specific Job Placement Strategies

www.lrz.de



Cpusets

- Named set of cpu and memory nodes
- Used for cpu and memory placement
- Direct support in Linux kernel



Batch Schedulers: HLRB II & Lessons learned

- **Availablity enhancements**
  - Redundant server and scheduling daemons
- **Scalability enhancements**
  - (Multithreaded) execution daemons capable to serve very large SMP partitions
  - (Multithreaded) scheduling daemon capable to serve hundred thousands of cores and placement sets
  - Communication protocol able to handle hundred thousand nodes/connections
- **Energy saving enhancements (energy-aware scheduling)**
  - Turn unused HW (nodes, cpu cores, I/O adapters) to sleeping mode

Possible Fields of Cooperations

- Topology- and energy-aware scheduling

- Enhancing energy efficiency of large computing centers
  - Optimize power usage efficiencies (PUE)
    - Best practices
    - Innovative approaches in power and cooling infrastructures