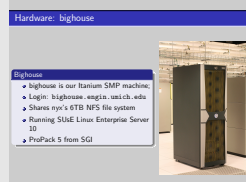


2007-10-05

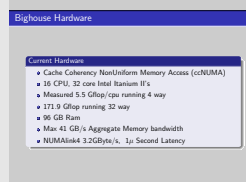
Bighouse Crash  
 Resources  
 Configuration  
 Hardware: bighouse



**ProPack:** Provides performance tools, hardware tools and MPT(MPI) libraries

2007-10-05

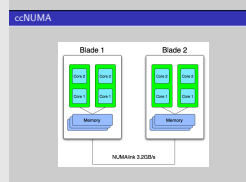
Bighouse Crash  
 Resources  
 Hardware  
 Bighouse Hardware



1. Keeps cache lines in sync both good and bad  
 Makes for easy programming, places upper limit vs. CRAY
2. HPL P=2 Q=2 N=20000, MKL no threads, MPT
3. HPL P=4 Q=8 N=20000, MKL no threads, MPT
4. 2 nodes have 24GB, 6 have 8GB

2007-10-05

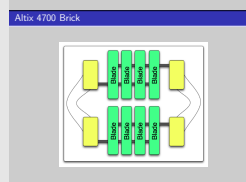
Bighouse Crash  
 Architecture  
 ccNUMA  
 ccNUMA



1. Itanium II's 9000's.  
 L1 16k/d 16k/i  
 L2 256k/d 1024/i  
 L3 4MB
2. SHUB2 I FORGOT IT!  
 It Sits between the cpus and memory Numa link connects to it.  
 This is where the magic happens

2007-10-05

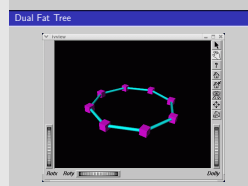
Bighouse Crash  
 Architecture  
 Altix 4700 Brick  
 Altix 4700 Brick



1. Each blade has 2 NUMALink connections, each goes to a differnt router, each router has a 200 nanoSec pass time.

2007-10-05

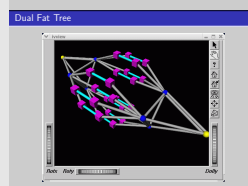
- Bighouse Crash
  - Architecture
    - Dual Fat Tree
      - Dual Fat Tree



1. This would be our layout but turns out its not this would apply to the 450 if we had it.

2007-10-05

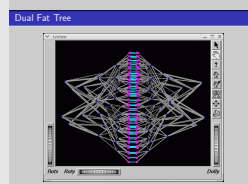
- Bighouse Crash
  - Architecture
    - Dual Fat Tree
      - Dual Fat Tree



1. this is our layout (at 8 blades), We only have half the ring though, max number of hops will equal up to 16 blades 64 cores

2007-10-05

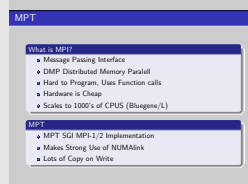
- Bighouse Crash
  - Architecture
    - Dual Fat Tree
      - Dual Fat Tree



1. Provides 1024 cores 512 sockets  
This is the max supported config from SGI, system can add one more router out for 1024 sockets 2048 cores, MTTF is to high though

2007-10-05

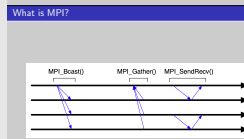
- Bighouse Crash
  - Software Performance
    - MPI Code
      - MPT



1. [www.mpi-forum.org](http://www.mpi-forum.org)
2. We have similar SM ability on nyx though OpenMPI

2007-10-05

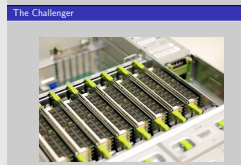
Bighouse Crash  
└ Software Performance  
    └ MPI Code  
        └ What is MPI?



1. Duplicates allot of data between processes
2. nothing shared unless given

2007-10-05

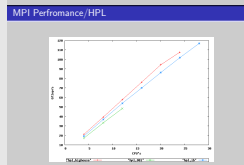
Bighouse Crash  
└ Software Performance  
    └ MPI Code  
        └ The Challenger



1. nyx801 Owned by Dr. J Norman MD, PHD.  
64 GB ram, on 8 sockets, dual core 8218's

2007-10-05

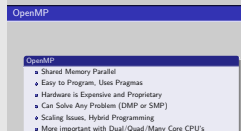
Bighouse Crash  
└ Software Performance  
    └ MPI Code  
        └ MPI Performance/HPL



1. **Hardware:**  
'hpl.bighouse' is bighouse
  - mpt
  - mkl no thread'hpl.801' is nyx801  
'hpl.ib' is EMike Nodes, dual core dual socket opt2220, 16 GB ram, DDR Infiniband 20Gbit/s < 4μ Sec. Latency
  - openmpi-1.2-pgi, OFED
  - goto-blas
2. point out gapping as number of CPUS increase  
Why Bighouse is superior, but not at this size and price

2007-10-05

Bighouse Crash  
└ Software Performance  
    └ OpenMP Code  
        └ OpenMP



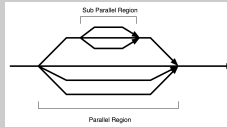
1. Thread sync issues, implemented with libpthread normally
2. in GCC 4.1
3. Can FLOOD bus/interconnect because of cache sync issues

2007-10-05

## Bighouse Crash

- └ Software Performance
  - └ OpenMP Code
    - └ OpenMP Fork and Join

### OpenMP Fork and Join



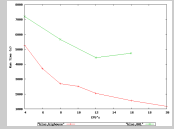
1. This is what all of Direct CAE apps use (Nastran Abaqus)  
Most iterative solvers are dense matrix solvers in DMP (LS-DYNA)
2. **STRESS** This is Bighouse's benefit, it can take the RAM and SMP ability to run these codes at a speed a regular cluster could never do

2007-10-05

## Bighouse Crash

- └ Software Performance
  - └ OpenMP Code
    - └ OpenMP Performance/dgemm 36,621 MByte

### OpenMP Performance/dgemm 36,621 MByte



1. [www.netlib.org/blas](http://www.netlib.org/blas)  
Bighouse uses MKL  
nyx801 Uses ACML-pgi-mp  
2 equal square matrix's of random numbers with a dim of: 40,000 Doubles.  
This is 3,200,000,000 (3.2 billion numbers)  
Same building block used in hpl

2007-10-05

## Bighouse Crash

- └ Software Performance
  - └ OpenMP Code
    - └ Example Cases

### Example Cases

- Example Cases
- NUMA Memory Placement dlook(1) dplace(1) dprint(1)
- Example Cpuset OH NO SWAP
- Memory placement cNUMA knows where to put memory (numa.h, numa.h)
- Example stream.c measures memory bandwidth

- **Example 1**, cpu sets
  - `cpuset -c brockp -f brockp/cpuset.conf`
  - `echo $$ >> /dev/cpusets/brockp/tasks`
- **Example 2**, link speeds
  - `linkstat -A`
  - `pmchart numa.mem.util.used`
  - `pmchart numa.link.send_bytes`
  - **run stream.c**