



## Lustre, SGE, & Physics Codes

R. Krivec  
*Department of Theoretical Physics  
J. Stefan Institute  
Ljubljana, Slovenia*

## A small GRID WITH (LOCAL) BRAGGING RIGHTS

- *Lustre 1.6.5 cluster* : **Infiniband**

- **X4200** MDS-1, MDS-2

- **X4600** 16 CPU 96 GB 64 Gbps

- **X4600** 16 CPU 96 GB 64 Gbps

- **X4500** OSS-1 6 TB 640 MBps/5 Gbps

- *Older servers* : **Ethernet**

- **v40z** 8 CPU 32 GB 26 Gbps

- **v40z** 8 CPU 32 GB 26 Gbps

- *Submission hosts* (**Sun Grid Engine 6.2**)

- **workstations** (Solaris, Linux)

- **SunRays**

- ...



## USER ENVIRONMENT: THE PAULI EFFECT (1/2)

- *heterogeneous grid*
  - client system and \$HOME areas remain independent
- *work area is unified*
  - `cd /grid/<username>`  
`qsub -cwd ...`
  - Lustre errors prevent me from declaring /grid a stable storage
- *openMP within 16 CPU*
  - three-body problem (collab. with Hebrew University)
  - superconductivity simulations, collab. with Kyoto University
  - collaborations tend to push users to do MPI instead

## HOW WE GOT TO LUSTRE

- **up to 76% CPU efficiency**  
three-body problem code (wave function of He atom to 15 decimals)

[http://www-f1.ijs.si/~krivec/bench/bench\\_bm\\_cfhh1.txt](http://www-f1.ijs.si/~krivec/bench/bench_bm_cfhh1.txt)

- **convert code to openMP** (1995)  
5-deep loop: inner (vector)  $\implies$  outer (parallel)  
single **LOOP\_PARALLEL** directive

[http://www-f1.ijs.si/~krivec/bench/par\\_dp\\_99a.pdf](http://www-f1.ijs.si/~krivec/bench/par_dp_99a.pdf)

- **useful even in OpenGL**  
DMA textures transfer in async thread
- **who said MPI?**

# SIMPLE LUSTRE BANDWIDTH TEST

- -----  
Number of CPUs: 16  
Number of threads: 8  
Total wall time: 19.927 s  
Total file size written: 6103.516 MB  
Average thread calculation time: 6.710  
Average thread I/O time: 11.283 s  
Aggregate I/O bandwidth: 540.971 MBps (from average I/O time)  
-----  
47.110u 33.910s 0:00.00 0.0% 0+0k 0+0io 0pf+0w

- **balanced:** saturates at 16 threads
- **expected:** most Lustre activity is on single OSS (60% CPU)

## GETTING USED TO LUSTRE (1/3)

- **NTP, YP**

is restarting ypbind periodically harmful?

- **May 27 MDS-1, OSS:**

09:50:08 mds-1 ... haven't heard ... .1@o2ib ... 241 seconds. ... evicting

09:56:27 oss-1 [same]

10:01:15 oss-1 ... haven't heard ... .101@o2ib ... 266 seconds. ... evicting

**lctl conf\_param ifs01-MDT0000.sys.timeout=1000**

Voltaire switch NOT reset

“EDAC k8 MC0:” errors every few days

- **Jul 15**

08:52:36 cn-1 ... 11-0: ... communicating ... .101@o2ib ... mds\_getattr ... -2

08:52:36 cn-1 ... 6144:0:(llite\_nfs.c:87:search\_inode\_for\_lustre())

failure -2 inode 18937377

NFS export overload on X4600, or RAM errors?

## GETTING USED TO LUSTRE (2/3)

- **Aug 08 first X4600 (NFS exporter) crashes**

10:42:30 **cn-1** ... EDAC k8 MC0: ... ECC chipkill x4 error

hard reset (via SP)

NFS export moved to second X4600

- **Aug 09 unexplained errors:**

17:31:09 **mds-1** ... nid **.111@o2ib** ... lost; ... wait for recovery to complete.

17:31:09 **mds-1** ... ost\_statfs operation failed with -107

17:47:49 **cn-1** ... :(mdc\_locks.c:586:mdc\_enqueue()) Idlm\_cli\_enqueue: -4

17:47:50 **cn-1** ... 11-0: ... communicating **.101@o2ib** ... mds\_connect ... -16

18:04:30 **mds-1** ... time source ... instable or ... driver ... hogging interrupts

18:11:16 **oss-1** ... haven't heard ... **.101@o2ib** ... 2405 seconds. ... evicting

Lustre recovery due to NFS exporter crash?

rebooted entire Lustre incl. Voltaire switch, recovery timer starts

forgot to set domainname on MDS, **errno: -43** cease after fixing next day

so far so good ...

## GETTING USED TO LUSTRE (3/3)

- **... or not? Aug 31**

08:57:13 **cn-1** ... pnic1v.e[12675]: segfault ... error 4

09:52:49 **cn-2** ... pnic1v.e[12174]: segfault ... error 4

but no "EDAC k8 MC0:" errors

just the "heartbeat ... These are nothing to worry about" messages

- **Sep 2**

11:24:57 **cn-2** ... 11-0: ... comm... **.101@o2ib** ... mds\_getattr\_lock ... -13

11:24:57 **cn-2** ... 30035:0:(llite\_nfs.c:252:ll\_get\_parent()) failure -13

inode 17530955 get parent

**30035 = NFSD**

11:24:57 **mds-1** ... 6136:0:(ldlm\_lib.c:1536:target\_send\_reply\_msg()) ... (-13)

**6136 = ll\_mdt\_12**



## GETTING USED TO LUSTRE (4/3)

- **Questions**

could “EDAC k8 MC0:” errors indirectly cause it?

worry about “These are nothing to worry about” messages?

NFS to multicore SMP/single eth0 dangerous (error -43)?

IB switch reboot?

Thanks.

APPENDIX ...

# APPENDIX: SIMPLE LUSTRE BANDWIDTH TEST

## ● sample output

```
mp1c.f90(45): (col. 9) remark: OpenMP DEFINED LOOP WAS PARALLELIZED.  
mp1c.f90(41): (col. 9) remark: OpenMP DEFINED REGION WAS PARALLELIZED.  
mp1c.f90(82): (col. 5) remark: LOOP WAS VECTORIZED.  
mp1c.f90(89): (col. 5) remark: LOOP WAS VECTORIZED.  
mp1c.f90(56): (col. 7) remark: LOOP WAS VECTORIZED.
```

---

Processors: 16 Threads: max 8

thread	slice	a(n1,n2,slice)	time	status
1	2	36.631366231498	5.3650	computed
0	1	36.631252893733	6.4449	computed
2	3	36.631479568491	6.5456	computed
7	8	36.632046241886	6.7199	computed
4	5	36.631706240163	6.7610	computed
6	7	36.631932908750	6.8278	computed
3	4	36.631592904713	6.9834	computed
5	6	36.631819574842	8.0329	computed
1	2		6.4449	wrote file
0	1		10.8367	wrote file
4	5		11.6963	wrote file
3	4		11.7790	wrote file
2	3		12.2206	wrote file
6	7		12.5516	wrote file
5	6		11.5612	wrote file
7	8		13.1700	wrote file

- (cont'd)

```
-----  
Number of CPUs:                16  
Number of threads:             8  
Total wall time:               19.927 s  
Total file size written:       6103.516 MB  
Average thread calculation time: 6.710  
Average thread I/O time:       11.283 s  
Aggregate I/O bandwidth:       540.971 MBps (from average I/O time)  
-----  
47.110u 33.910s 0:00.00 0.0% 0+0k 0+0io 0pf+0w
```

- **balanced:** saturates at 16 threads
- **expected:** most Lustre activity is on single OSS (60% CPU)

# TEST PROGRAM

```
...
np = OMP_GET_NUM_PROCS()           ! number of CPUs available
call OMP_SET_DYNAMIC(.FALSE.)

call OMP_SET_NUM_THREADS(n3)       ! set number of threads

t0 = zwttime()                     ! initialize timer
mt = OMP_GET_MAX_THREADS()         ! check max. allowed threads
...

!$OMP PARALLEL

nt = OMP_GET_NUM_THREADS()         ! check number of threads allowed
...
```

(cont'd)

```
!$OMP DO PRIVATE(i1, i2, i3, jt, jn)

do  i3 = 1,n3
  ts(i3) = zwttime()           ! absolute time at start of thread
  jt = OMP_GET_THREAD_NUM()   ! current thread number
  jn = OMP_GET_NUM_THREADS()  ! check number of threads

  do  i2 = 1,n2
    do  i1 = 1,n1
      a(i1,i2,i3) = ...           ! CPU
    enddo
  enddo

  tc(i3) = zwttime() - ts(i3)    ! thread computation time
  ...

  write (10+io+i3) ((a(i1,i2,i3), i1 = 1,n1), i2 = 1,n2)      ! I/O
  tw(i3) = zwttime() - (ts(i3) + tc(i3))  ! thread disk write time
  ...
enddo
```

## SOLARIS → LINUX: THE PAULI EFFECT (2/2)

- *Solaris/Sparc*

- everything was working

- \* **large RAM was P.O.C. (piece of cake)**

- *Sparc → AMD*

- oh no, it's MHz again

- \* **but: Sun Studio on x86 did not support 128-bit arithmetic**

- *Linux*

- everything is working

- \* **large RAM is also P.O.C. (proof of concept)**