Performance Measurement and Visualization on the Cray XT4 MPI Recent Work

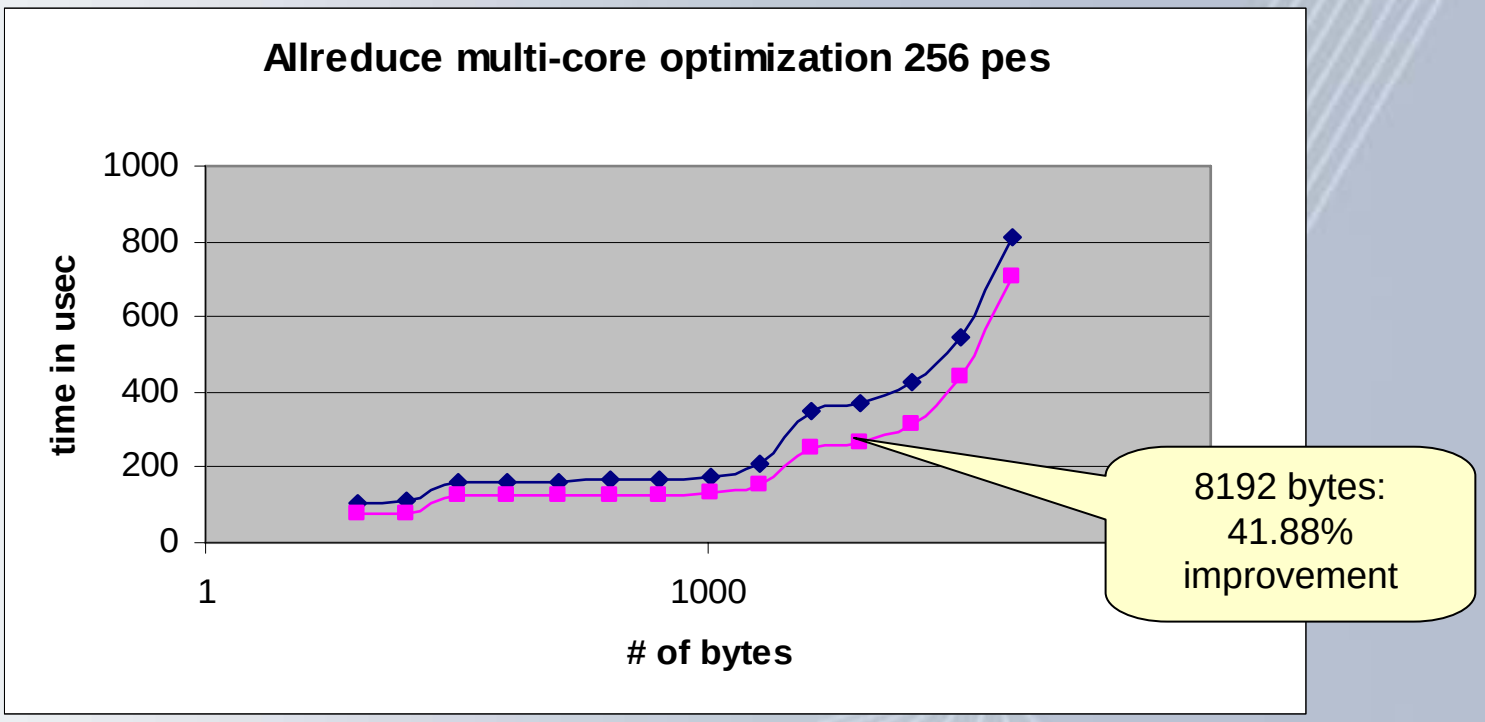
Luiz DeRose
Programming Environment Director
Cray Inc.
ldr@cray.com

NERSC
September 18, 2007

Recent MPI Work

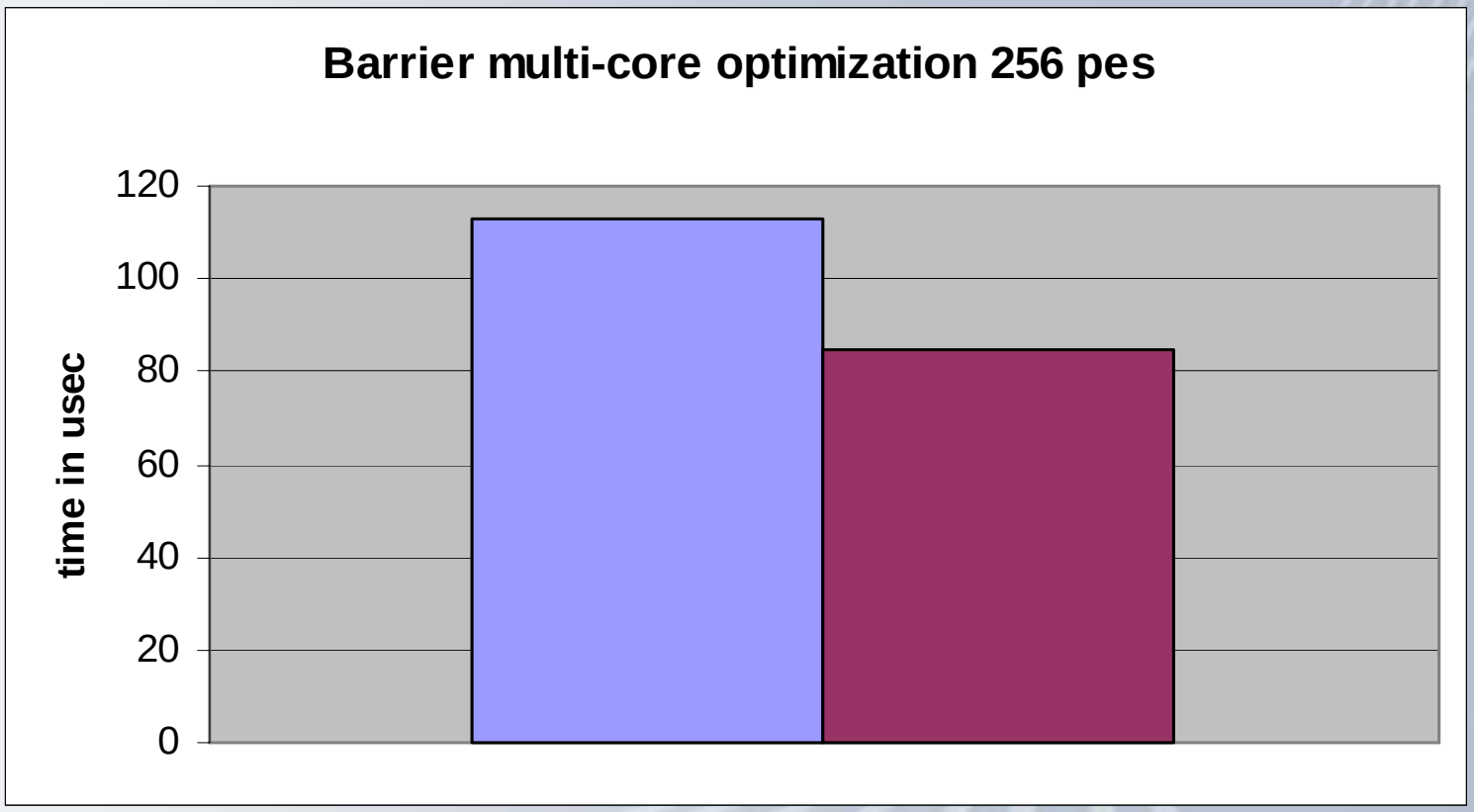
- MPI dual-core performance improvements
  - Short-circuit and master/slave optimizations
    - Allows SMP-aware collective optimizations
    - Expect latency near 0.5 usec for 0 byte on node
  - Rank Placement (MPICH_RANK_REORDER_METHOD – 1.5.08)
    - DLPOLY – SMP style – 13% improvement (512pes)
    - HPCC(PTRANS) – custom placement -> 2x improvement (10000pes)
  - MPI collectives optimizations (MPI_COLL_OPT_ON – 1.5.11)
    - MPI_Allreduce - 30% improvement for 16K bytes or less
    - MPI_Barrier – 30% improvement for 256 PEs in Pallas
  - Optimized memcpy (MPICH_FAST_MEMCPY - 1.5.30)
  - Portals match off (MPICH_PTL_MATCH_OFF – 1.5.39)

Allreduce 256 PEs dual-core



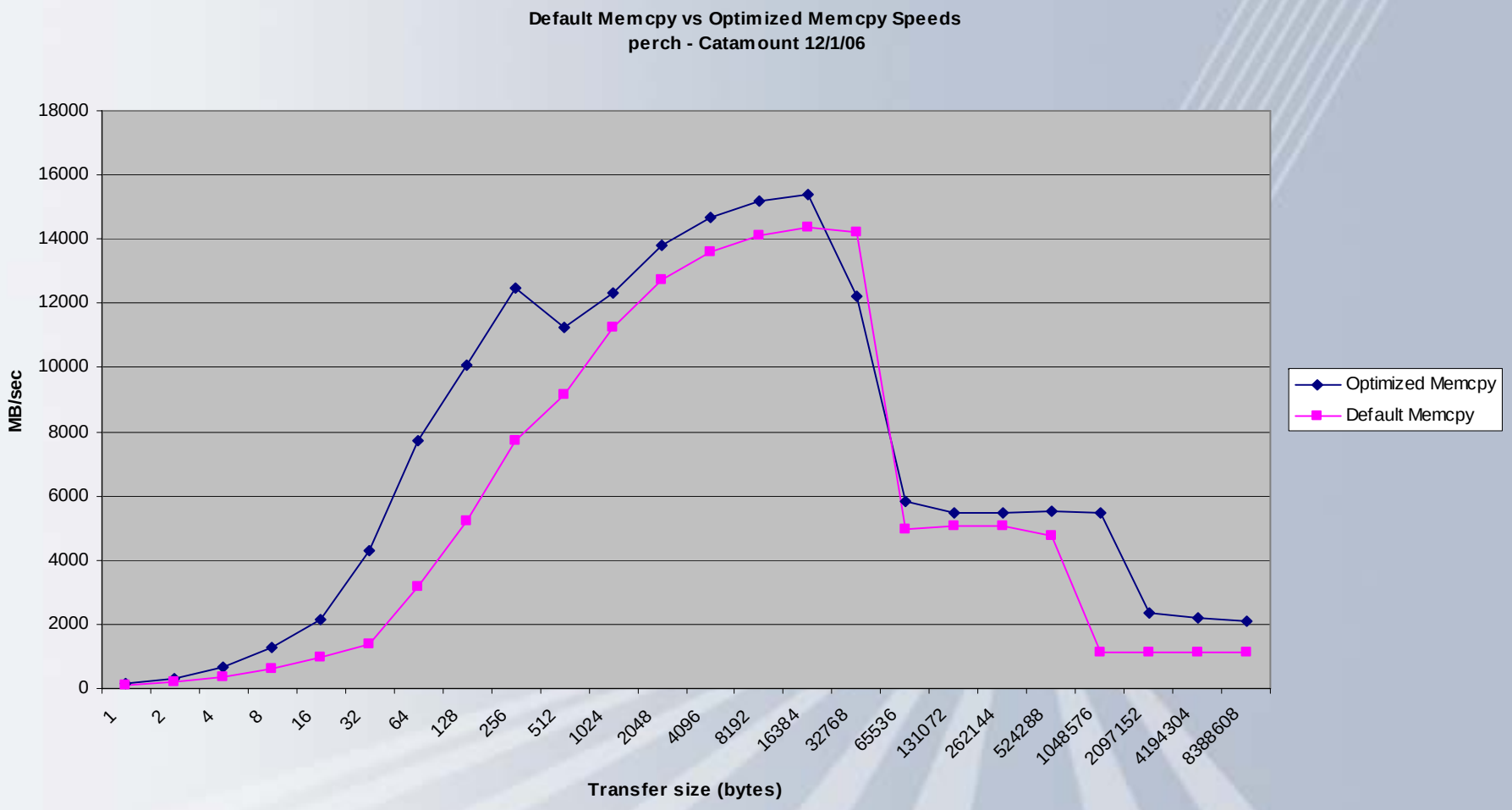
- Optimization focuses on messages below 65K
  - see at least 14% performance improvement for Pallas Allreduce

Barrier improvements 256 PEs

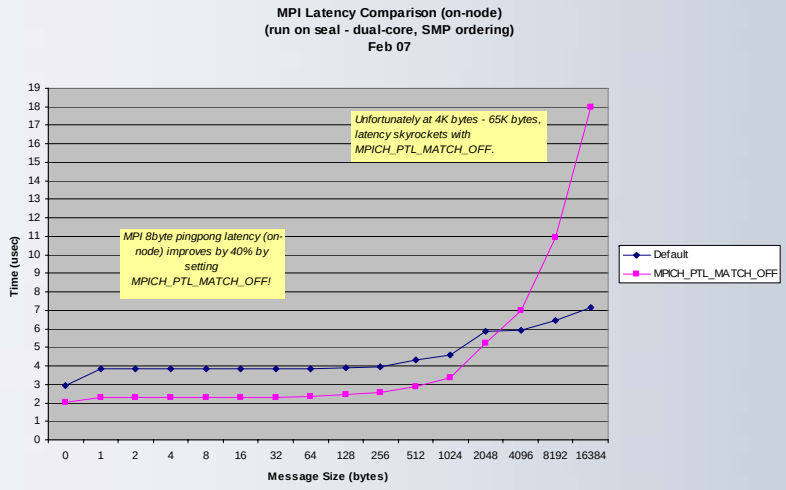


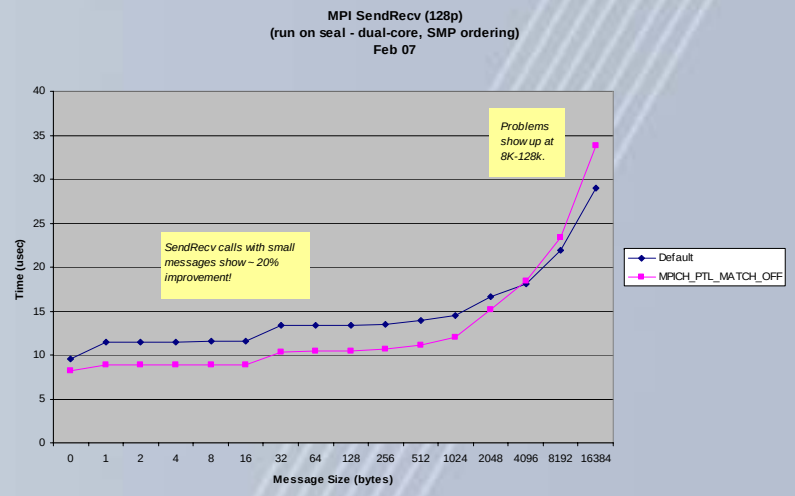
- See over 30% performance improvement with 256 PEs for MPI_Barrier in Pallas benchmark

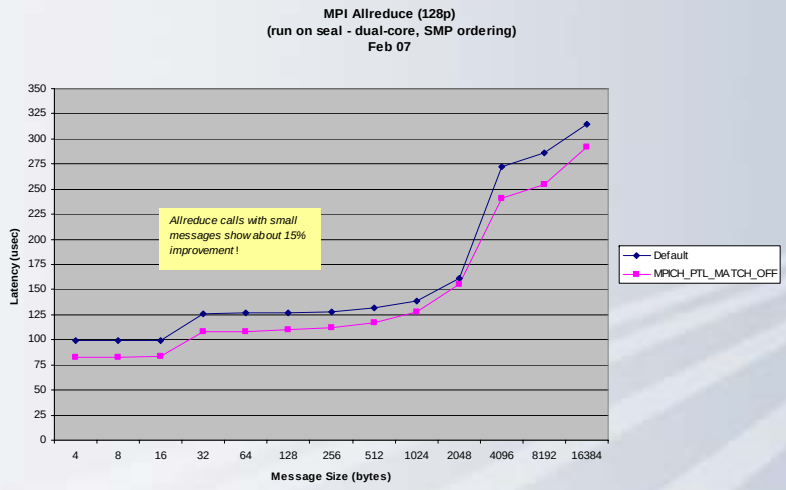
Memcpy

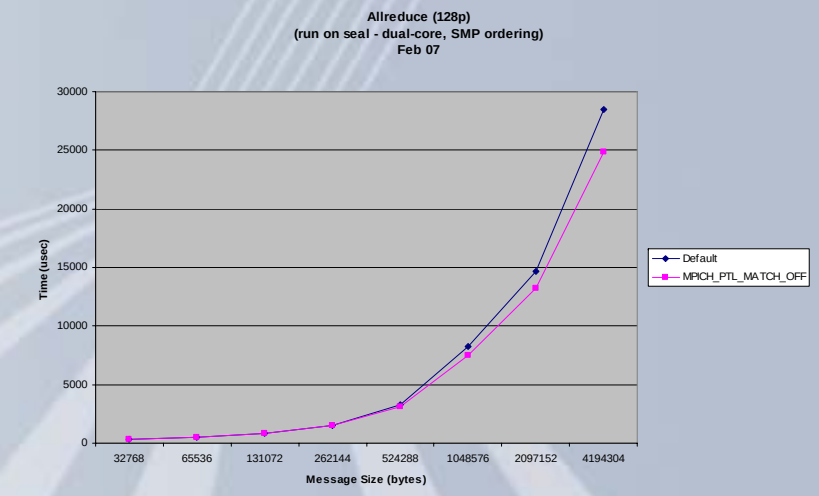

Default Memcpy vs Optimized Memcpy Speeds
perch - Catamount 12/1/06
MB/sec
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
8388608
Transfer size (bytes)
Optimized Memcpy
Default Memcpy

Portals Match Off



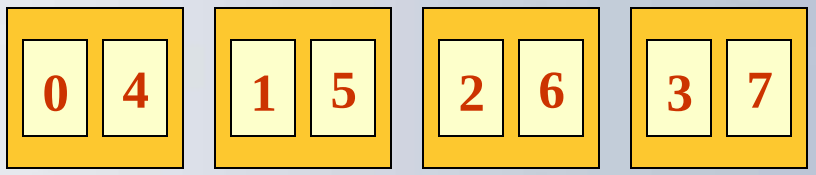






MPI Rank Reorder

- MPI rank placement with environment variable

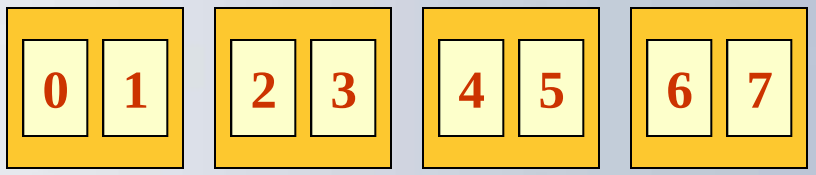


  - Distributed placement

- Performance improvements depend on application
  - DLPOLY – SMP style – 13% gain (512pes)
  - HPCC(PTRANS) – custom placement -> 2x gain (10000pes)

MPI Rank Reorder

- MPI rank placement with environment variable

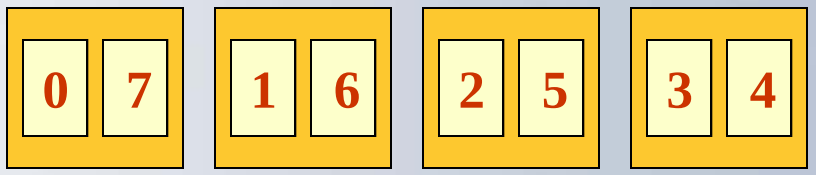


  - Distributed placement
  - SMP style placement

- Performance improvements depend on application
  - DLPOLY – SMP style – 13% gain (512pes)
  - HPCC(PTRANS) – custom placement -> 2x gain (10000pes)

MPI Rank Reorder

- MPI rank placement with environment variable

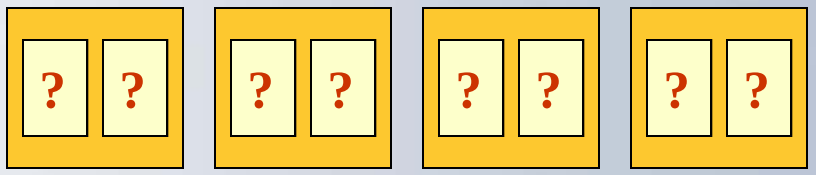


  - Distributed placement
  - SMP style placement
  - Folded rank placement

- Performance improvements depend on application
  - DLPOLY – SMP style – 13% gain (512pes)
  - HPCC(PTRANS) – custom placement -> 2x gain (10000pes)

MPI Rank Reorder

- MPI rank placement with environment variable



  - Distributed placement
  - SMP style placement
  - Folded rank placement
  - User provided rank file

- Performance improvements depend on application
  - DLPOLY – SMP style – 13% gain (512pes)
  - HPCC(PTRANS) – custom placement -> 2x gain (10000pes)

Rank Reordering

- Rank ordering can increase MPI performance significantly on point-to-point transfers, since it brings the rank neighbors closer together and reduces network congestion
- Likewise we see a substantial boost in performance for some commonly used collectives like Allreduce, Alltoall, Bcast and Reduce
  - For various message sizes, we see a 10% - 45% improvement using SMP ordering compared to the distributed rank ordering
- Benchmark Gains using Various Rank Re-Ordering Methods
  - DLPOLY – SMP style –> 13% improvement (512 PEs)
    - performance boost is likely due to the overall reduction in network congestion, and the improvement in Allreduce
  - CPMD – Custom placement –> 10% improvement (6676 PEs)
    - likely due to reduced network congestion and the improvement seen with Alltoall
  - HPCC(PTRANS) – custom placement -> 2.5x improvement (10000pes)
    - due to each PTRANS rank communicating with only one other PTRANS rank. When these communicating ranks are placed on the same node, the benchmark bandwidth is significantly increased
- Rank reordering can also improve performance on imbalanced applications

Rank Reorder Example - hycom

```
pat_report -O load_balance

Table 2:  Load Balance across PE's by FunctionGroup

  Time % |    Cum. |        Time |       Calls |Group
         | Time % |             |             | PE[mmm]

  100.0% | 100.0% | 482.705844 | 7446623155 |Total
|-----------------------------------------------------
|  57.7% |  57.7% | 278.657370 | 7329740077 |USER
||----------------------------------------------------
||   0.5% |   0.5% | 361.310805 |   33130409 |pe.201
||   0.4% |  58.2% | 311.898417 |   34020074 |pe.45
||   0.0% | 100.0% |  23.780267 |     320096 |pe.184
||====================================================
|  42.3% | 100.0% | 204.048383 |  116783478 |MPI
||----------------------------------------------------
||   0.9% |   0.9% | 476.662251 |     399087 |pe.184
||   0.3% |  61.9% | 167.921814 |     422197 |pe.37
||   0.2% | 100.0% | 119.123503 |     514637 |pe.201
|=====================================================
```

Rank Reorder Example - hycom

```
pat_report -O load_balance -s pe=ALL

Table 2:  Load Balance across PE's by FunctionGroup

  Time % |    Cum. |        Time |       Calls |Group
         | Time % |             |             | PE

  100.0% | 100.0% | 482.705844 | 7446623155 |Total
|-------------------------------------------------
|  57.7% |  57.7% | 278.657370 | 7329740077 |USER
||------------------------------------------------
||   0.5% |   0.5% | 361.310805 |   33130409 |pe.201
||   0.5% |   1.0% | 349.849557 |   30460022 |pe.182
||   0.5% |   1.5% | 346.919713 |   33685730 |pe.200
||   0.5% |   2.0% | 342.844256 |   34879988 |pe.188
||   0.5% |   2.5% | 342.308415 |   34913960 |pe.172
. . .
||   0.1% |  99.8% |  45.464691 |    3000260 |pe.248
||   0.1% |  99.9% |  35.970972 |     399401 |pe.213
||   0.0% |  99.9% |  27.431543 |     340673 |pe.232
||   0.0% | 100.0% |  25.142167 |     117620 |pe.240
||   0.0% | 100.0% |  23.780267 |     320096 |pe.184
||================================================
```

Rank Reorder Example - hycom

After custom placement (10% performance improvement):

```
Table 2:  Load Balance with MPI Sent Message Stats

  Time % |       Time | Sent Msg |     Sent Msg | Avg Sent |Group
         |            |    Count |  Total Bytes | Msg Size | PE[mmm]

  100.0% | 437.418783 | 17161829 | 289328285840 | 16858.83 |Total
|-----------------------------------------------------------------
|  60.2% | 263.211966 |       -- |           -- |       -- |USER
||----------------------------------------------------------------
||   0.5% | 322.019049 |       -- |           -- |       -- |pe.158
||   0.4% | 286.179471 |       -- |           -- |       -- |pe.126
||   0.0% |  23.318648 |       -- |           -- |       -- |pe.184
||================================================================
|  39.8% | 174.206510 | 17161829 | 289328285840 | 16858.83 |MPI
||----------------------------------------------------------------
||   1.0% | 414.091071 |    62224 |    635942368 | 10220.21 |pe.184
||   0.3% | 151.242560 |    68002 |   1039329136 | 15283.80 |pe.126
||   0.3% | 115.396258 |    68002 |   1039329136 | 15283.80 |pe.158
|=================================================================
```

Performance Measurement and Visualization on the Cray XT4 MPI Recent Work

Questions / Comments

Thank You!

NERSC
September 18, 2007