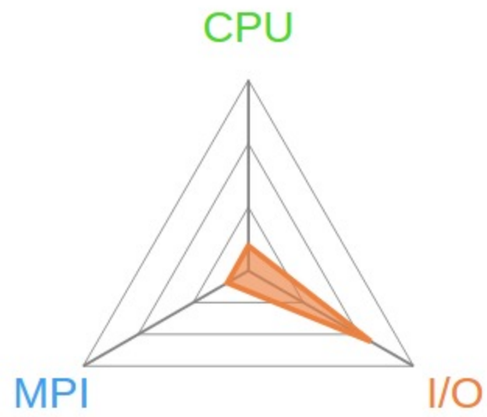
Executable:	MADbench2
Resources:	4 processes, 1 node
Machine:	Namaste-II
Start time:	Mon Nov 4 13:40:21 2013
Total time:	14 seconds (0 minutes)
Full path:	/home/mark/Work/MADbench2
Notes:	4-core laptop / SSD / 4 readers + writers



Summary: MADbench2 is I/O-bound in this configuration

The total wallclock time was spent as follows:

CPU	13.1%	Time spent running application code. High values are usually good. This is **low**; it may be worth improving I/O performance first.
MPI	12.7%	Time spent in MPI calls. High values are usually bad. This is **low**; this code may benefit from increasing the process count.
I/O	74.2%	Time spent in filesystem I/O. High values are usually bad. This is **very high**; check the I/O breakdown section for optimization advice.

This application run was I/O-bound. A breakdown of this time and advice for investigating further is in the I/O section below.

As very little time is spent in MPI calls, this code may also benefit from running at larger scales.

CPU

A breakdown of how the 13.1% total CPU time was spent:

Scalar numeric ops	8.8%
Vector numeric ops	0.0%
Memory accesses	91.2%
Other	0.0

The per-core performance is memory-bound. Use a profiler to identify time-consuming loops and check their cache performance.

No time was spent in vectorized instructions. Check the compiler's vectorization advice to see why key loops could not be vectorized.

MPI

Of the 12.7% total time spent in MPI calls:

Time in collective calls	100.0%
Time in point-to-point calls	0.0%
Estimated collective rate	33.7 bytes/s
Estimated point-to-point rate	0 bytes/s

All of the time is spent in collective calls with a very low transfer rate. This suggests a significant load imbalance is causing synchronization overhead. You can investigate this further with an MPI profiler.

I/O

A breakdown of how the 74.2% total I/O time was spent:

Time in reads	10.8%
Time in writes	89.2%
Estimated read rate	737 Mb/s
Estimated write rate	88.2 Mb/s

Most of the time is spent in write operations, which have a low transfer rate. This may be caused by contention for the filesystem or inefficient access patterns. Use an I/O profiler to investigate which write calls are affected.

Memory

Per-process memory usage may also affect scaling:

Mean process memory usage	486 Mb
Peak process memory usage	543 Mb
Peak node memory usage	41.2%

The peak node memory usage is average. You may be able to reduce the total number of CPU hours used by running with fewer MPI processes and more data on each process.