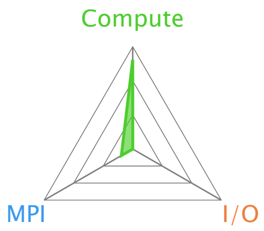
arm PERFORMANCE REPORTS

Command:	mpirun -np 5 ../../dspec finite_beta_V5_vmec1
Resources:	1 node (16 physical, 32 logical cores per node)
Memory:	31 GiB per node
Tasks:	5 processes
Machine:	r496
Start time:	Fri Dec 7 2018 04:40:47 (UTC+11)
Total time:	8434 seconds (about 141 minutes)
Full path:	/short/y08/zq1102/SPEC



Summary: dspec is Compute-bound in this configuration

Compute	87.1%	Time spent running application code. High values are usually good. This is **high**; check the CPU performance section for advice
MPI	12.8%	Time spent in MPI calls. High values are usually bad. This is **very low**; this code may benefit from a higher process count
I/O	<0.1%	Time spent in filesystem I/O. High values are usually bad. This is **very low**; however single-process I/O may cause MPI wait times

This application run was Compute-bound. A breakdown of this time and advice for investigating further is in the CPU 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 the 87.1% CPU time:

Scalar numeric ops	26.8%
Vector numeric ops	1.0%
Memory accesses	72.1%

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

Little time is spent in vectorized instructions. Check the compiler's vectorization advice to see why key loops could not be vectorized.

MPI

A breakdown of the 12.8% MPI time:

Time in collective calls	100.0%
Time in point-to-point calls	0.0%
Effective process collective rate	100 kB/s
Effective process point-to-point rate	0.00 bytes/s

I/O

A breakdown of the <0.1% I/O time:

Time in reads	0.0%
Time in writes	100.0%
Effective process read rate	0.00 bytes/s
Effective process write rate	1.28 MB/s

Most of the time is spent in write operations with a very low effective 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.

Threads

A breakdown of how multiple threads were used:

Computation	0.0%
Synchronization	0.0%
Physical core utilization	31.2%
System load	125.2%

No measurable time is spent in multithreaded code.

The system load is high - multiple processes may be sharing one core.

Memory

Per-process memory usage may also affect scaling:

Mean process memory usage	651 MiB
Peak process memory usage	852 MiB
Peak node memory usage	48.0%

The peak node memory usage is low. Running with fewer MPI processes and more data on each process may be more efficient.

Energy

A breakdown of how energy was used:

CPU	not supported %
System	not supported %
Mean node power	not supported W
Peak node power	0.00 W

Energy metrics are not available on this system.

CPU metrics are not supported (no intel_rapl module)