Memory Use on ARCHER

Analysis of memory use on the UK national supercomputing service

Andy Turner a.turner@epcc.ed.ac.uk
Simon McIntosh-Smith S.McIntosh-Smith@bristol.ac.uk

Background

Why do we care about memory use?

Two main reasons:

1. **Procurement**

   Memory cost is now a major factor in large-scale procurements. A factor of 2 difference in memory per node has a big effect.

2. **Future architectures**

   Small-but-fast high bandwidth memories such as HBM and MCDRAM will become increasingly common over the next 3 years, even on CPUs. Will we be able to exploit these for mainstream HPC systems?

What did we do?

We set out to collect a rigorous dataset from a large-scale, mainstream HPC service (ARCHER), over a substantial timeframe (~1 year).

To be rigorous, we collected data on real application memory use during production runs.

Our goal was to produce a contemporary dataset on application memory use that could be useful for anyone running procurements or considering future designs which rely on HBM-like technologies.

Data collection

archer

epcc

EPSRC
Engineering and Physical Sciences Research Council

NERC
SCIENCE OF THE ENVIRONMENT

CRAY
THE SUPERCOMPUTER COMPANY

Data collection and munging

- We used Cray Resource Usage Reporting (RUR)
  - Gathers data for every parallel application run on the system
- Needed to link this data to other streams to get useful information
  - Identify application used
  - Project and user information
  - Job sizes, job IDs
- Used EPCC's SAFE system, which links multiple data streams
  - Allows us to query memory usage by application, project, user, job size, etc.

Results

Caveats

- We assume homogeneous memory usage per process
  - This is a reasonable assumption for most use cases on ARCHER
  - Can overestimate memory requirements
  - We've seen at least one case where this assumption is false
- Initial pass through the data
  - Much more analysis to be done

Overall memory use

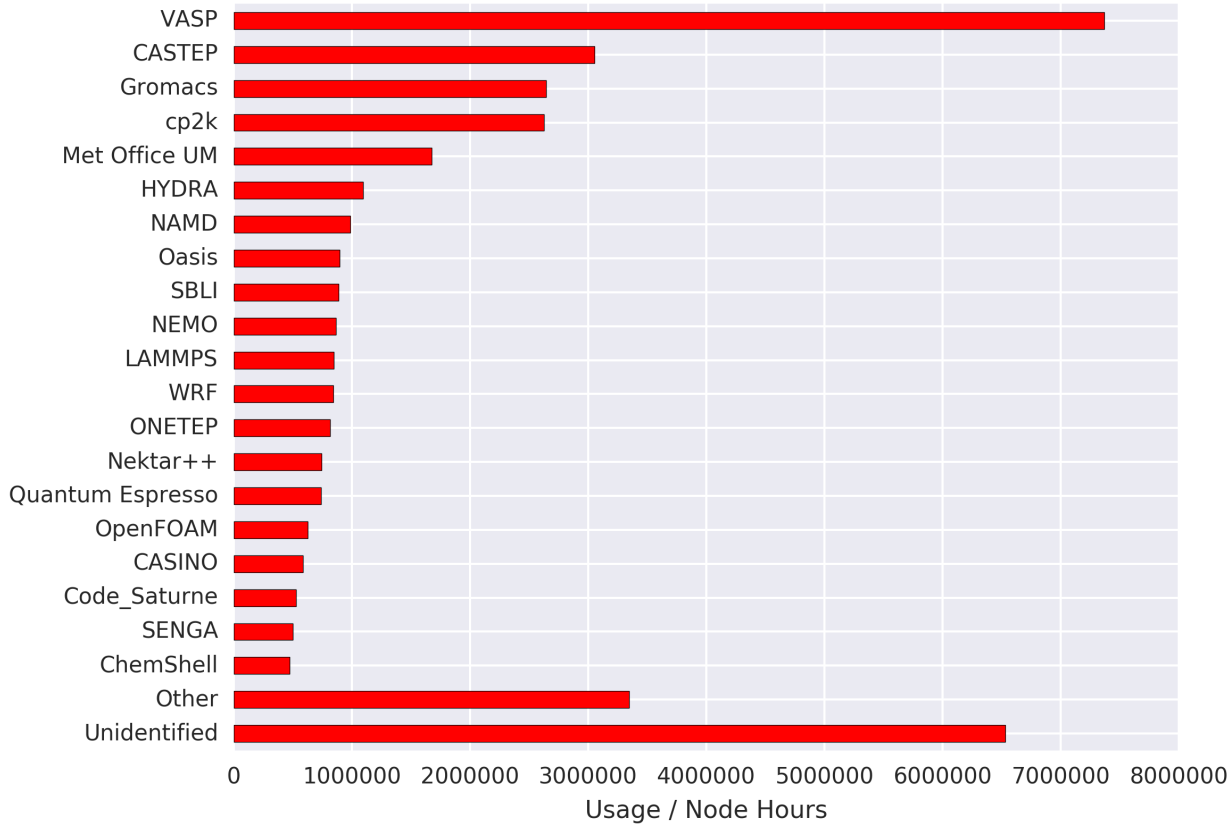
24 cores per node

% use by max. memory

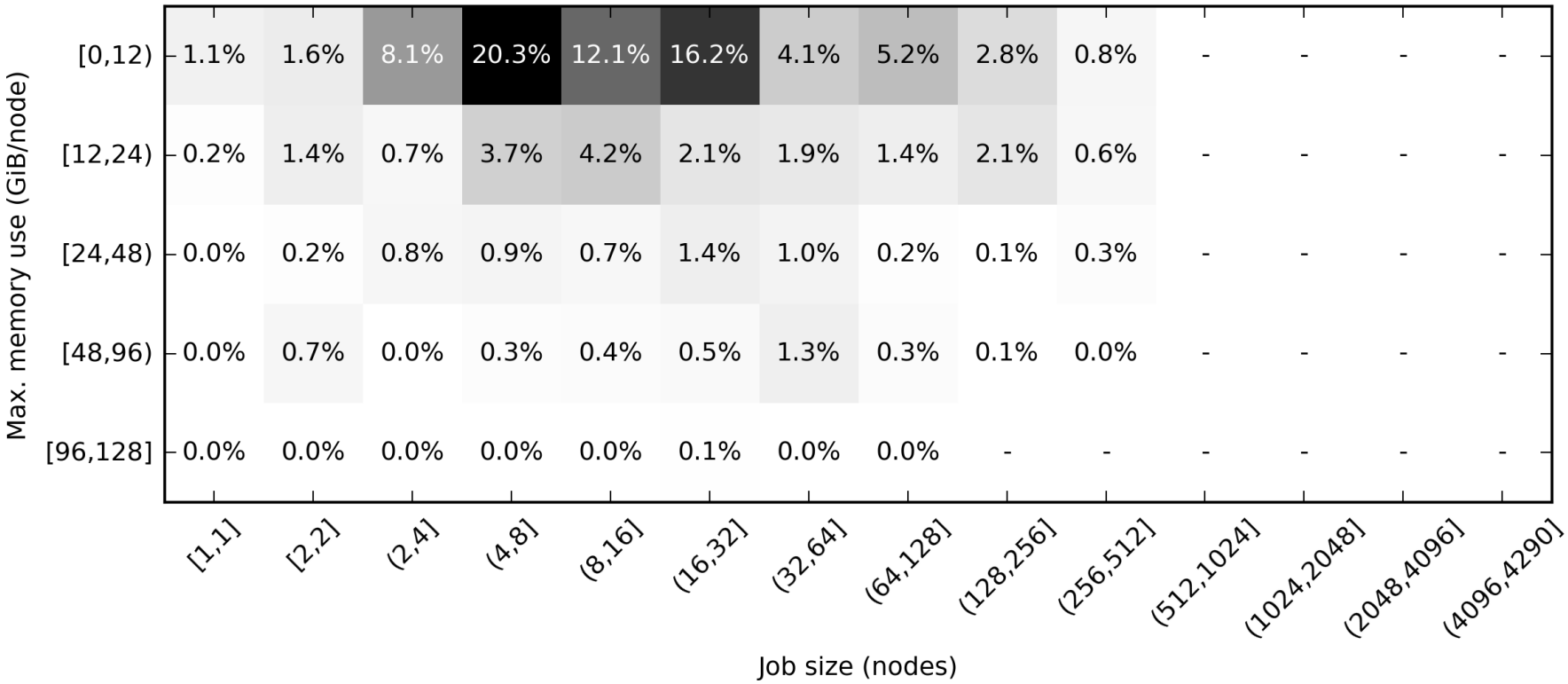
Max. Memory Use (GiB/node)	Small (<= 32 nodes)	Large (> 32 nodes)	All	All (Cumulative)
(0,12)	68.6%	55.4%	65.4%	65.4%
[12,24)	20.0%	25.7%	21.4%	86.8%
[24,48)	8.5%	12.1%	9.4%	96.2%
[48,96)	2.7%	6.7%	3.7%	99.9%
[96,128)	0.1%	0.1%	0.1%	100.0%

24 cores per node

ARCHER Application Use



VASP – Periodic Electronic Structure



24 cores per node

GROMACS – *N*-body Modelling

24 cores per node

Met Office UM – Climate Modelling

24 cores per node

Conclusions & next steps

- > 80% of use on ARCHER uses less than 24 GiB/node (1 GiB/core)
  - > 60% uses less than 12 GiB/node (0.5 GiB/core)
- Memory requirements often increase as job size increases
- Opportunity to exploit limited-capacity, high-bandwidth memory
  - Many applications are memory-bandwidth bound
  - Performance needs to be available without code modification

- Improve analysis to catch non-homogeneous memory use
- Work with other centres to understand differences/similarities
- Work with user communities to improve understanding of application memory use

Questions?

Andy Turner a.turner@epcc.ed.ac.uk
Simon McIntosh-Smith S.McIntosh-Smith@bristol.ac.uk