CATALYST UK AT EPCC

Prof Mark Parsons

EPCC Director
Associate Dean for e-Research

THE UNIVERSITY of EDINBURGH

epcc

The CATALYST UK programme

- Collaboration between
  - Hewlett Packard Enterprise, Arm & SUSE
  - EPCC, The University of **Edinburgh**
  - University of **Bristol**
  - University of **Leicester**
- Deploy Arm-based HPC systems across the 3 academic sites in the UK
- Programme running for 3 years from December 2018

The aim of the programme

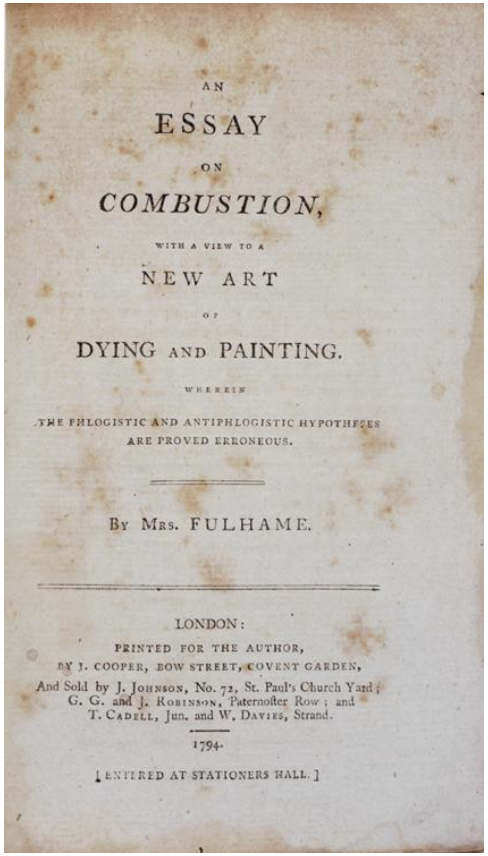
- To investigate and showcase the potential of Arm-based HPC systems
- To further drive supercomputer adoption in the UK in general, and in the commercial sector in particular
- To cooperate with the UK industry to jointly develop critical applications and workflows to best exploit the Arm system capabilities
- To provide training for researchers to successfully and productively work with Arm-based systems in the future

The systems

- HPE Apollo 70
- The clusters at each university are largely identical
  - Designed, built and supported by HPE
- 64 nodes (i.e. 4096 cores) per cluster
  - 2 racks
  - Dual-socket nodes with Cavium ThunderX2 CPUs (32 cores per CPU)
  - 128GB (16 x 8GB) DDR4 per node
  - Mellanox EDR InfiniBand

EPCC's system

- Fulhame was delivered and installed at the end of Nov'18
  - Named after Elizabeth Fulhame - 18th century Scottish chemist, first to describe the process of catalysis
- It was used for a PRACE Advanced Training Centre course on the 3rd & 4th December
  - **Less than 1 week after delivery!**
- Since then, work has focused on code porting & benchmarking


AN
ESSAY
ON
COMBUSTION,
WITH A VIEW TO A
NEW ART
OF
DYING AND PAINTING.
WHEREIN
THE PHLOGISTIC AND ANTIPHLOGISTIC HYPOTHESES
ARE PROVED ERRONEOUS.
By Mrs. FULHAME.
LONDON:
PRINTED FOR THE AUTHOR,
BY J. COOPER, BOW STREET, COVENT GARDEN,
And Sold by J. Johnson, No. 72, St. Paul's Church Yard;
G. G. and J. Robinson, Paternoster Row; and
T. Cadell, Jun. and W. Davies, Strand.
1794.
[ENTERED AT STATIONERS HALL.]

Porting & benchmarking

- Port most heavily used ARCHER and Cirrus packages
  - Make them available as modules
  - Built with different compilers, libraries, etc
- No specific performance optimisation in the first instance
  - Focus on making as many applications available as possible

Rank	Code Name	Usage / Nh	% Usage	Jobs	% Jobs	Users	Mean Job Size / nodes	Median Job Size / nodes
1	VASP	451961	16.73	9055	20.58	87	43.27	17
2	Gromacs	188732	6.99	1976	4.49	31	20.93	21
3	cp2k	155805	5.77	3762	8.55	39	38.64	22
4	CASTEP	139383	5.16	6177	14.04	31	168.06	16
5	Quantum Espresso	116045	4.30	614	1.40	10	95.66	68
6	NEMO	107968	4.00	1256	2.86	12	63.95	25
7	iIMB	102601	3.80	350	0.80	3	41.50	27
8	LAMMPS	79910	2.96	2857	6.49	26	9.31	5
9	ONETEP	65543	2.43	68	0.15	5	52.04	60
10	MITgcm	44673	1.65	761	1.73	7	15.03	12
11	WRF	44028	1.63	99	0.23	7	42.14	4
12	EPOCH	40176	1.49	85	0.19	4	181.53	80
13	Oasis	38625	1.43	231	0.53	6	19.21	17
14	NAMD	37336	1.38	178	0.40	6	18.17	21
15	Met Office UM	36773	1.36	354	0.80	9	11.37	6
16	SBLI	36258	1.34	74	0.17	3	130.63	32
17	OpenFOAM	31593	1.17	231	0.53	19	29.62	20
18	Code_Saturne	23527	0.87	201	0.46	5	16.51	11
19	SENGA	22347	0.83	54	0.12	4	53.85	9
20	CASINO	18572	0.69	80	0.18	1	13.67	8
21	CRYSTAL	17013	0.63	386	0.88	8	11.98	9
22	Nektar++	15614	0.58	19	0.04	3	98.15	50
23	FHI aims	14238	0.53	157	0.36	1	40.97	20
24	ChemShell	9855	0.36	170	0.39	6	11.45	8
25	JOREK	8946	0.33	11	0.03	1	249.04	210
26	CESM	6636	0.25	92	0.21	4	5.70	3
27	OSIRIS	5357	0.20	16	0.04	5	413.89	324
28	HYDRA	4713	0.17	61	0.14	2	12.13	2
29	Nek5000	3520	0.13	6	0.01	1	80.00	80
30	GS2	3042	0.11	18	0.04	2	35.03	22

```
mwngio@eslogin005:~> module avail gromacs

---------------------- /opt/modules/packages-archer -----------------------
gromacs/4.6.5                   gromacs/5.1.2-plumed2.3b
gromacs/4.6.5-plumed            gromacs/5.1.4
gromacs/4.6.7-plumed            gromacs/5.1.4-plumed2.3.3
gromacs/5.0.2                   gromacs/5.1.4-plumed2.4.1-adjmat
gromacs/5.1                     gromacs/2016.3(default)
gromacs/5.1-plumed              gromacs/2016.4
gromacs/5.1.1-plumed            gromacs/2018.2
gromacs/5.1.2                   gromacs/2018.2-plumed2.4.2
mwngio@eslogin005:~>
```

```
[mweiland@cirrus-login0 ~]$ module avail netcdf

--------------------------- /lustre/sw/modulefiles ---------------------------
netcdf/4.4.1                        netcdf-parallel/4.5.0-intel17
netcdf-parallel/4.5.0               netcdf-parallel/4.5.0-intel17-mpt214
netcdf-parallel/4.5.0-gcc6-mpt214
[mweiland@cirrus-login0 ~]$
```

epcc

HPCG - High Performance Conjugate Gradient

- Kernel benchmark aiming to exercise
  - Floating point performance
  - Memory bandwidth
  - Network bandwidth and latency
  - Implemented with C++, MPI & OpenMP
  - Performance measured in Gflop/s

Single node

System	Performance
HPE Apollo 70	30.529
SGI ICE XA	21.115
Cray XC30	15.650
Dell EMC (AVX512)	34.581
Dell EMC (AVX2)	28.120

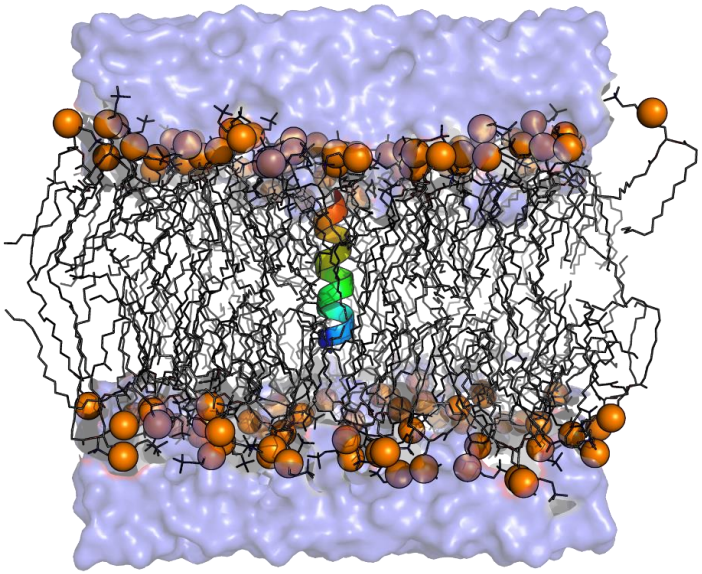
Multi-node

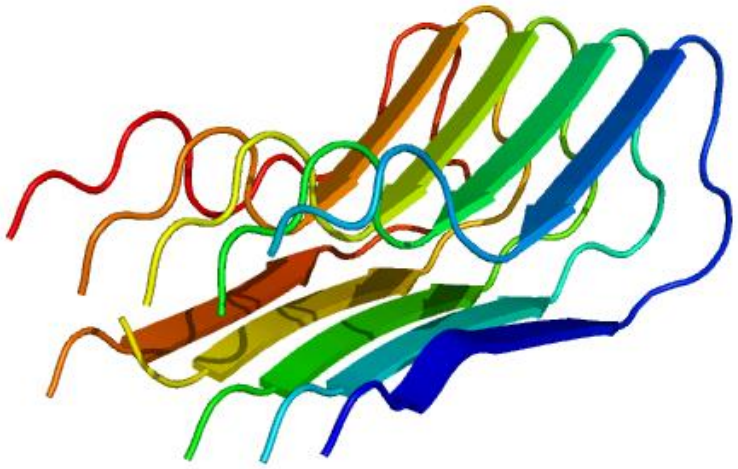
Nodes	HPE Apollo 70	SGI ICE XA	Cray XC30	Dell EMC (AVX2)
2	61.185	40.491	32.048	56.505
4	120.972	80.317	61.617	111.598
8	241.363	158.774	120.749	214.358
16	482.202	305.534	245.870	400.320
32	939.109	602.092	491.101	747.066

epcc

GROMACS

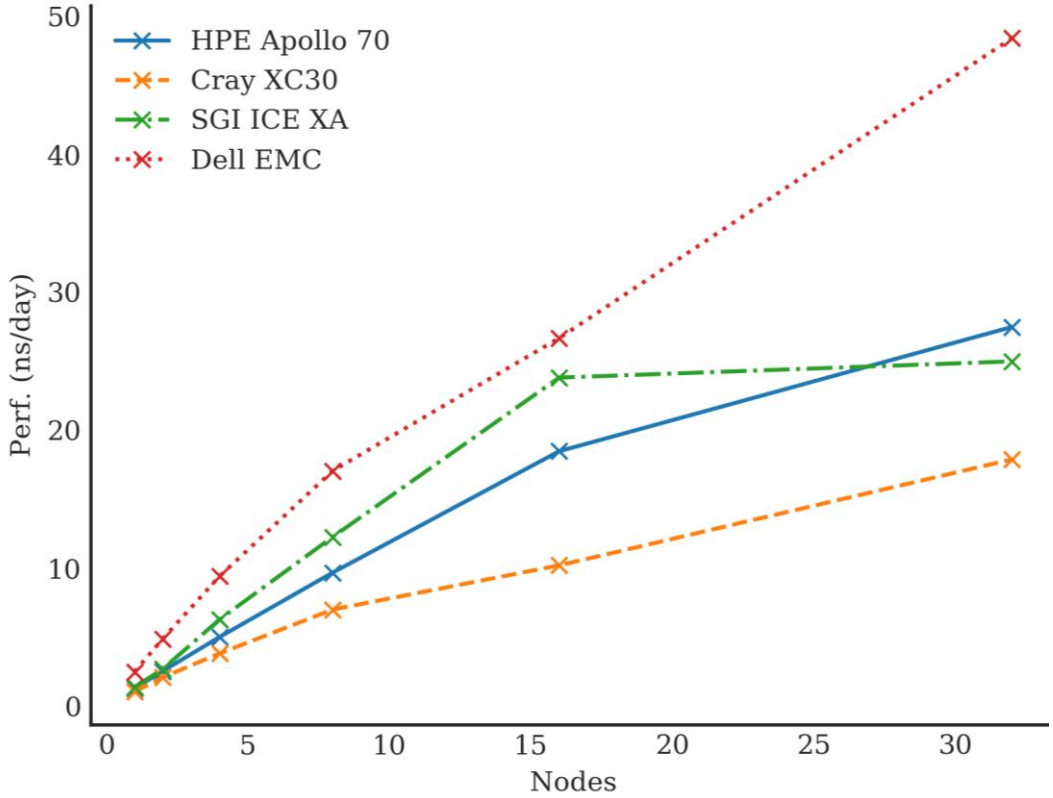
- Molecular dynamics simulations
- Primarily designed for biochemical molecules but also used on systems like polymers
- Implemented in C with MPI & OpenMP





GROMACS 1400k atom benchmark

GROMACS			
HPE Apollo 70	Arm Clang 19.0.0	-std=c++11 -mcpu=native -O3 -DNDEBUG -funroll-all-loops -fexcess-precision=fast -fPIC -O3	OpenMPI 4.0.0 FFTW 3.3.8
SGI ICE XA	GCC 6.2.0	-march=core-avx2 -std=c++0x -O3 -funroll-all-loops -fexcess-precision=fast	SGI MPT 2.16 FFTW 3.3.5
Cray XC30	GCC 6.3.0	-mavx -static -O3 -ftree-vectorize -funroll-loops -std=c++11 -O3 -funroll-all-loops -fexcess-precision=fast	Cray MPICH2 7.5.5 FFTW 3.3.6
Dell EMC	Intel 17.4	-xCORE-AVX512 -mkl=sequential -std=gnu99 -std=c++11 -O3 -ip -funroll-all-loops -alias-const -ansi-alias -no-prec-div -fimf-domain-exclusion=14 -qoverride-limits	Intel MPI library 17.4 Intel MKL 17.4



Detailed results

ACM DL DIGITAL LIBRARY

Evaluating the Arm Ecosystem for High Performance Computing

Full Text: PDF

2019 Article
Research
Refereed limited

Authors:
Adrian Jackson EPCC, The University of Edinburgh, Edinburgh, United Kingdom
Andrew Turner EPCC, The University of Edinburgh, Edinburgh, United Kingdom
Michèle Weiland EPCC, The University of Edinburgh, Edinburgh, United Kingdom
Nick Johnson EPCC, The University of Edinburgh, Edinburgh, United Kingdom
Olly Perks Arm, Warwick, United Kingdom
Mark Parsons EPCC, The University of Edinburgh, Edinburgh, United Kingdom

Bibliometrics

· Citation Count: 0
· Downloads (cumulative): 17
· Downloads (12 Months): 17
· Downloads (6 Weeks): 17

Published in:

· Proceeding
PASC '19 Proceedings of the Platform for Advanced Scientific Computing Conference
Article No. 12

Zurich, Switzerland — June 12 - 14, 2019
ACM New York, NY, USA ©2019
table of contents ISBN: 978-1-4503-6770-7 doi>10.1145/3324989.3325722

epcc

Summary

- In general porting was very straightforward
  - GNU compilers and Arm performance library help significantly
  - Arm compilers can bring performance benefits
- Performance is generally good
  - Not necessarily better than top end Intel
  - But reasonably comparable and potentially much cheaper
- Memory bandwidth dominated codes benefit significantly
- The Catalyst UK programme will give the Arm HPC-ecosystem a significant boost by creating a mature and stable environment for high-performance scientific and industry applications