Scientific Application Performance on HPC, Private and Public Cloud Resources: A Case Study Using Climate, Cardiac Model Codes and the NPB Benchmark Suite

Peter Strazdins
(Research School of Computer Science),
Jie Cai, Muhammad Atif and Joseph Antony
(National Computational Infrastructure),
The Australian National University

Workshop on Parallel and Distributed Scientific and Engineering Computing, Shanghai, 25 May 2012

(slides available from http://cs.anu.edu.au/~Peter.Strazdins/seminars)

THE AUSTRALIAN NATIONAL UNIVERSITY

1 Overview

- motivating scenario
- experimental setup
  - system (private and public cloud),
  - software
  - applications: Chaste cardiac and MetUM atmosphere simulations
- results
  - OSU communication microbenchmarks
  - NAS Parallel Benchmarks
  - applications
- conclusions and future work

2 Motivations: a Cloud-bursting Supercomputer Facility

- supercomputing facilities provide access to state-of-the-art cluster computers
  - also provide comprehensive software stacks to support a diverse range of applications
- the supercomputing cluster is typically highly contended resource
  - users may be restricted to limited resources
  - may have long turnaround times
  - some workloads may not make good use of cluster
- $\Rightarrow$ may be better off using a private or even public cloud
  - requires easily replication of software stack on cloud resources
  - ideally, migration of jobs onto cloud would be transparent
  - recent frameworks can transparently profile HPC jobs for cloud suitability, e.g. ARRIVE-F, (and migrate VMs accordingly)

3 Experimental Setup: Systems

Platform		private cloud	public cloud	premiere cluster
Platform		DCC	EC2	Vayu
# of Nodes		8	4	1492
CPU	Model	Intel Xeon E5520	Intel Xeon X5570	Intel Xeon X5570
	Clock Spd	2.27GHz	2.93GHz	2.93GHz
	#Cores	8 (2 slots)	8×2	8 (2 slots)
	L2 Cache	8MB (shared)	8MB (shared)	8MB (shared)
Memory per node		40GB	20GB	24GB
Operating System		Centos 5.7	CentOS 5.7	CentOS 5.7
Virtualization		VMware ESX 4.0	Xen	–
File System		NFS	NFS	Lustre
Interconnect		1 GigE (dual)	10 GigE	QDR IB

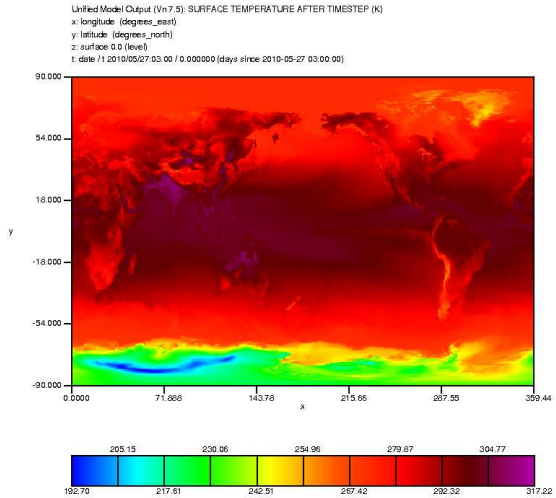
- DCC: 1 VM/node; filesystems mounted via external cluster via two QLogic channel fibre HBAs
- vayu: QDR IB used for both compute and storage

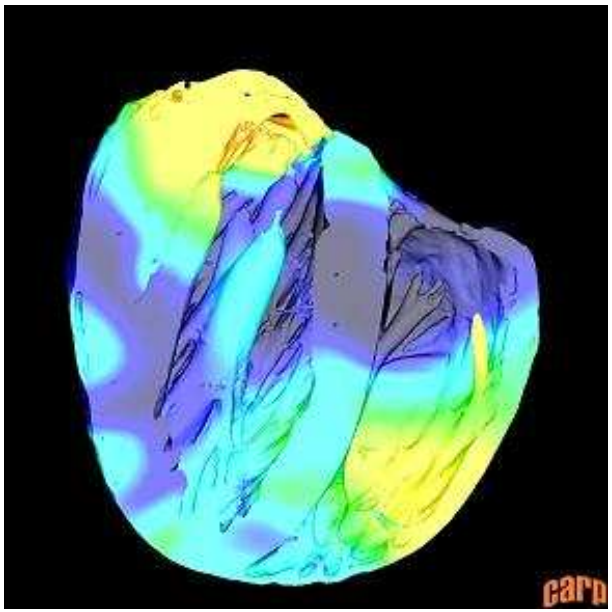
4 Experimental Setup: Software

- EC2: StarCluster instance to automate the build, configuration & management of HPC compute nodes
- vayu `/apps` directory: system-wide compilers, libraries, and application codes
  - user environment is configured via `module` package
- rsync `/apps` and user home/project directories onto the VM to replicate stack
  - minimizes interference of existing stack on the clouds
  - only occasionally needed to recompile for the clouds
- benchmarking software
  - OSU MPI communication micro-benchmarks: bandwidth and latency
  - NAS Parallel Benchmark MPI suite 3.3, class B
    - 5 kernels & 3 pseudo-applications derived from CFD applications

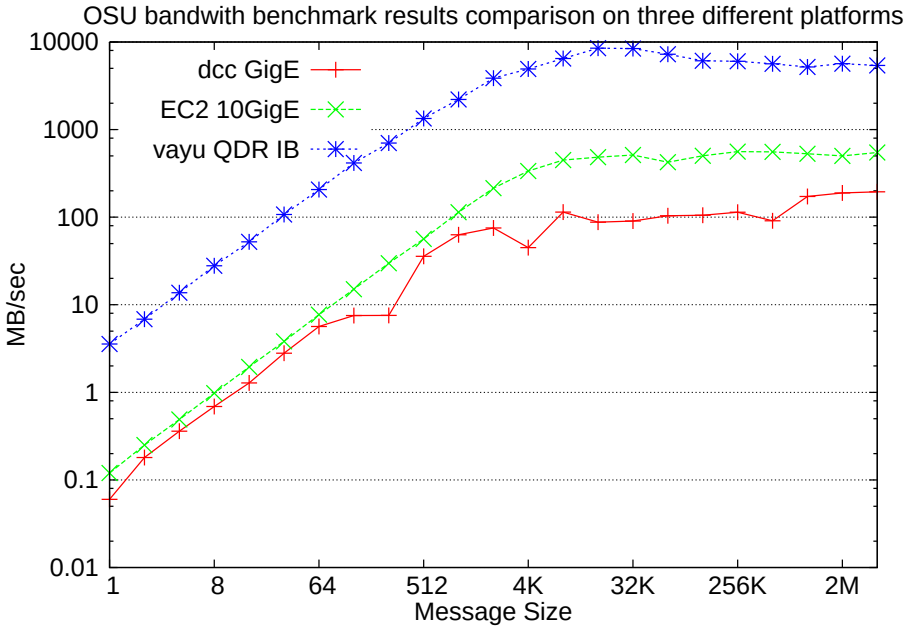
5 Experimental Setup: Applications

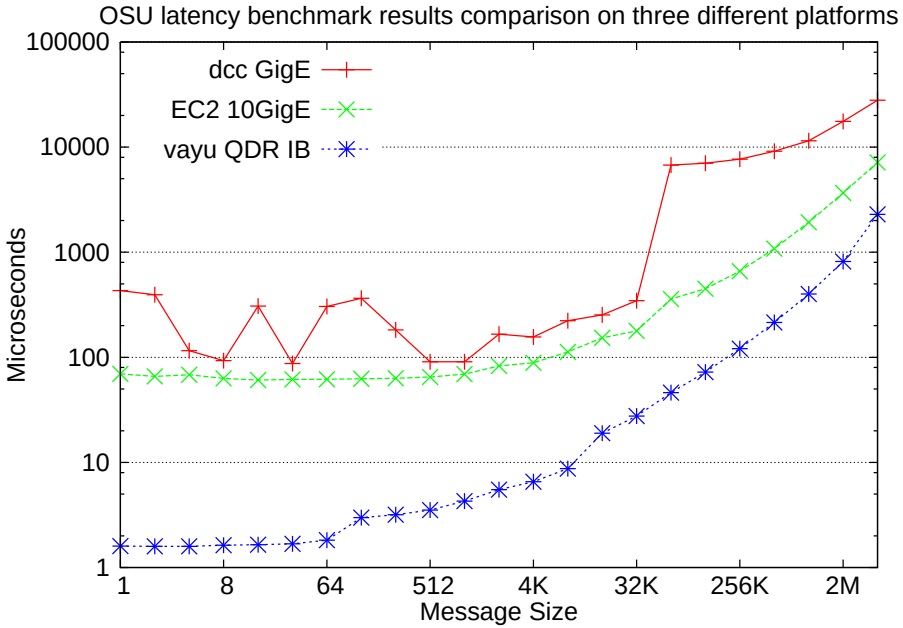
- UK Met Office Unified Model (MetUM) version 7.8
  - used for operational weather forecasts in UK, Australia, S. Korea, ...
  - benchmark: global atmosphere model using a $640 \times 481 \times 70$ grid
- Chaste version 2.1 cardiac simulation
  - model electric field propagations and ion transport
  - benchmark: 4M node, 24M element rabbit heart mesh
  - more memory-intensive than the UM benchmark!
- dominant part of both is a linear system solve on each timestep





6 Results: Communication Micro-benchmarks



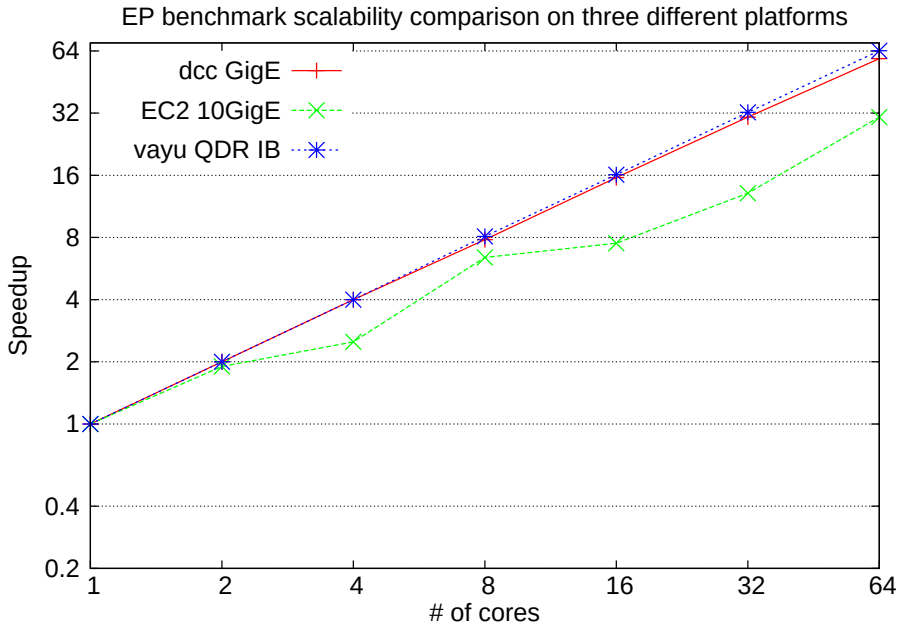


OSU MPI bandwidth tests (MB/s vs message size)

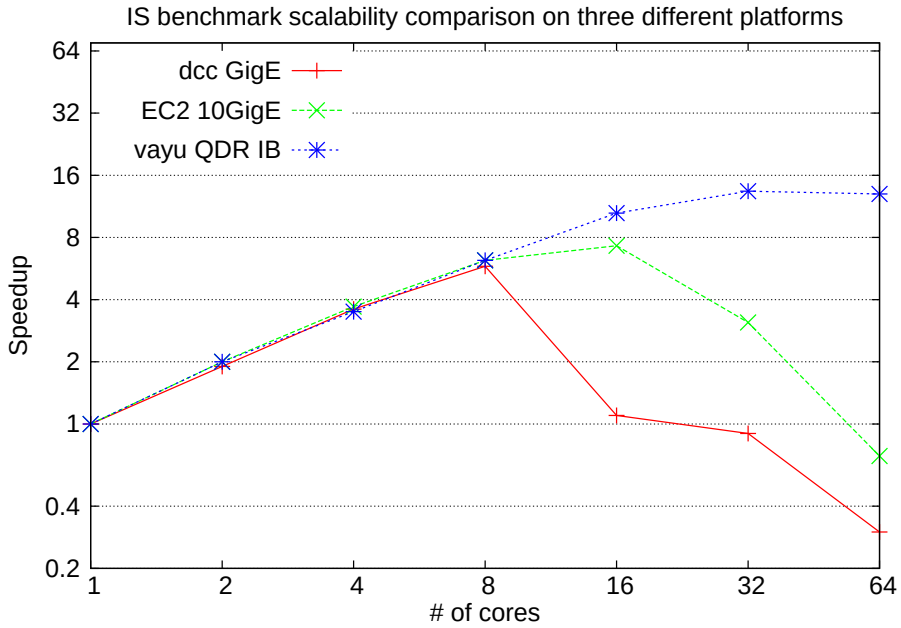
OSU MPI latency tests (time (μs) vs message size)

- trends as expected per theoretical specifications: more than one order of magnitude better performance on vayu
- fluctuations on DCC suspected from CPU scheduling by hypervisor

7 Results: NAS Parallel Benchmarks (I)



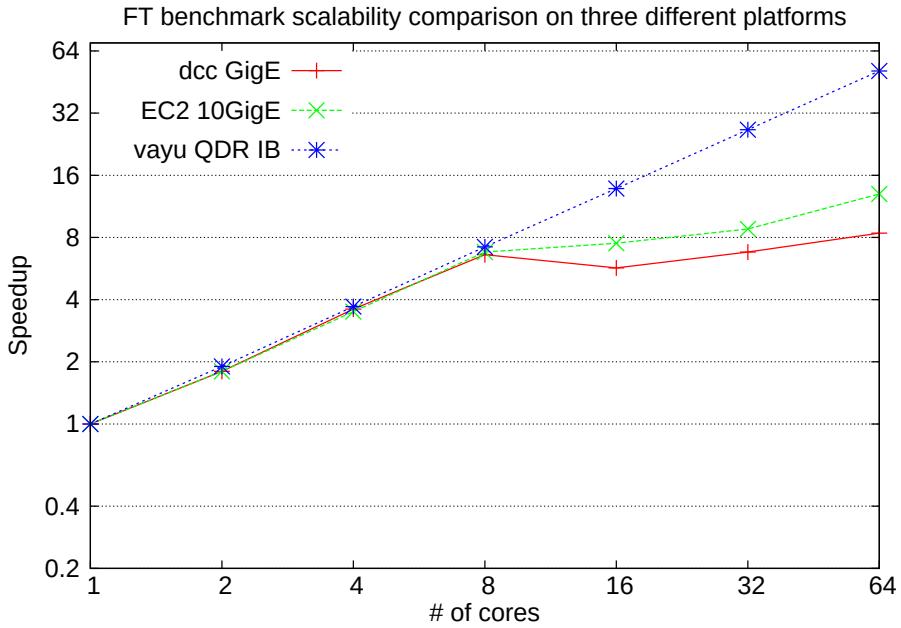
EP.B speedup, 1 to 64 cores



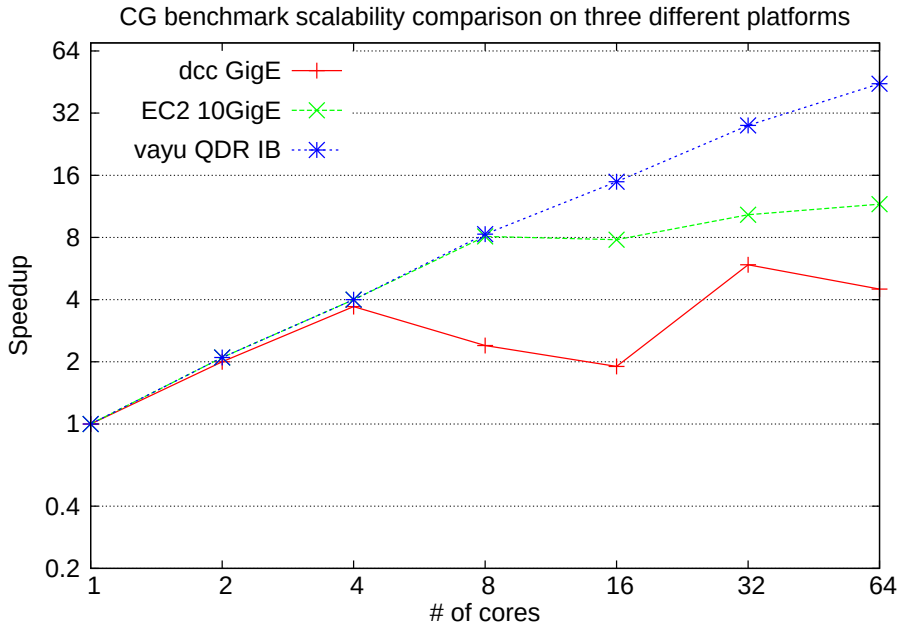
IS.B speedup, 1 to 64 cores

- EC2 fluctuations for EP.B suspected due to jitter (CPU scheduling and hyperthreading
- IS shows the poorest scaling of all benchmarks:
  - IPM profiling shows % communication at 64 cores is 98% (DCC), 85% (DCC) and 68% (vayu)

8 Results: NAS Parallel Benchmarks (II)



FT.B speedup, 1 to 64 cores

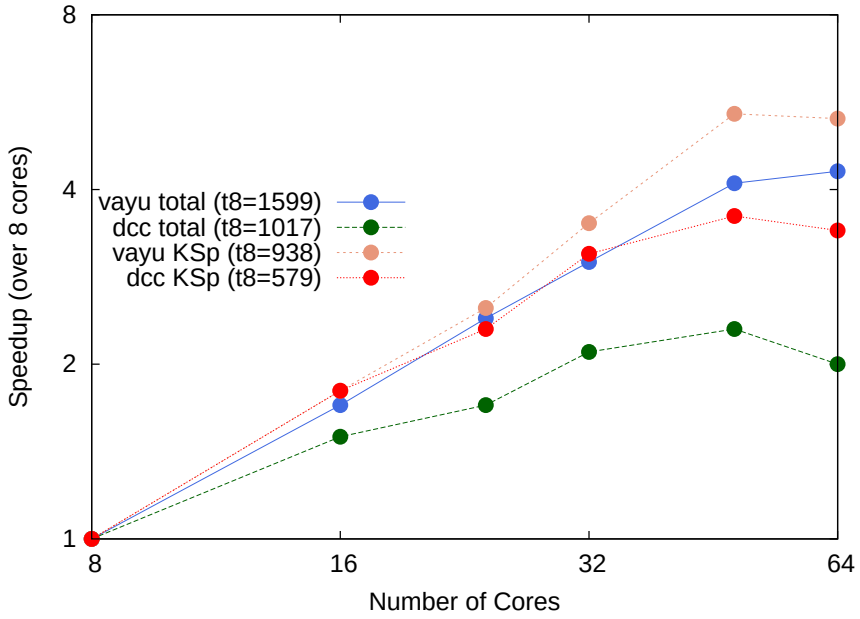


CG.B speedup, 1 to 64 cores

- CG.B: IPM profiling shows % communication at 64 cores is 90% (DCC), 58% (DCC) and 22% (vayu)
- drop-offs on clouds occur when intra-node communication is required
- BT.B, MG.B, SP.B and LU/B showed similar scaling to FT.B
- single core performance consistently 20% (30%) faster on EC2 (vayu)

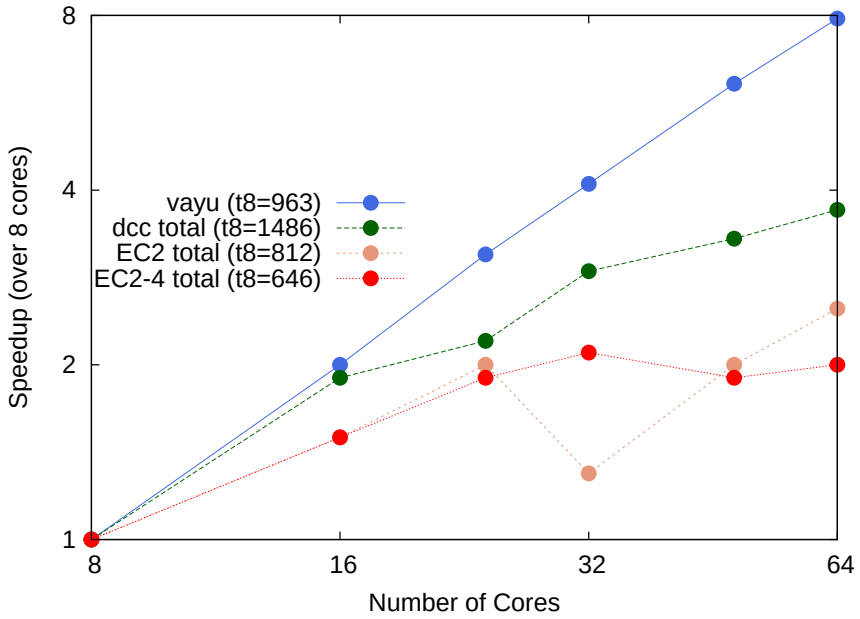
9 Results: Chaste Cardiac Simulation

- (results not available on EC2 due to complex dependencies)
- scaling of the KSp linear solver determines overall trends
  - note: benchmark scales to 1024 cores on vayu
- input mesh: 1.4× faster on vayu (8 cores), scaled the same on both
- output: 2.6× faster on vayu (8 cores) but scaled better on dcc



- @ 32 cores, 48% vs 11% of time spent in communication on dcc vs vayu
  - 13× more spent in KSp solver on dcc (large numbers of collectives)
- IPM profiles also indicated a greater degree and a higher irregularity of load imbalance on DCC
- ⇒ dcc performance hurt by high message latencies & jitter

10 Results: MetUM Global Atmosphere Simulation



	Vayu	DCC	EC2	EC2-4
time(s)	303	624	770	380
r_{comp}	1.0	1.37	2.39	1.17
r_{comm}	1.0	6.71	3.53	1.61
%comm	13	42	18	18
%imbal	13	4	18	19
I/O (s)	4.5	37.8	9.1	7.6

Details at 32 cores (EC2-4: 4 nodes used, uniformly 2× faster)

- overall load imbalance least on dcc, but generally higher & more irregular across individual sections (NUMA effects)
- EC-2 shows similar imbalance and communication profiles to vayu
- dcc spent most time in communication, particularly in sections where there were large numbers of collectives
- read-only I/O section: dcc much slower to vayu, EC2 similar, to vayu

11 Conclusions and Future Work

- largely successful in creating x86-64 binaries on HPC system & replicating all software dependencies into the VMs on clouds
- MPI micro-benchmarks and NPB benchmarks showed communication bound applications were disadvantaged on the virtualized platforms
  - large numbers of short messages were especially problematic
  - corroborated by the two applications
- over-subscription of cores and hidden hardware characteristics (e.g. NUMA) also affected cloud performance
  - only saw minor effects (e.g. jitter) directly attributable to virtualization
  - the underlying filesystem affected application I/O performance
- future work:
  - use metrics from the ARRIVE-F framework to assess candidate workloads for private/public science clouds
  - using StarCluster, cloud burst onto OpenStack based resources locally & externally

Acknowledgements

- thanks to Michael Chapman, David Singleton, Robin Humble, Ahmed El Zein, Ben Evans and Lindsay Botten at the NCI-NF for their support and encouragement!

Questions???