Cloudcom 2010
November 1, 2010
Indianapolis, IN

Performance of HPC Applications on the Amazon Web Services Cloud

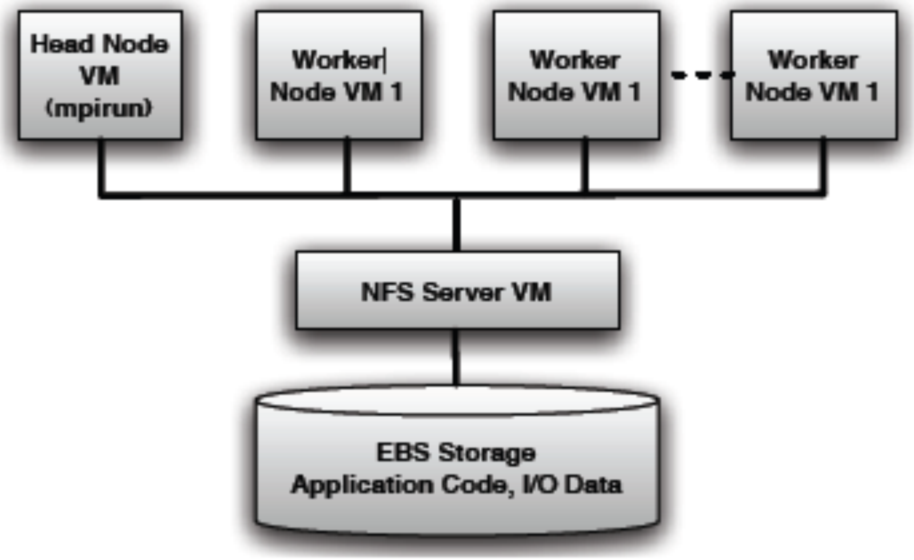
Keith R. Jackson, Lavanya Ramakrishnan, Krishna Muriki, Shane Canon, Shreyas Cholia, Harvey J. Wasserman, Nicholas J. Wright
Lawrence Berkeley National Lab

Goals

- Understand the performance of Amazon EC2 for realistic HPC workloads
  - Cover both the application space and algorithmic space of typical HPC workloads
- Characterize EC2 performance based on the communication patterns of applications

Methodology

- Create cloud virtual clusters
    - configure a file server, head node, and a series of worker nodes.
- Compile codes on local LBNL system with Intel Compilers / OpenMPI, move binary to EC2



Virtual Cluster Architecture

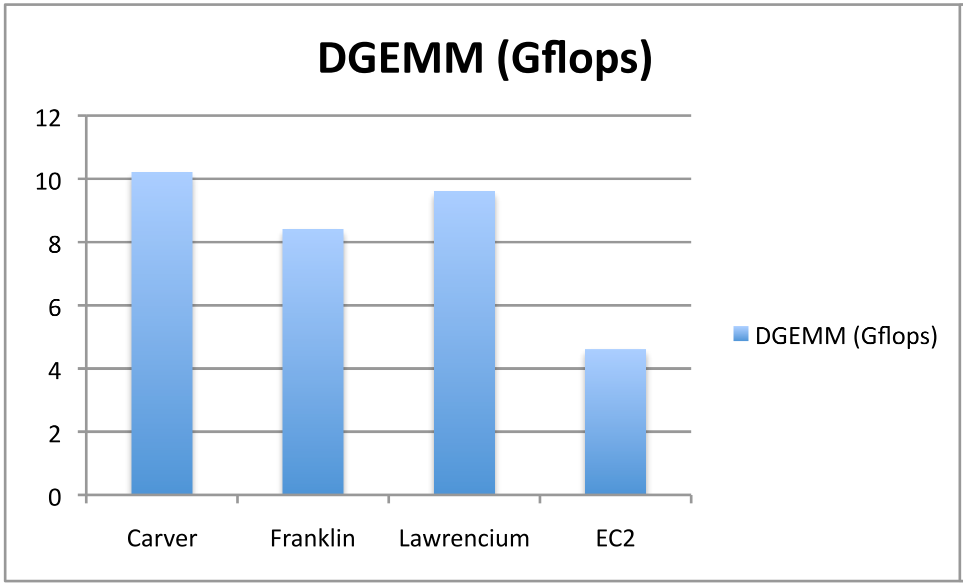
Hardware Platforms

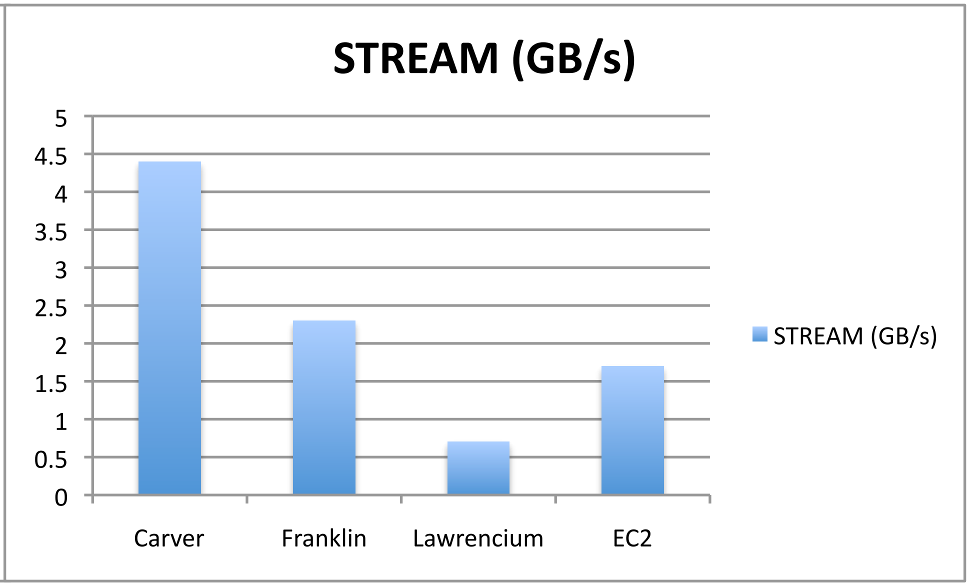
- Franklin:
  - Cray XT4
  - Linux environment / Quad-core, AMD Opteron / Seastar interconnect, Lustre parallel filesystem
  - Integrated HPC system for jobs scaling to tens of thousands of processors; 38,640 total cores
- Carver:
  - Quad-core, dual-socket Linux / Nehalem / QDR IB cluster
  - Medium-sized cluster for jobs scaling to hundreds of processors; 3,200 total cores

Hardware Platforms

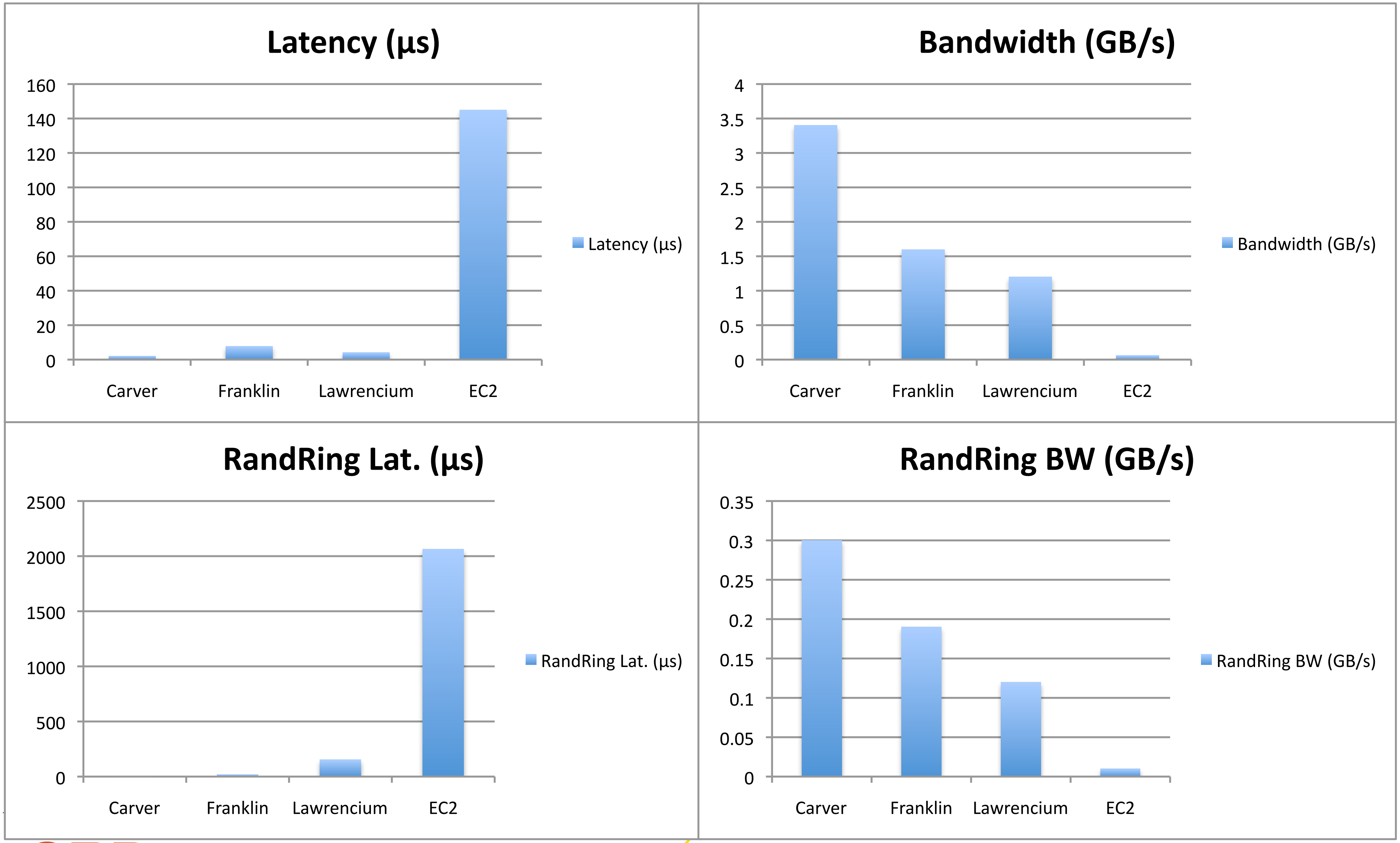
- Lawrencium:
  - Quad-core, dual-socket Linux / Harpertown / DDR IB cluster
  - Designed for jobs scaling to tens-hundreds of processors; 1,584 total cores
- Amazon EC2:
  - m1.large instance type: four EC2 Compute Units, two virtual cores with two EC2 Compute Units each, and 7.5 GB of memory
  - Heterogeneous processor types

HPC Challenge





HPC Challenge (cont.)



CRD computational research division

NERSC

BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

NERSC 6 Benchmarks

- A set of applications selected to be representative of the broader NERSC workload
  - Covers the science domains, parallelization schemes, and concurrencies, as well as machine-based characteristics that influence performance such as message size, memory access pattern, and working set sizes

Applications

- CAM: The Community Atmospheric Model
  - Lower computational intensity
  - Large point-to-point & collective MPI messages
- GAMESS: General Atomic and Molecular Electronic Structure System
  - Memory access
  - No collectives, very little communication
- GTC: GyrokineticTurbulence Code
  - High computational intensity
  - Bandwidth-bound nearest-neighbor communication plus collectives with small data payload

Applications (cont.)

- IMPACT-T: Integrated Map and Particle Accelerator Tracking Time
  - Memory bandwidth & moderate computational intensity
  - Collective performance with small to moderate message sizes
- MAESTRO: A Low Mach Number Stellar Hydrodynamics Code
  - Low computational intensity
  - Irregular communication patterns
- MILC: QCD
  - High computation intensity
  - Global communication with small messages
- PARATEC: PARAllel Total Energy Code
  - Global communication with small messages

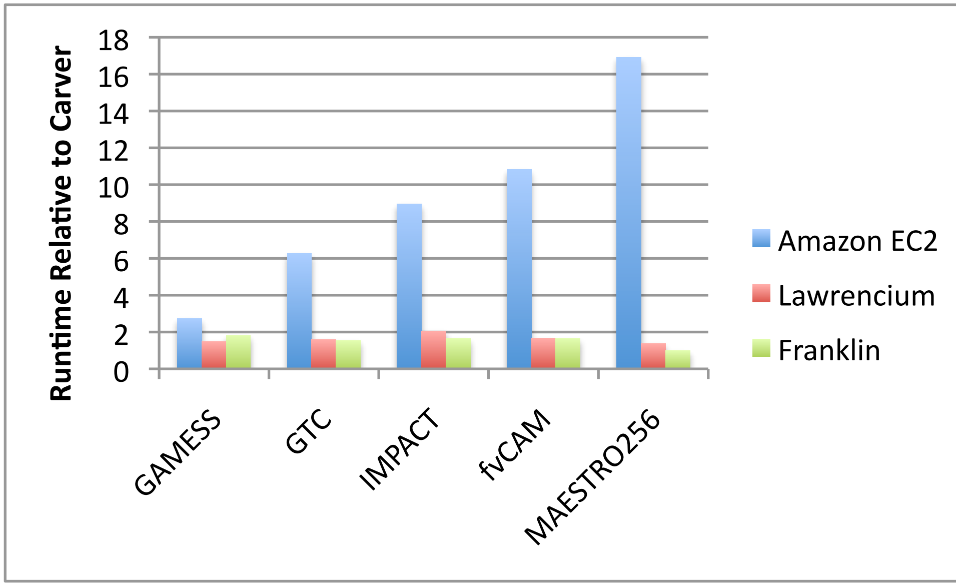
Application and Algorithmic Coverage

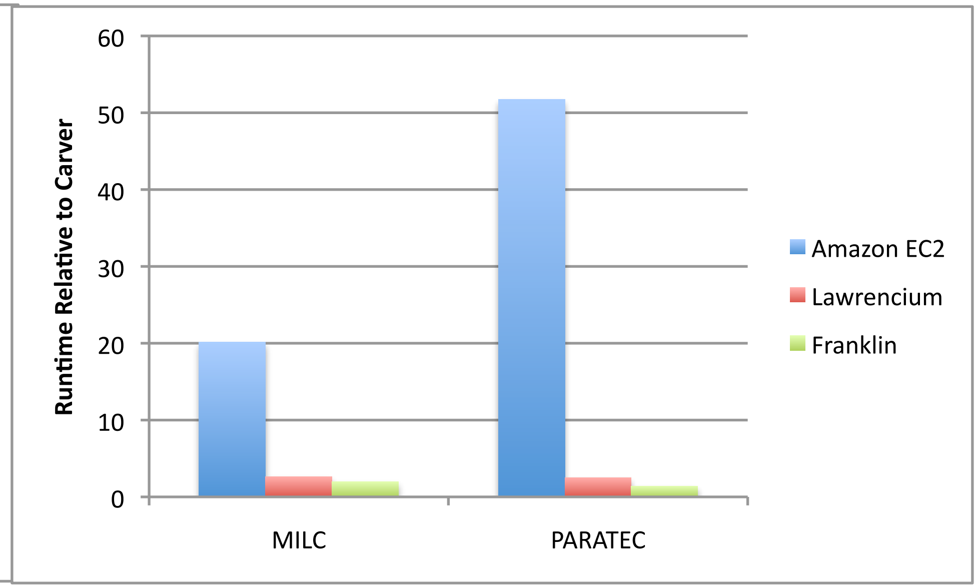
Science areas	*Dense linear algebra*	*Sparse linear algebra*	*Spectral Methods (FFT)s*	*Particle Methods*	*Structured Grids*	*Unstructured or AMR Grids*
Accelerator Science		X	X **IMPACT-T**	X **IMPACT-T**	X **IMPACT-T**	X
Astrophysics	X	X **MAESTRO**	X	X	X **MAESTRO**	X **MAESTRO**
Chemistry	X **GAMESS**	X	X	X		
Climate			X **CAM**		X **CAM**	X
Fusion	X	X		X **GTC**	X **GTC**	X
Lattice Gauge		X **MILC**	X **MILC**	X **MILC**	X **MILC**	
Material Science	X **PARATEC**		X **PARATEC**	X	X **PARATEC**	

Performance Comparison

Communication intensive applications suffer disproportionately on EC2

(Time relative to Carver)





NERSC Sustained System Performance (SSP)

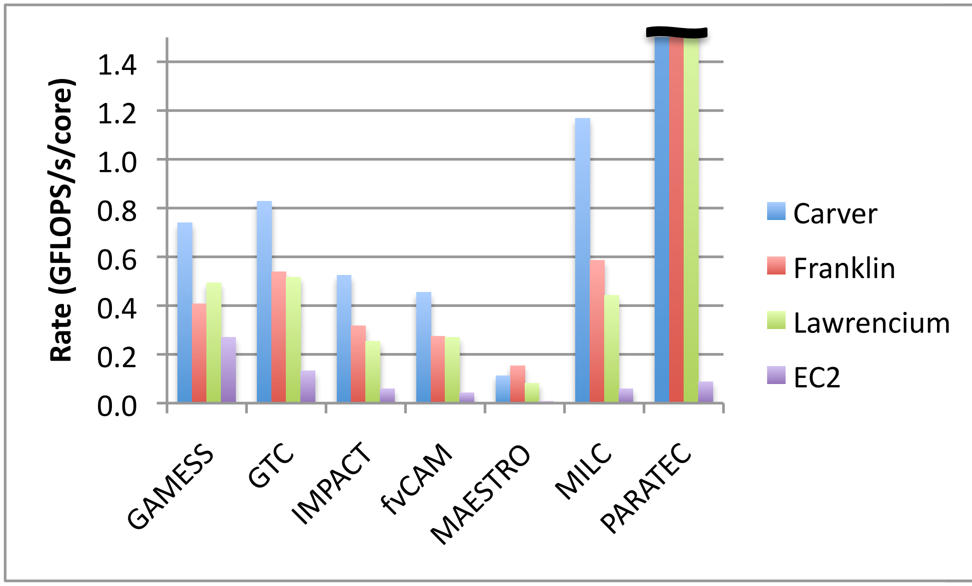
SSP represents delivered performance for a real workload

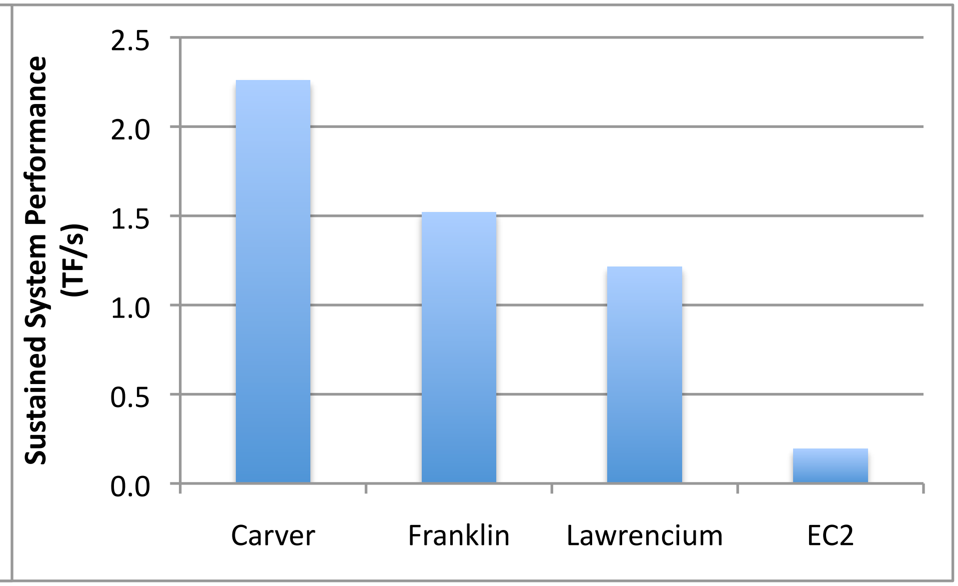
CAM Climate	*GAMESS* Quantum Chemistry	*GTC* Fusion	*IMPACT-T* Accelerator Physics	*MAESTRO* Astro-physics	*MILC* Nuclear Physics	*PARATEC* Material Science

- **SSP: aggregate measure of the workload-specific, delivered performance of a computing system**
- **For each code measure**
  - **FLOP counts on a reference system**
  - **Wall clock run time on various systems**
  - ***N* chosen to be 3,200**
- **Problem sets drastically reduced for cloud benchmarking**

$$SSP = N \left(\prod_{i=1}^{M} P_i \right)^{(1/M)}$$

Application Rates and SSP





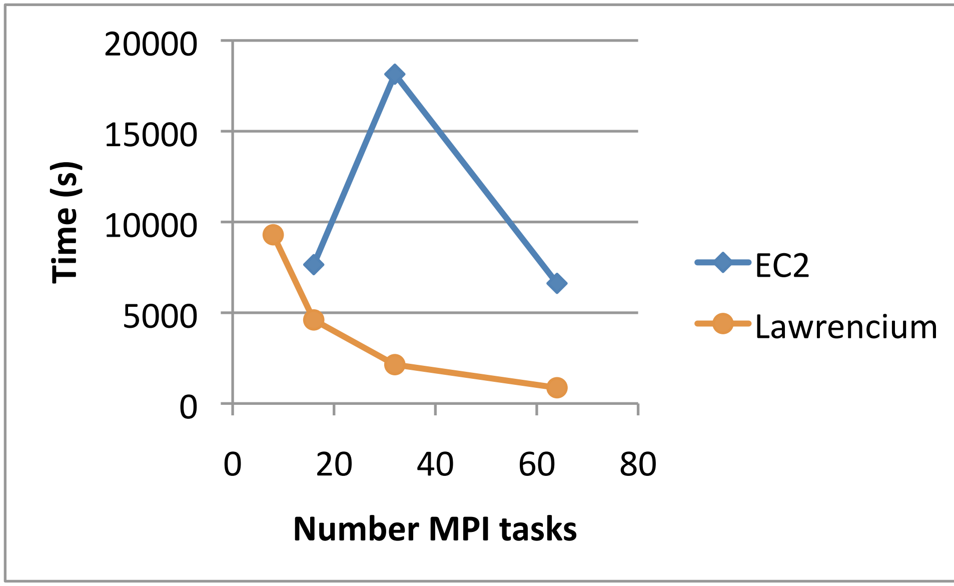
Problem sets drastically reduced for cloud benchmarking

CRD computational research division

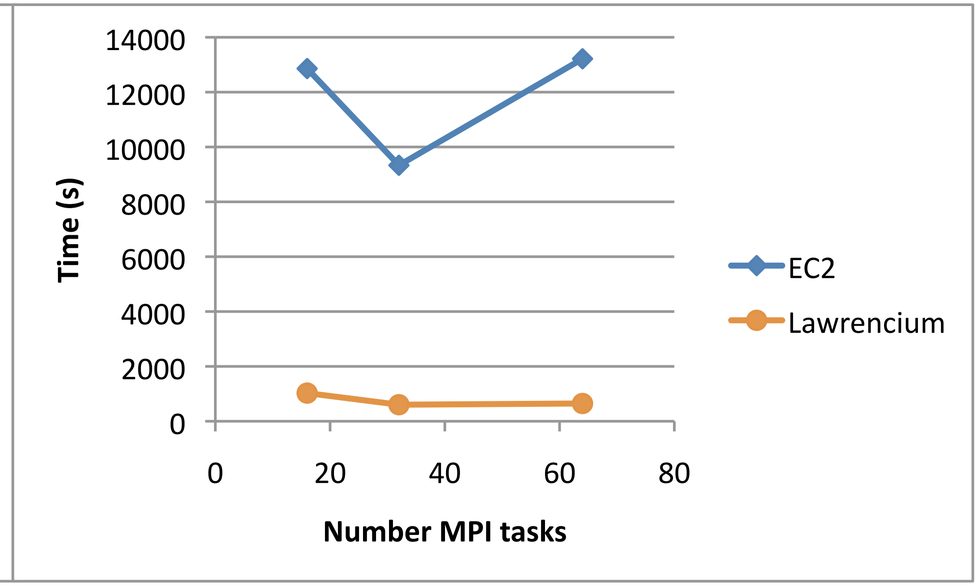
NERSC

BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

Application Scaling & Variability

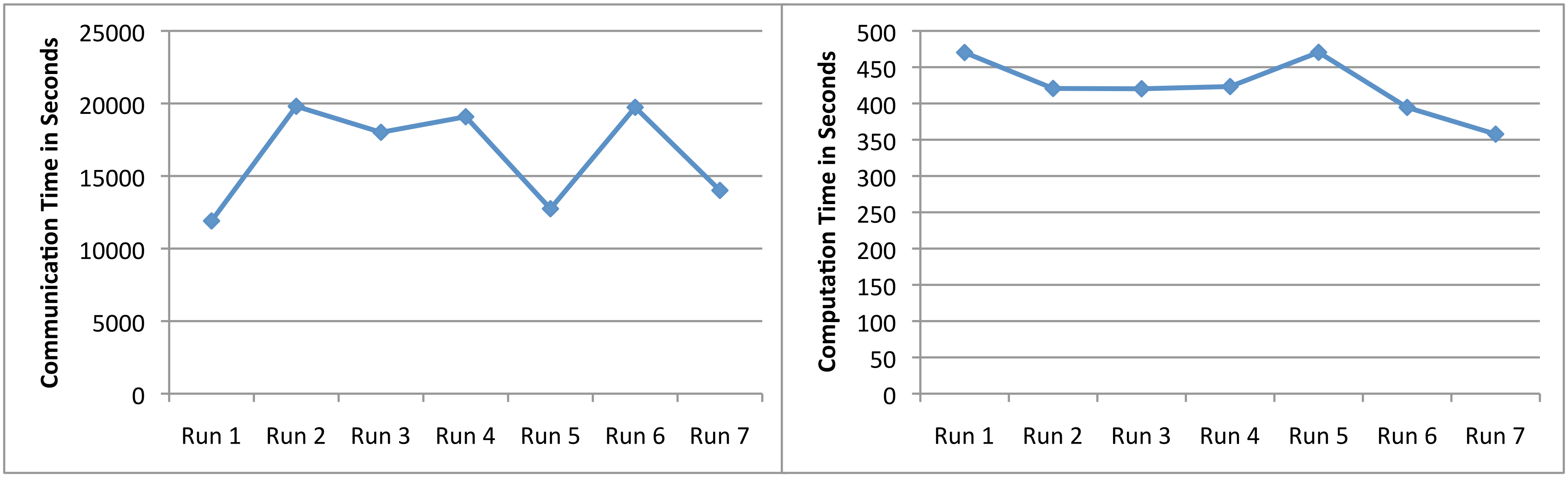


MILC

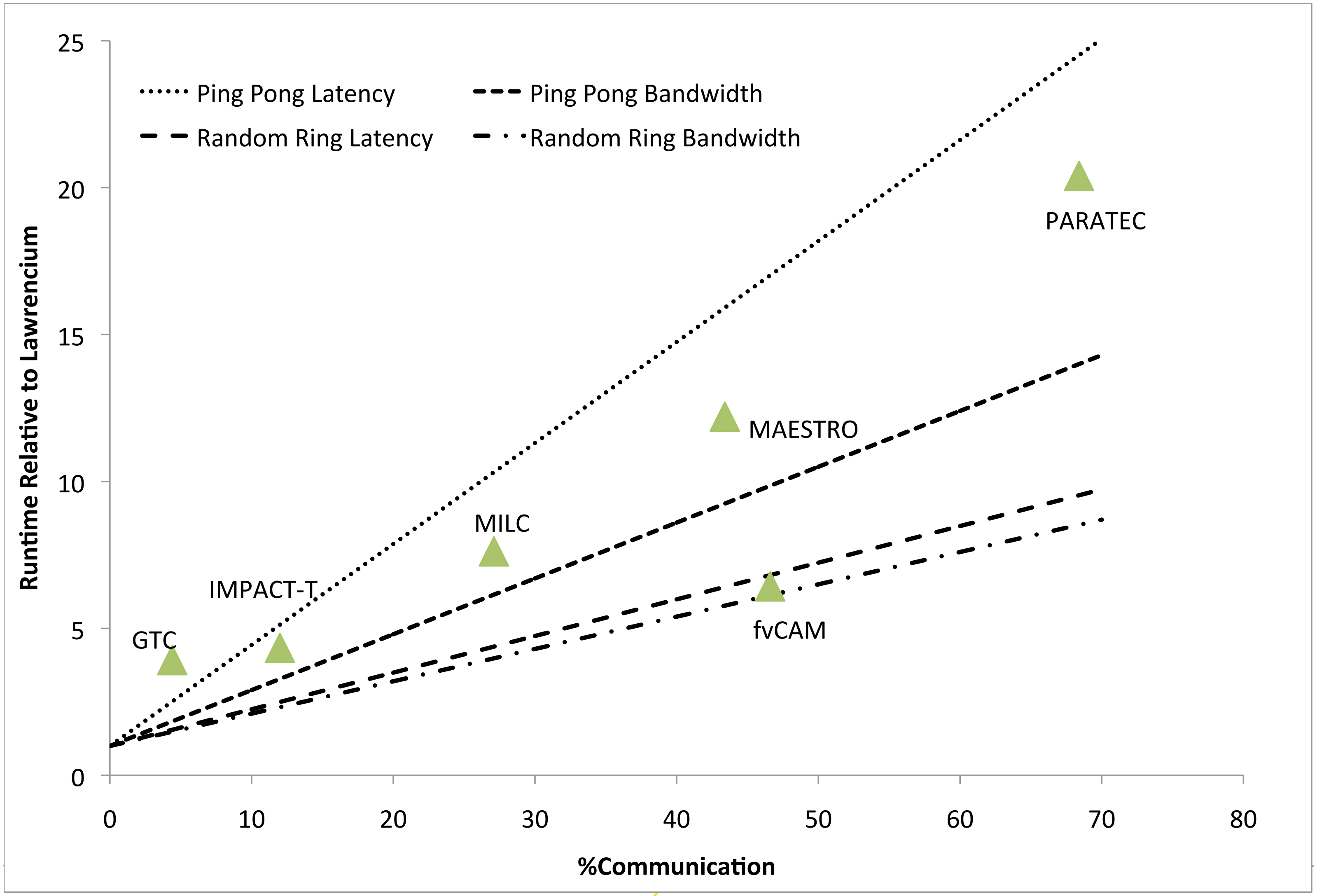


PARATEC

PARATEC Variability



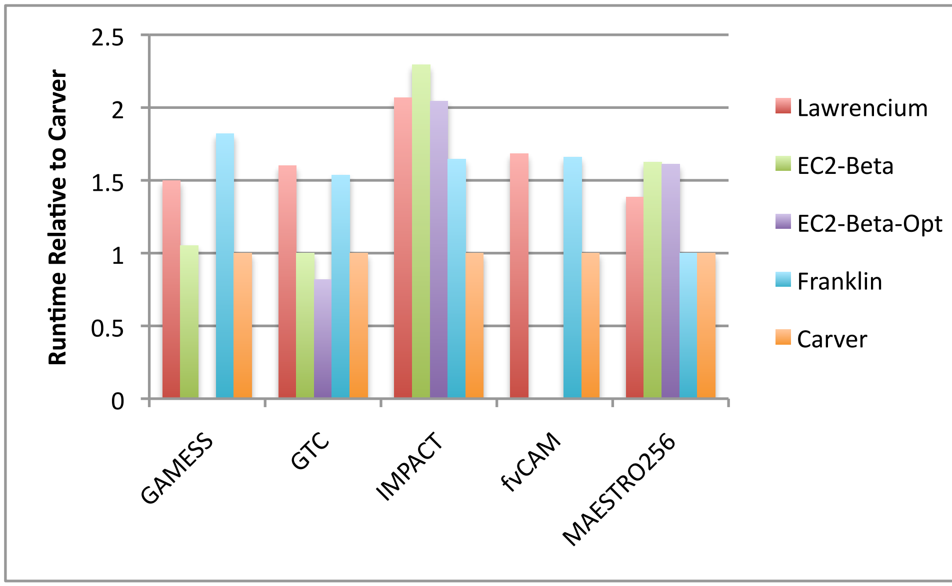
Communication/Perf Correlation

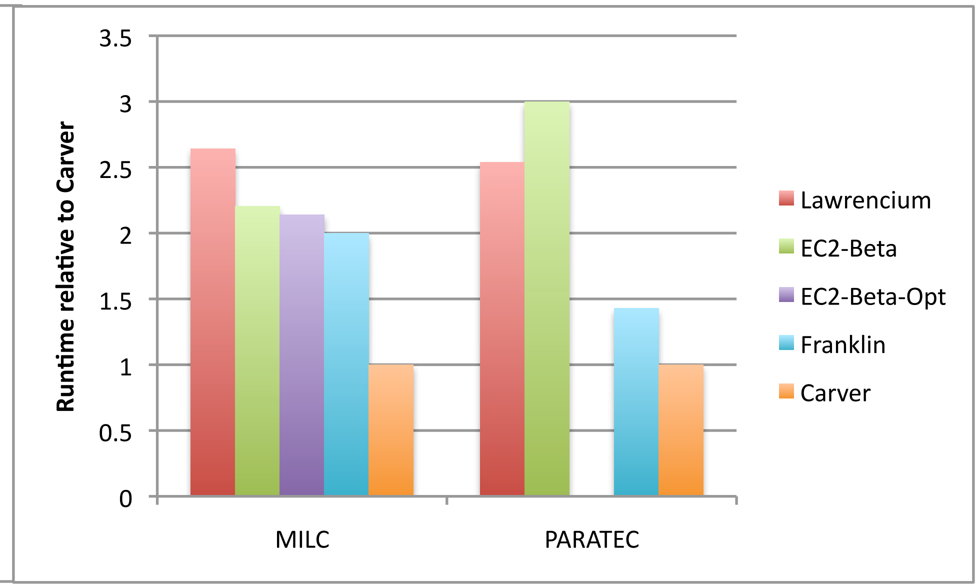


Observations

- Significant performance variability
  - Heterogeneous cluster: 2.0-2.6GHz AMD, 2.7GHz Xeon
  - Shared interconnect
  - sharing un-virtualized hardware?
- Significantly higher MTBF
- Variety of transient failures, including
  - inability to access user data during image startup;
  - failure to properly configure the network;
  - failure to boot properly;
  - intermittent virtual machine hangs;
  - Inability to obtain requested resources.

Amazon Cluster Compute Instances





Conclusions

- Standard EC2 performance degrades significantly as applications spend more time communicating
- Applications that stress global, all-to-all communication perform worse then those that mostly use point-to-point communication
- The Amazon Cluster Compute offering has significantly better performance for HPC applications then standard EC2

Acknowledgements

- This work was funded in part by the Advanced Scientific Computing Research (ASCR) in the DOE Office of Science under contract number DE-C02-05CH11231
- Nicholas J Wright was supported by the NSF under award OCI-0721397
- Special thanks to Masoud Nikravesh and CITRIS, UC Berkeley for their generous donation of Amazon EC2 time