Cloud Computing Workshop, Argonne, 2009

Scientific Cloud Computing

F. Vila, J. P. Gardner, L. Svec and J. J. Rehr

Department of Physics, University of Washington
Seattle, WA 98195-1560

J. J. Rehr, J. P. Gardner, L. Svec and F. Vila, arXiv:0901.0029 (2008)

Supported by NSF DMR-0848950

Challenge of NSF Grant

- Is Cloud Computing feasible for on-demand, High-Performance Computing (HPC) for scientific research in the face of declining budgets?
- Who is interested?
- Is it for everybody?
- What kind of code could benefit from it?
- How do we make it possible?

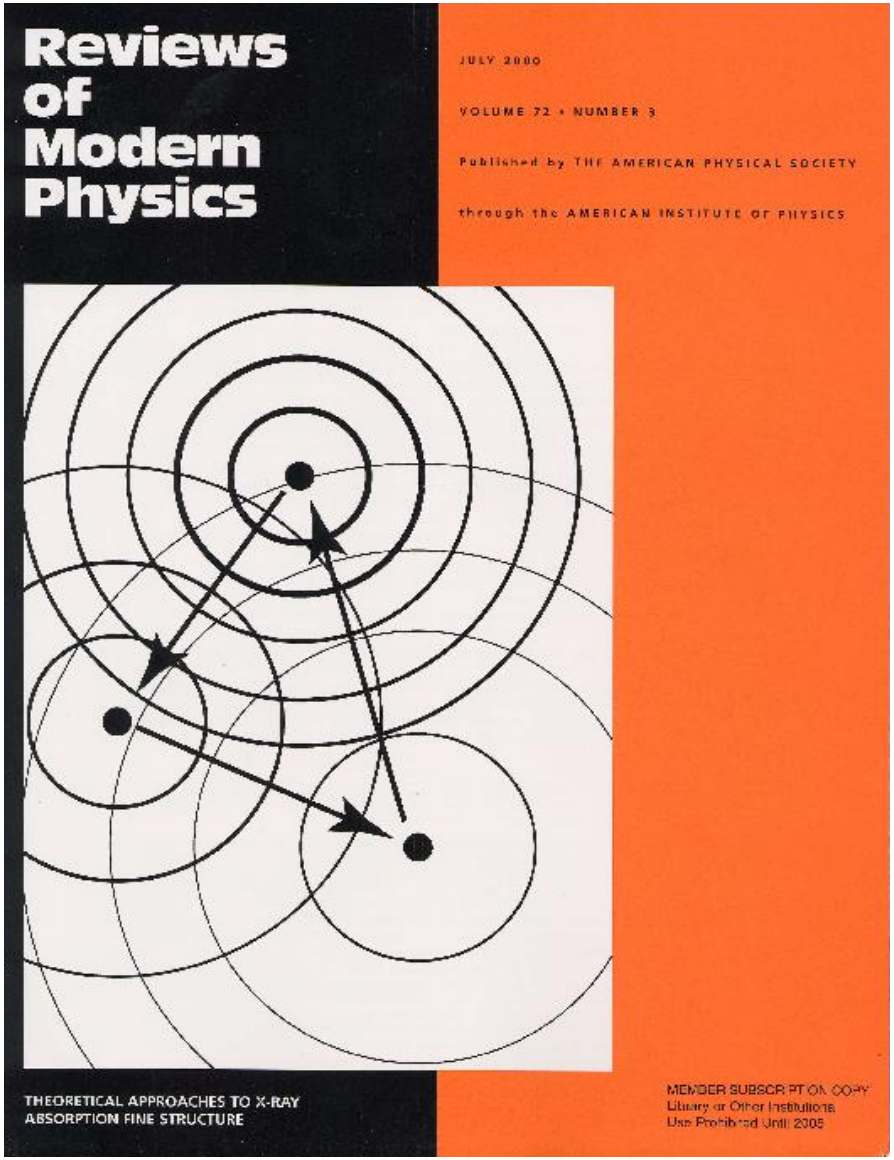
Disadvantages of Current HPC Approach

- Expensive infrastructure:

  Big clusters = ~1000$/node + capital costs + power + cooling + …
- Expensive HPC staff & maintenance
- Need expertise in HPC to use efficiently

Advantages of CC for Scientific Computing

- For “casual” HPC users:
  - On-demand access without the need to purchase, maintain, or even understand HPCs
  - Lease *vs* buy: lease as many as needed at ~10¢/cpu-hr
  - Plug & Play HPC scientific codes
- For developers:
  - Scientific codes can be optimized and pre-installed
- For administrators & funding agencies:
  - HPC access to a wider class of scientists at lower costs



Sample scientific application

FEFF:

Real-space Green's function code for electronic structure, x-ray spectra, EELS, …

Naturally parallel:

Each CPU calculate a few points in the energy grid

Loosely coupled:

Very little communication between processes

J. J. Rehr & R.C. Albers

Rev. Mod. Phys. **72**, 621 (2000)

http://leonardo.phys.washington.edu/feff/

The FEFF User Base

- Not usually “gurus”:
  - Very little experience with code compilation and optimization
  - Very few have experience with parallel computing
  - Little access to HPC resources
  - “Would rather be doing science..”
- Scientific problem usually limited by:
  - The amount of memory required by the job
  - The amount of time it would take to complete it

Scientific Cloud Computing Development Strategy

1. Develop AMI (Amazon Machine Image) customized for HPC scientific applications
2. Test single-instance performance
3. Develop shell-scripts that make the EC2 look and run like a local HPC cluster (″virtual supercomputer on a laptop″)
4. Test parallel performance

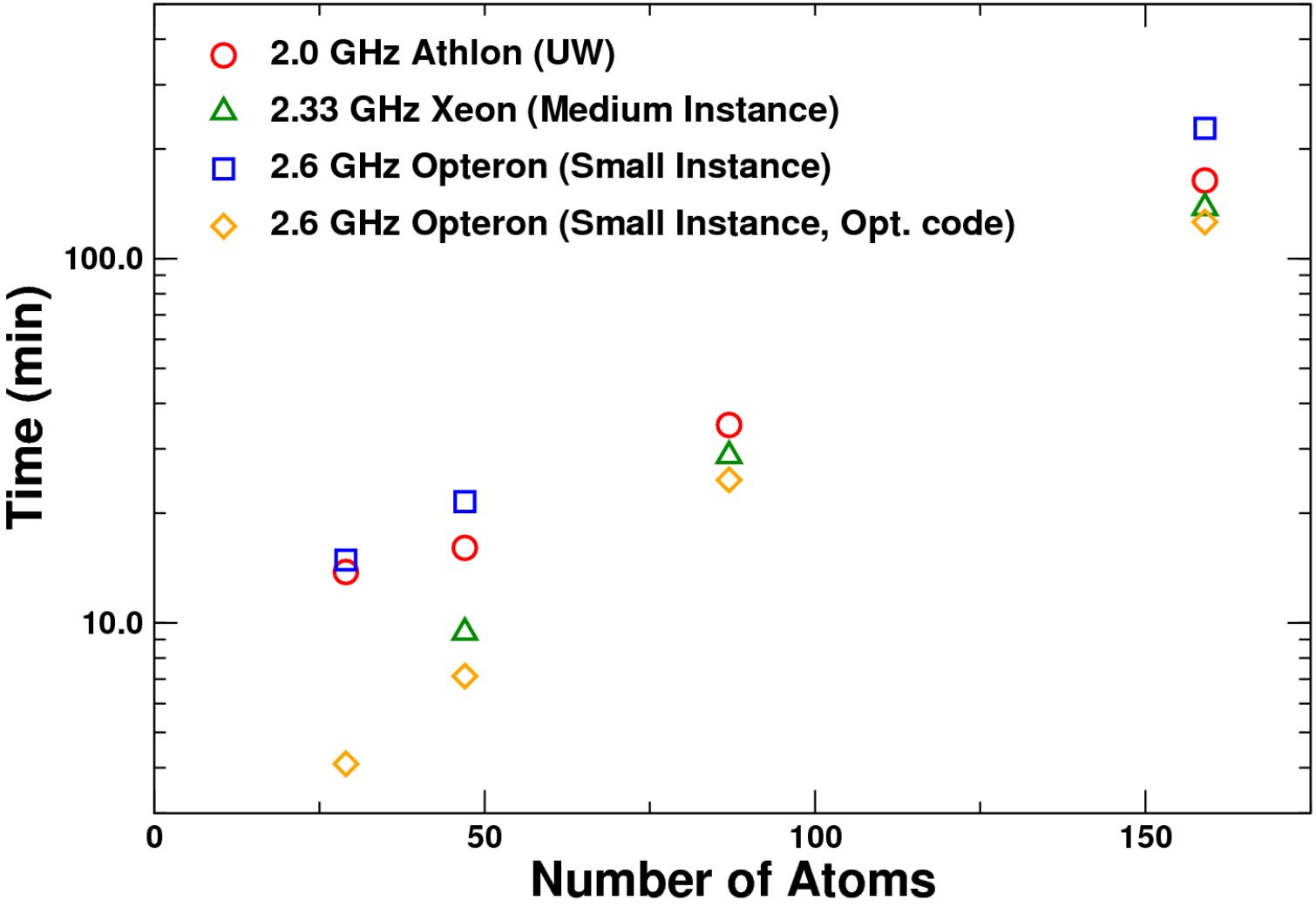
FEFFMPI EC2 AMI

Custom Linux distribution replicated on each instance in cluster

- Standard Linux AMI:

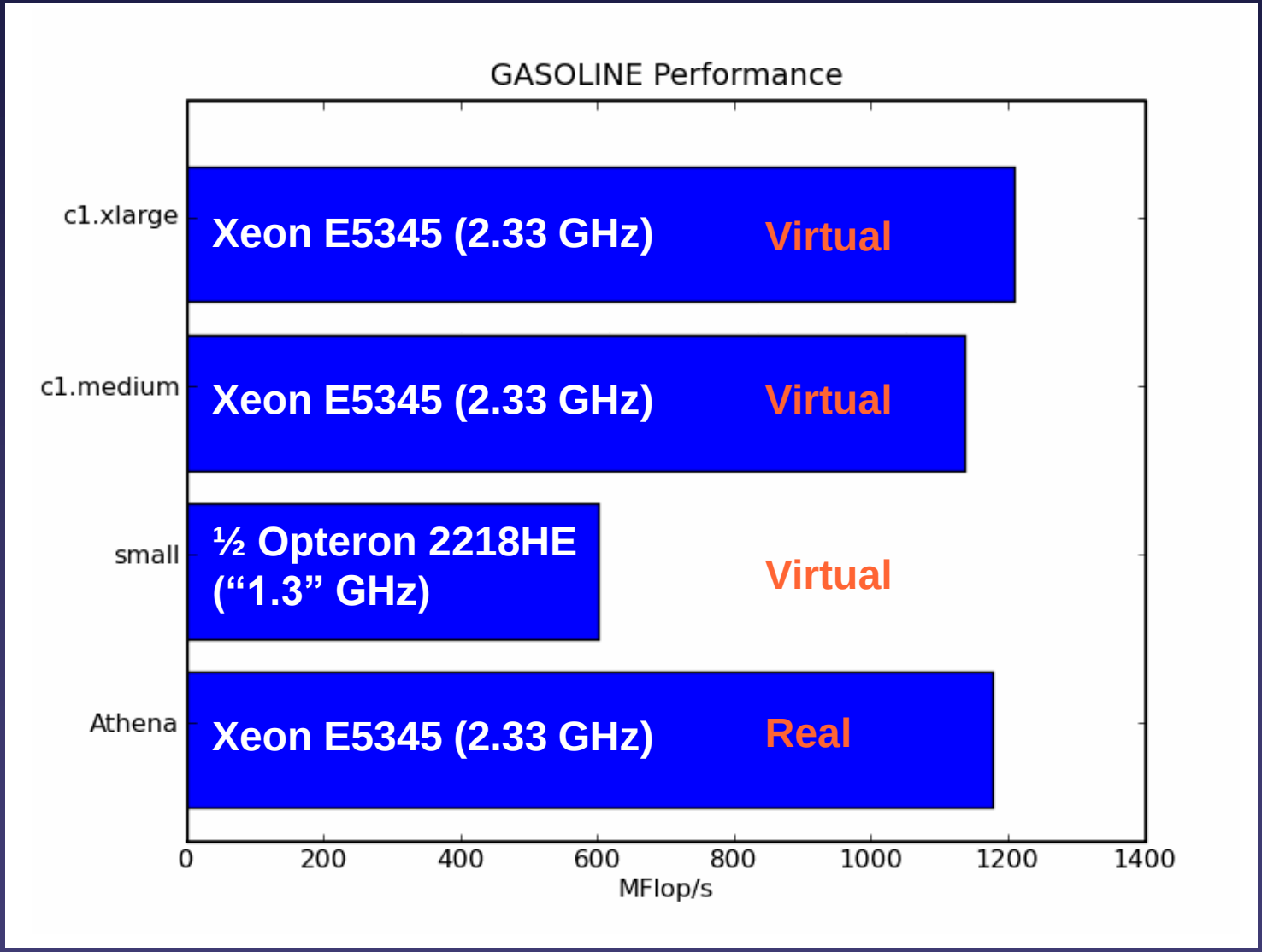
  Fedora 8 32-bit distribution with Gnu FORTRAN compilers (gfortran and g77)
- AWS tools for the EC2: AMI, API and S3 tools
- LAM 7.1.4 for parallel MPI codes
- Java Runtime Environment 6
- Java Development Kit 1.6
- EC2 Cluster tools
- FEFF8.4 serial and parallel versions
- JFEFF graphical interface for FEFF8.4

Serial Performance of FEFF on the EC2



Virtual machine performance similar to “real” one

Serial Performance of Gasoline on the EC2

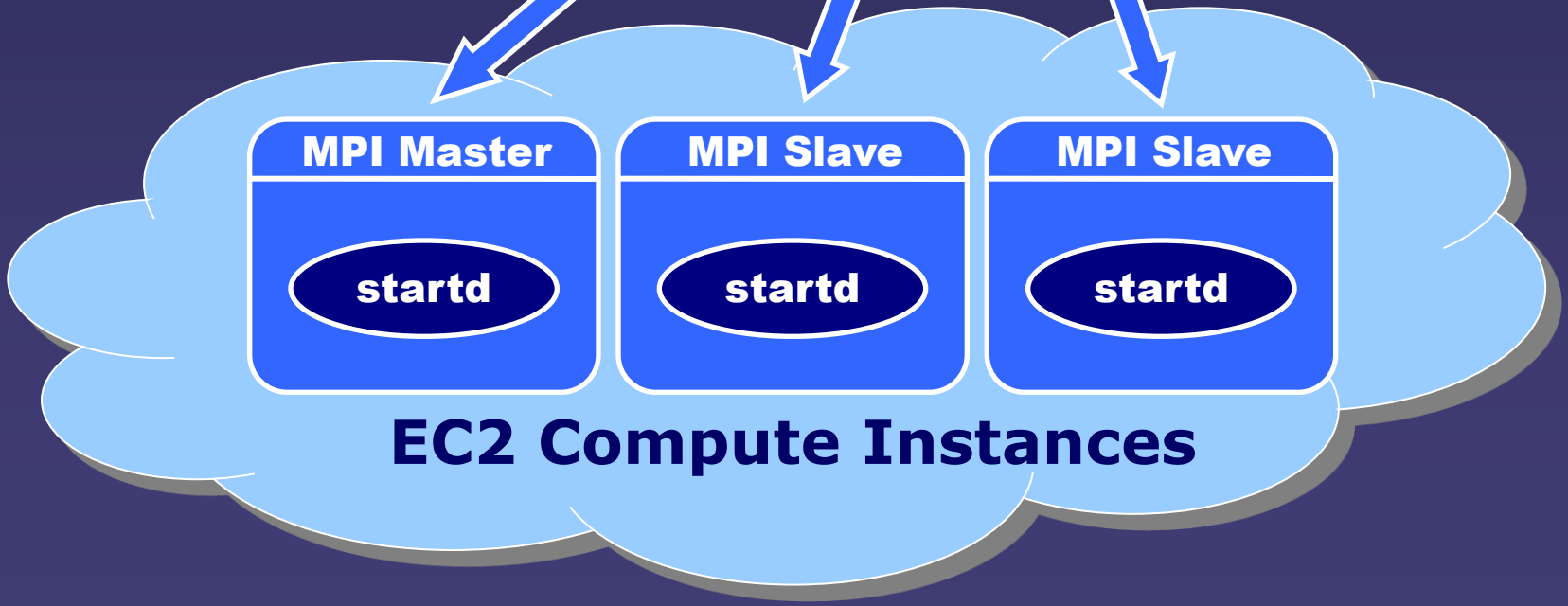


No penalty from virtualization

Current MPI Scenario

User interacts with control workstation

1. Start cluster
2. Configure nodes



UW EC2 Cluster Tools

Tools in the local control machine

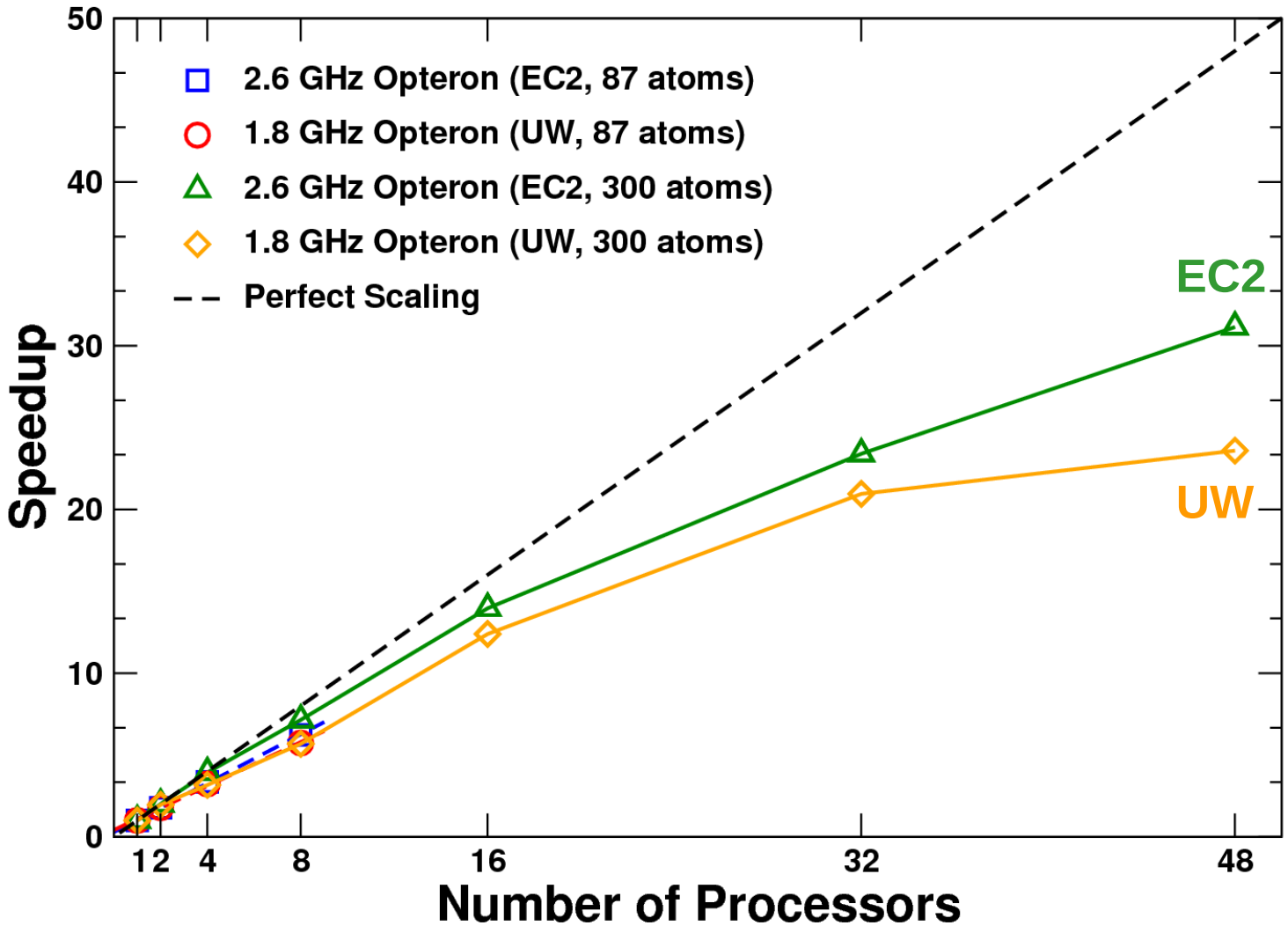
Name	Function	Analog
`ec2_clust_launch` *N*	Launches cluster with N instances	boot
`ec2_clust_connect`	Connect to a cluster	ssh
`ec2_clust_put`	Transfer data to EC2 cluster	scp
`ec2_clust_get`	Transfer data from EC2 cluster	scp
`ec2_clust_list`	List running clusters	
`ec2_clust_terminate`	Terminate a running cluster	shutdown

The tools hide a lot of the “ugliness”:

ec2_clust_connect

ssh -i /home/fer/.ec2_clust/.ec2_clust_info.7729.r-de70cdb7/key_pair _fdv.pem root@ec2-72-44-53-27.compute-1.amazonaws.com

FEFFMPI on the EC2



EC2 works well for highly parallelized applications like FEFF

Conclusions

- Benchmark before doing science on EC2
- EC2 well suited for loosely coupled applications
- Network performance on par with in-building fabrics
- Virtualization overhead is minimal
- EC2 likely advantageous for code developers and users
- In principle, EC2 is a cost effective HPC solution

Open Questions

- Will CC work for tightly coupled applications?
- How can we efficiently balance cost with performance?
- Are “High-CPU Extra Large Instances” cost-effective?
- Is it better to run on budget instances?
- How do we safeguard our code and the users data?

Reference: arXiv:0901.0029; fdv@uw.edu

Cloud Computing Workshop, Argonne, 2009

Scientific Cloud Computing

Amazon:	**J. Barr, T. Laxdal, D. Singh, P. Sirota,**
	P. Sridharan, R. Valdez, and W. Vogels
UW:	**R. Coffey, E. Lazowska, M. Prange,**
	C. Reschke
Datawrangling:	**P. Skomoroch**
NSF:	**C. Bouldin**

SCC is supported by NSF grant DMR-0848950

FEFF is supported by DOE-BES grant DE-FG03-97ER45623

... and thank you

F. Vila, J. P. Gardner, L. Svec and J. J. Rehr

Department of Physics, University of Washington

Seattle, WA 98195-1560