Finding Tropical Cyclones On Clouds: Using Parallel Virtualization for Large-Scale Data Analysis

Daren Hasenkamp, UC Berkeley & Berkeley Lab

Climate Simulation Generates Massive Data

- FVCAM program run for 15 simulated years with mesh point resolution .5 degrees latitude by .625 degrees longitude, generated ~500GB data
- Would like to run/analyze hundreds or thousands of simulations with varying initial conditions and other input parameters, hence, many terabytes to several petabytes data!

Difficulties in large-scale data analysis

- Analysis programs often are single-threaded, and take a long time to produce results.
- Software dependencies and run-time requirements make running code in unfamiliar environments difficult.
- Moving large datasets to compute nodes introduces extra burden on data management.
- Example: TSTORMS, written by Tom Knutson's group at the Geophysical Fluid Dynamics Laboratory, is a single-threaded analysis program to find tropical storms in climate data. It requires NetCDF library, NCO operators, and customized execution environment.

Parallelization through Virtualization

- Cloud computing paradigm supports configurable, massive parallelism.
- Virtualization easily replicates original execution environment and software dependencies.
- **Needs**: Coordinate virtual machines (VMs) for parallel data analysis.
- **Needs**: To manage data for analysis on VMs, stage in and out data between VM and remote repository.

VM Coordination using Leader Election

- **Idea**: Virtual machines use a distributed leader election algorithm to select a leader. The Leader acts as a synchronized queue for all virtual machines to coordinate data accesses.
- In our system, VMs use UDP broadcast to discover each other and elect a leader. The leader initializes a service that other VMs can contact to collect data URLs. VM instances stage in the data specified by the URL to their local file system, perform analysis, and stage results out to another remote repository.
- **Advantages**: Scalable, self-contained method of coordination.
- **Disadvantages**: Underlying network must support communication among all VM instances.

VM Coordination using a Remote Service

- **Idea**: VMs include a service address to contact, where the service acts as a synchronized queue for all VMs to pull data identifiers.
- In our system, VMs pull URLs from the remote service, stage in the data referred to by the URL, analyze it, and stage the results out to another remote repository.
- **Advantages**: VM instances can be spread far from each other. Easy to implement.
- **Disadvantages**: Added complexity for configuring and maintaining a remote service.

Results

- We ran the TSTORMS code on a ~500GB input dataset using both of our systems and MPI.
- The leader election system was run on the Magellan Scientific Cloud at Argonne National Lab.
- The remote service system was run at the Grid Laboratory of Wisconsin (GLOW).
- The MPI system was run on the Carver cluster at NERSC.
- Analysis speed is more predictable using MPI, but overall the speed is comparable.
- The variance and slowdown in virtual machine analysis time, especially on GLOW, is caused by the stress placed on the network by staging in large amounts of data from a remote repository, whereas the analysis with MPI accessed data through a local area network.
- We plan to improve this by adding functionality to the leader/remote service to limit simultaneous connections.

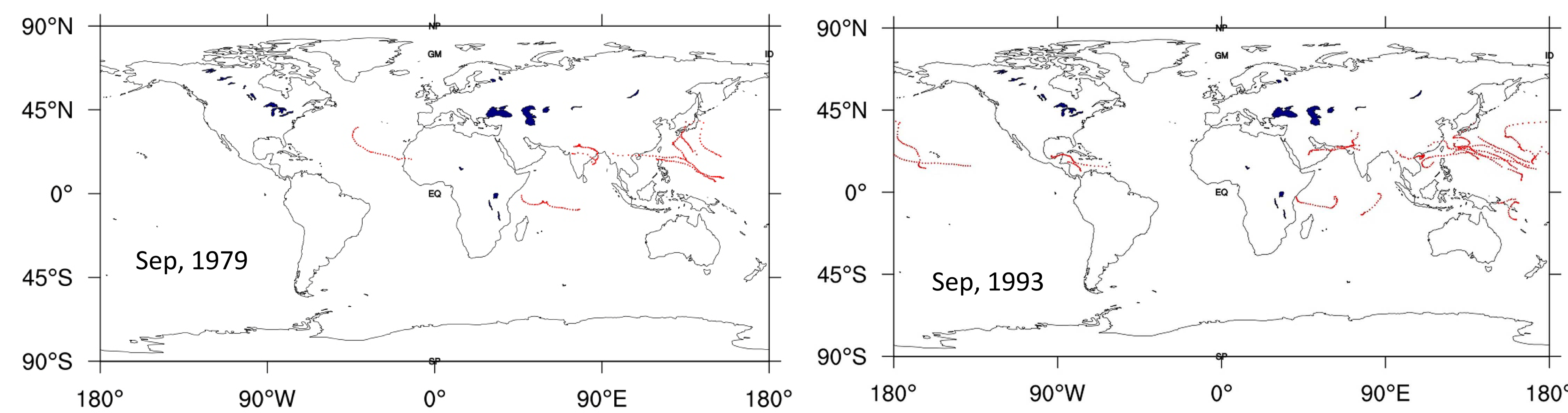

Figure 1: Finding Tropical Cyclones

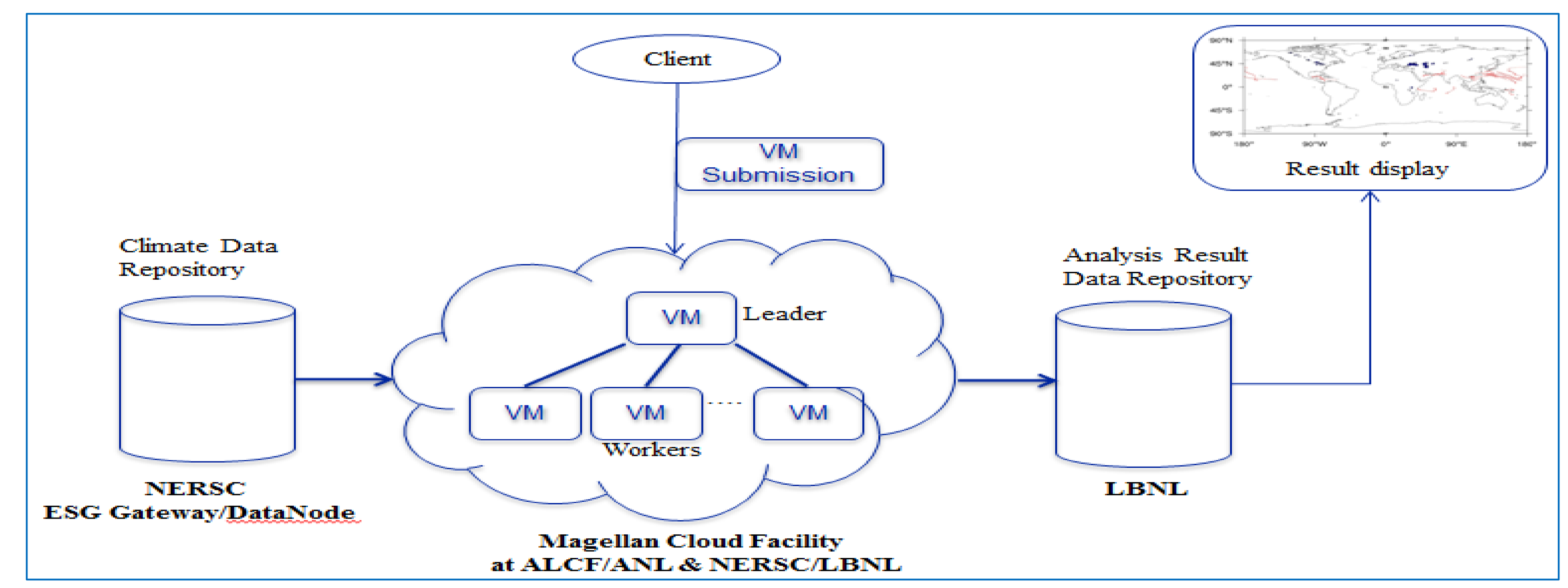

Figure 2: VM coordination with leader election

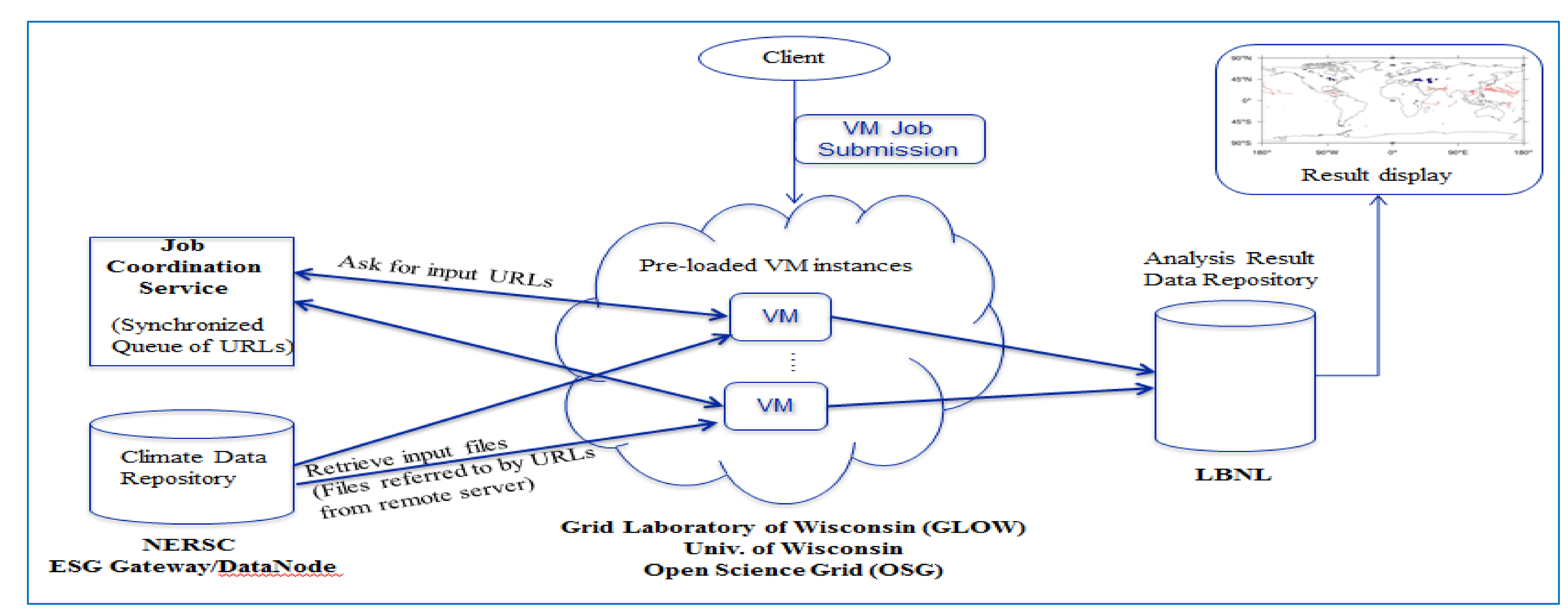

Figure 3: VM coordination with remote service

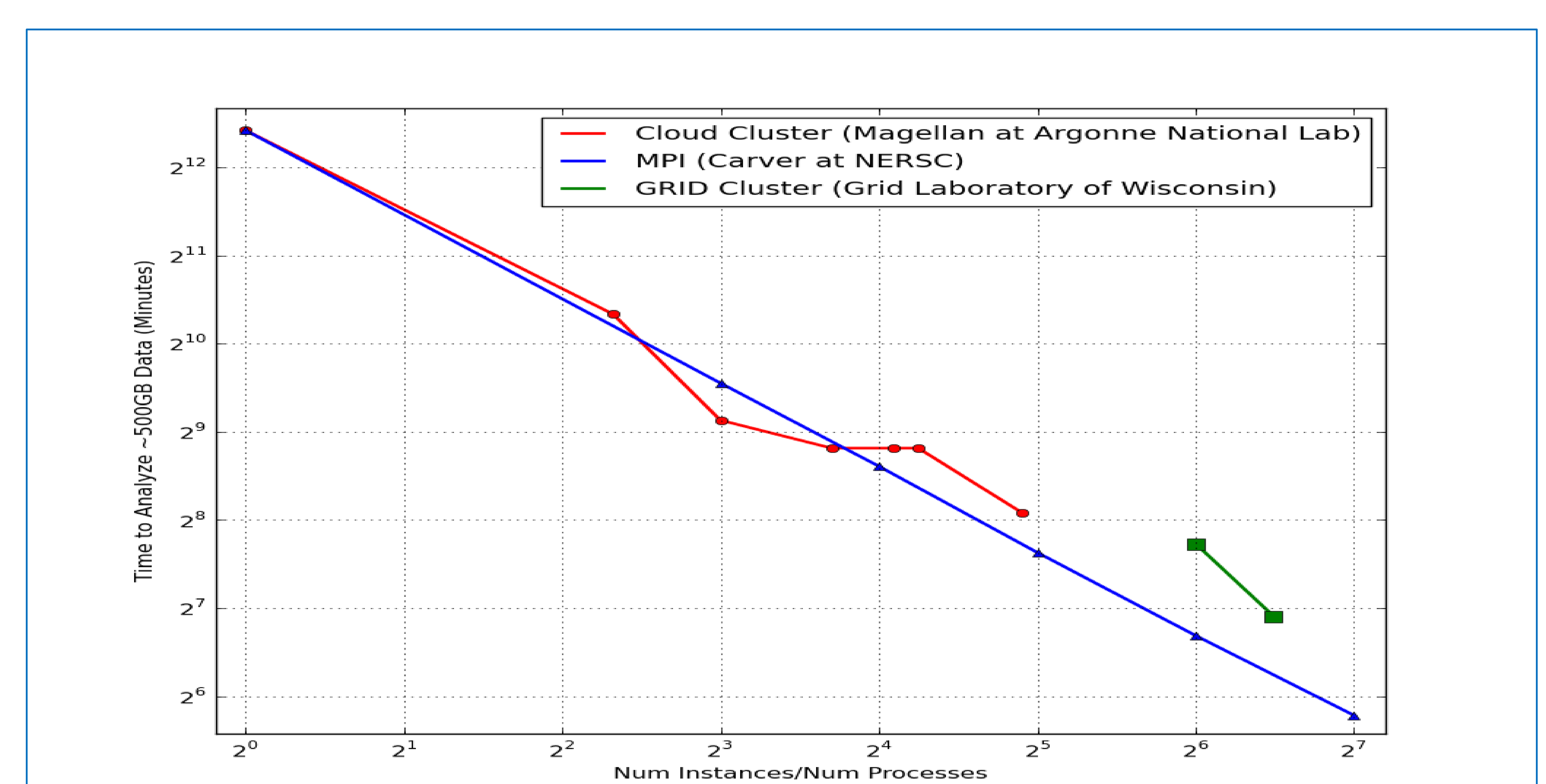

Figure 4: Timing results