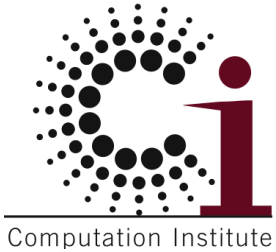


Clouds and Science: Opportunities and Obstacles

Daniel S. Katz

d.katz@ieee.org

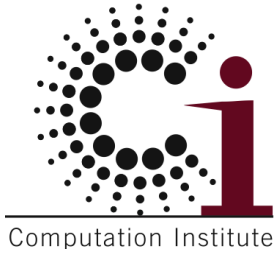
TeraGrid GIG Director of Science

Senior Fellow,
Computation Institute,
University of Chicago & Argonne
National Laboratory

Goal

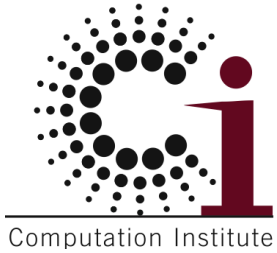
- Convince you that clouds are useful for science
  - But, some clouds are better than others for some applications
  - Or, different types of applications fir on different types of clouds
- Tell you about two applications
  - Finite Difference Time Domain (FDTD) Electromagnetics
  - Montage (astronomical image mosaics)
- Think about what is important about clouds
  - How applications are developed and mapped

Cloud basics



- NIST definition:
  - a computing capability that provides an abstraction between the computing resource and its underlying technical architecture (e.g., servers, storage, networks), enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction

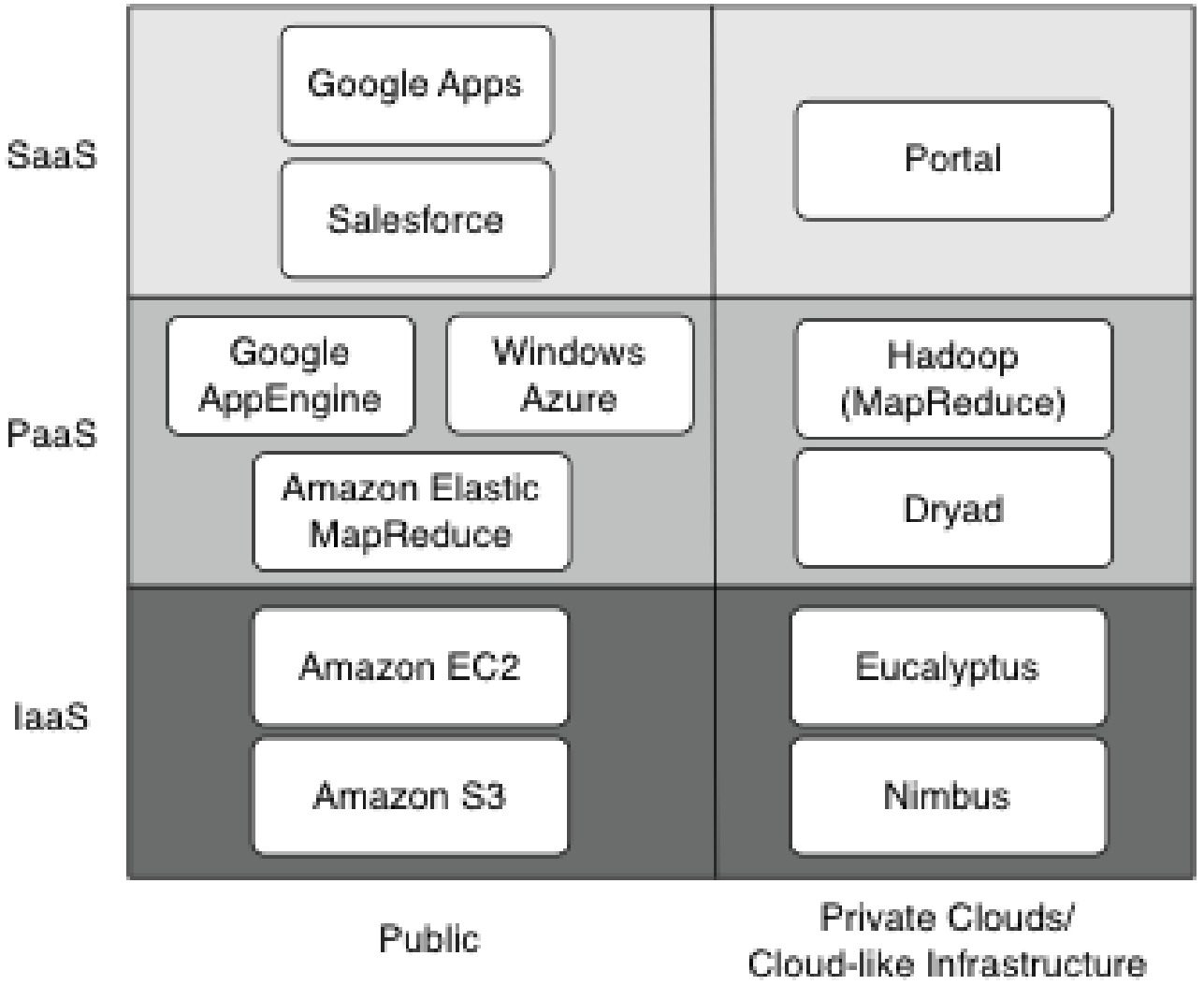
Clouds vs. Grids



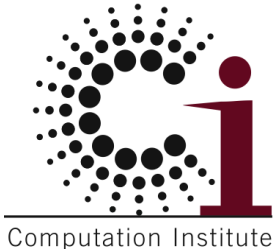
- Rich Wolski's assertion: Clouds and Grids are distinct
- Cloud
  - Individual user can only get a tiny fraction of the total resource pool
  - No support for cloud federation except through the client interface
  - Opaque with respect to resources
- Grid
  - Built so that individual users can get most, if not all of the resources in a single request
  - Middleware approach takes federation as a first principle
  - Resources are exposed, often as bare metal
- These differences mandate different architectures for each

Credit: Rich Wolski, Eucalyptus Systems

Clouds

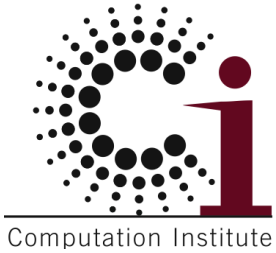

SaaS
PaaS
IaaS
Google Apps
Salesforce
Portal
Google AppEngine
Windows Azure
Amazon Elastic MapReduce
Hadoop (MapReduce)
Dryad
Amazon EC2
Amazon S3
Eucalyptus
Nimbus
Public
Private Clouds/ Cloud-like Infrastructure

Outline



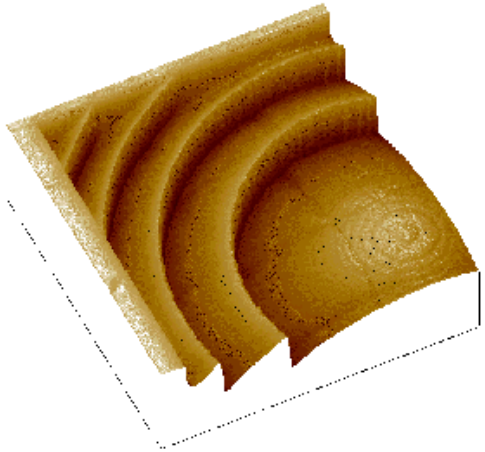
- Electromagnetics (FDTD): Sequential -> Parallel
- Astronomy (Montage): Parallel -> Grid
- Clouds

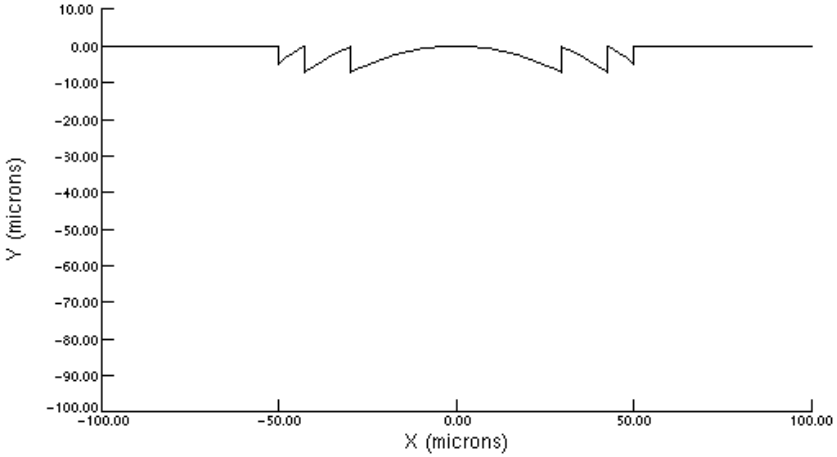
Electromagnetics



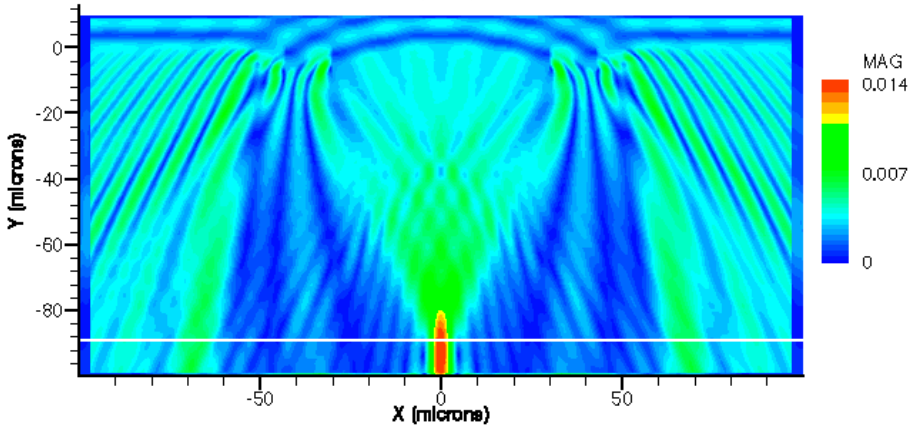
- Maxwell's Equations
  - Lots of versions, pick the right set for your problem and methodology
- Wavelength and frequency are inversely related
  - An object of size 1 m is one wavelength long at 300 MHz or 2 wavelengths long at 150 MHz
  - Either frequency or size can be scaled as needed
- Radar Cross Section (RCS) as an example problem
  - A plane wave at some incident angle and some frequency illuminates a target.
  - Monostatic or Backscatter RCS: what energy comes back?
  - Bistatic RCS: what energy goes off in another direction?

Dielectric Lens





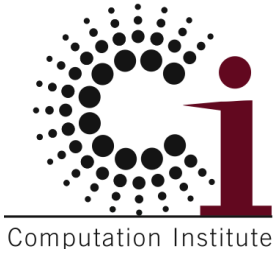
Lens etched into media with 3.1 index of refraction



- Dielectric lenses can be made in different materials with different properties
- Above quantum well infrared photodetector (QWIP), can increase QWIP sensitivity by 14x
- 20 THz plane wave incident downward

Image: Scanning Electron Microscope (SEM) of a portion of the dielectric lens (credit: Dan Wilson, JPL Microdevices Laboratory)

Maxwell's Equations in Curl Form



- Maxwell's (curl) Equations:

$$\frac{\partial \overline{B}}{\partial t} = -\nabla \times \overline{E} \qquad \frac{\partial \overline{D}}{\partial t} = \nabla \times \overline{H}$$

$\overline{E}$ = Electric field vector

$\overline{B}$ = Magnetic flux density vector

$\overline{D}$ = Electric flux density vector

$\overline{H}$ = Magnetic field vector

- In linear, isotropic, non-dispersive media

$$\overline{B} = \mu \overline{H} \qquad \overline{D} = \varepsilon \overline{E}$$

$$\mu = \text{magnetic permeability} \qquad \varepsilon = \text{electric permeability}$$

Maxwell's Equations in Curl Form

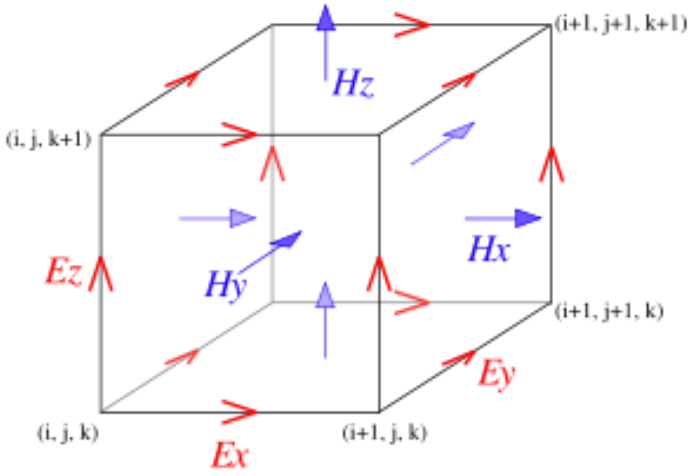
$$\frac{\partial \bar{H}}{\partial t} = \frac{1}{\mu}\left(-\nabla \times \bar{E}\right) \qquad \frac{\partial \bar{E}}{\partial t} = \frac{1}{\varepsilon}\left(\nabla \times \bar{H}\right)$$

Writing out the vector components:

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y}\right) \qquad \frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}\right)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z}\right) \qquad \frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x}\right)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial z}\right) \qquad \frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}\right)$$



- Assume E only has a z component, and that everything is constant in y:

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x}\right) \qquad \frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x}\right)$$

(image from wikipedia)

1-D FDTD

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x}\right) \qquad \frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x}\right)$$

- Apply 2nd order differencing

$$\mu \frac{H_y^{q+\frac{1}{2}}\left[m+\frac{1}{2}\right] - H_y^{q-\frac{1}{2}}\left[m+\frac{1}{2}\right]}{\Delta_t} = \frac{E_z^q[m+1] - E_z^q[m]}{\Delta_x}$$

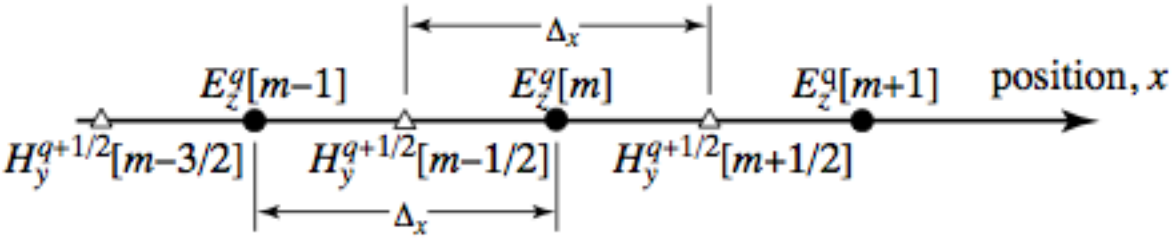
$$H_y^{q+\frac{1}{2}}\left[m+\frac{1}{2}\right] = H_y^{q-\frac{1}{2}}\left[m+\frac{1}{2}\right] + \frac{\Delta_t}{\mu\Delta_x}\left(E_z^q[m+1] - E_z^q[m]\right)$$

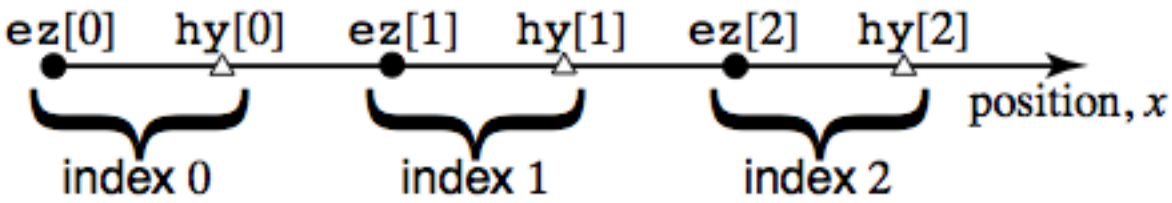
$$\epsilon \frac{E_z^{q+1}[m] - E_z^q[m]}{\Delta_t} = \frac{H_y^{q+\frac{1}{2}}\left[m+\frac{1}{2}\right] - H_y^{q+\frac{1}{2}}\left[m-\frac{1}{2}\right]}{\Delta_x}$$

$$E_z^{q+1}[m] = E_z^q[m] + \frac{\Delta_t}{\epsilon\Delta_x}\left(H_y^{q+\frac{1}{2}}\left[m+\frac{1}{2}\right] - H_y^{q+\frac{1}{2}}\left[m-\frac{1}{2}\right]\right)$$

1-D FDTD details

- Non-rigorously:
  - Energy should not propagate more than one spatial step in each temporal step
  - $\Delta x \geq \frac{1}{c}\Delta t$
- Computer implementation:





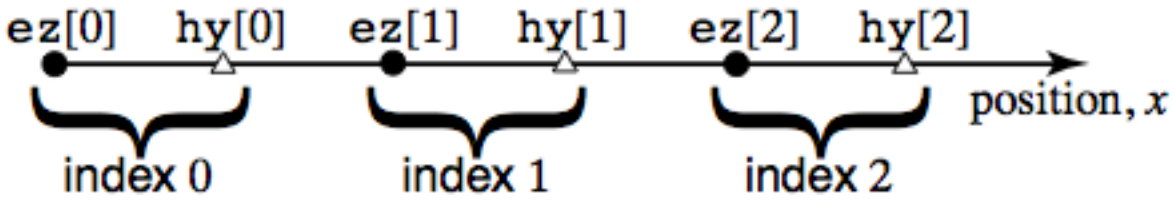
1-D FDTD Code

Computation Institute

- Define media (ca, cb)
- Initialize fields to zero
- Loop over time (n = 1 to nmax)
  - Loop over space for ez (i=0 to imax)
    - ez[i] += ca[i]*(hy[i]-hy[i-1])
  - Loop over space for hy (i=1 to imax-1)
    - hy[i] += cb[i]*(ez[i+1]-ez[i])

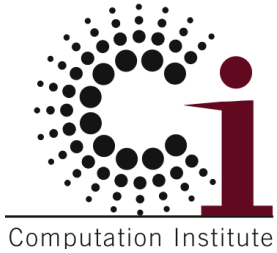
1-D FDTD Code - BC

- What about Ez[0] and Ez[imax]?
  - We need boundary conditions to ensure that waves propagate past these points without reflecting



  - Simple choice, if dt/dx=c
    - $Ez^{n}[0] = Ez^{n-1}[1]$
  - Mathematic/geometric option in 2d and 3d
    - Mur RBC (1981) – Mur RBC
  - Model absorbing material (virtual range)
    - Berenger (1994) – Berenger PML

1-D FDTD Code - Inputs



- How to input energy into the system?
  - Use a hard source
    - ez[10] = cs*sin(omega*dt*timestep)
    - Simple, but leads to reflections
  - Use a soft source
    - Ampere's Law

$$\nabla \times \mathbf{H} = \mathbf{J} + \epsilon \frac{\partial \mathbf{E}}{\partial t} \quad \Longrightarrow \quad \frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\epsilon} \nabla \times \mathbf{H} - \frac{1}{\epsilon} \mathbf{J}$$

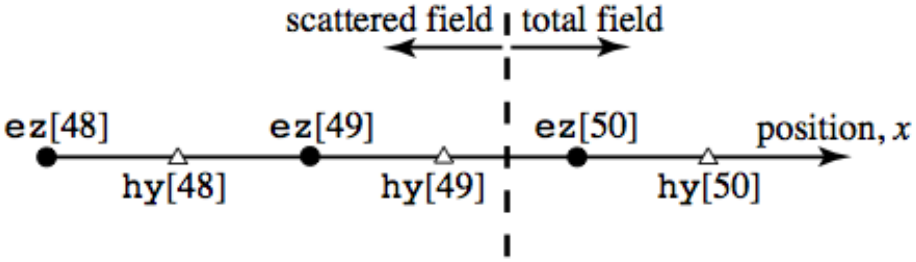
    - Apply finite differences

$$E_z^{q+1}[m] = E_z^q[m] + \frac{\Delta_t}{\epsilon \Delta_x} \left(H_y^{q+\frac{1}{2}}\left[m + \frac{1}{2}\right] - H_y^{q+\frac{1}{2}}\left[m - \frac{1}{2}\right] \right) - \frac{\Delta_t}{\epsilon} J_z^{q+\frac{1}{2}}[m]$$

    - Separate into normal update and additive source
    - ez[i] += ca[i]*(hy[i]-hy[i-1])
    - ez[10] += cs*sin(omega*dt*timestep)

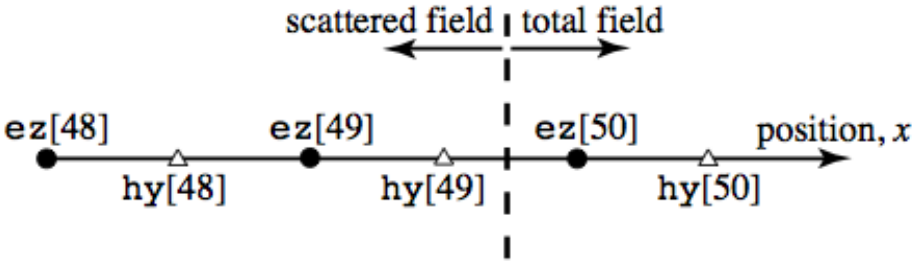
1-D FDTD Code - Scatterers

- How to find scattered field?
  - Use a total field / scattered field formulation for the main grid



  - Compute two 1-D grids, one for the incident field and one for the total/scattered field
  - Incident grid is homogeneous; TF/SF grid has scatterer geometry
  - Add/subtract incident field on total field/scattered field boundaries

1-D FDTD Code – Scatterers (2)



- $ez^{total}[50]$ += ca[50]*($hy^{total}[50]$-$hy^{total}[49]$)
  - Correct update from difference equation, but doesn't match grid
    - $hy^{total}[49] = hy^{inc}[49] + hy^{scat}[49]$
    - $ez^{total}[50]$ += ca[50]*($hy^{total}[50]$-$hy^{scat}[49]$) (normal update)
    - $ez^{total}[50]$ -= ca[50]*$hy^{inc}[49]$ (special update for TF/SF interface)
- Similar changes needed for hy[49] update, and ez and hy at TF/SF interface on right side of grid

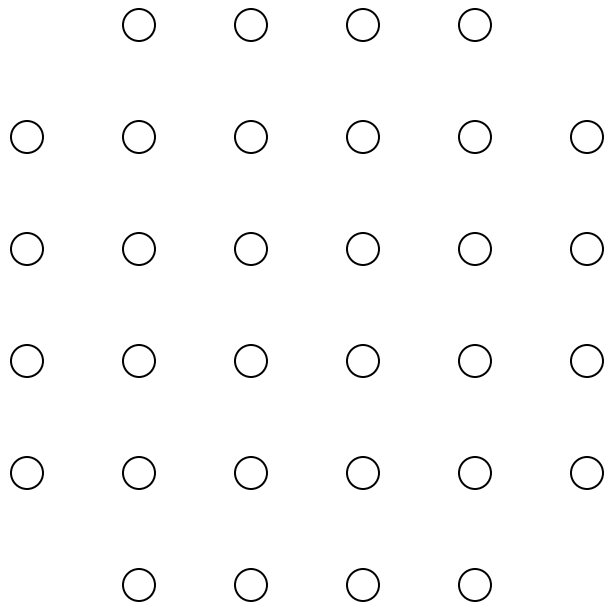
Parallel FDTD

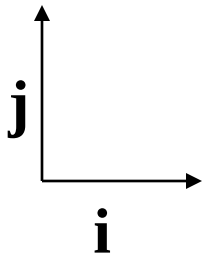
Computation Institute

- Try to use: 286-based hypercube from Intel
- Spring 1987-88
- We had 16 nodes (iPSC/d4)
- Used isend and irecv call to communicate data from one node to another
- Had previously vectorized code, and also used shared-memory parallelism (now OpenMP)

Photo from Paul Pierce (http://www.piercefuller.com/collect/other.html)

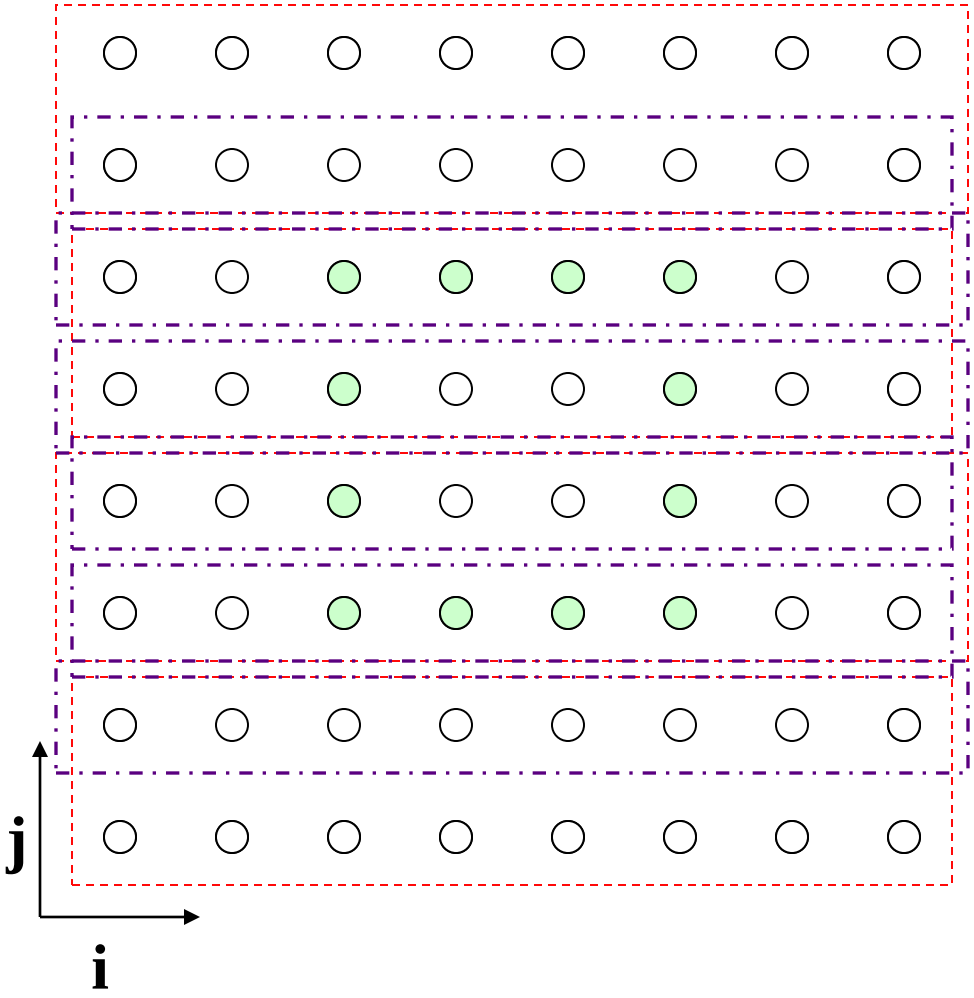
Ghost Cells (2D)





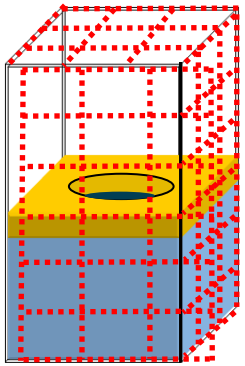
- Parallel Implementation
- Need to update these cells on a given processor, using second order central differences (one cell on each side)
- In order to update outer cells, need cells one step further away
- These have to be communicated from neighboring processors

Load Balancing



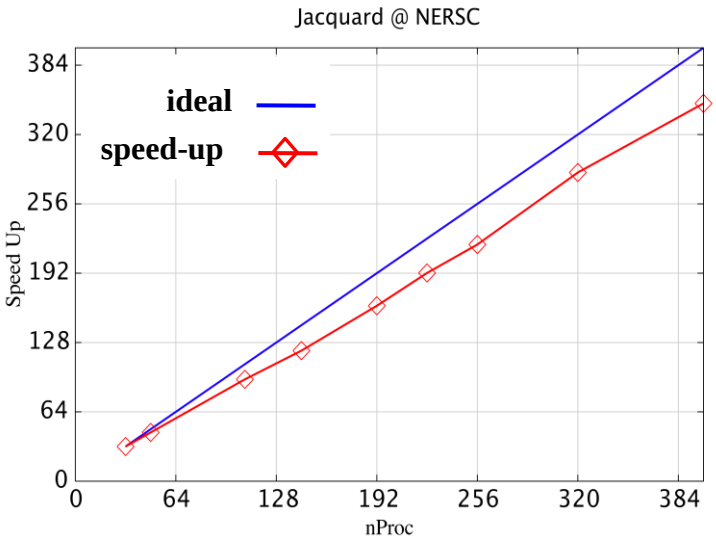
- How to divide this domain for 4 procs?
- MPI: worry about
  - Memory
  - Work
  - Communication

Parallel FDTD Modeling Example: Periodic Plasmonic System

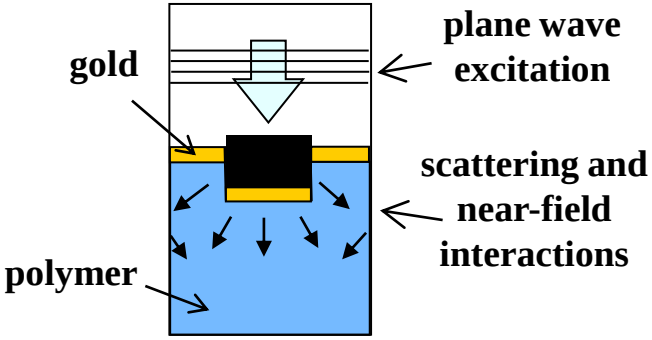


wraparound boundary conditions for side domain walls. PML for top and bottom boundary

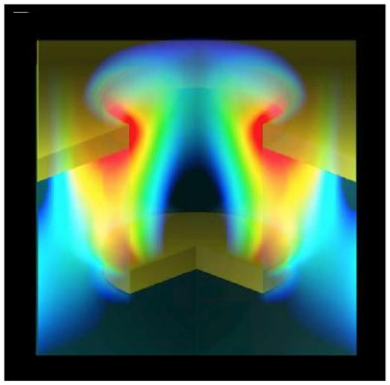
3D FDTD domain of unit cell and domain decomposition



strong scaling result (overall domain size: 262 × 262 × 1040 grid cells)



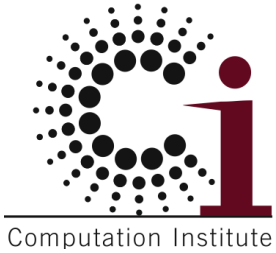
cross-sectional view



Result: 3D intensity distribution (front quarter section is cut out to show inner gold structure)

Credit: Tae-Woo Lee

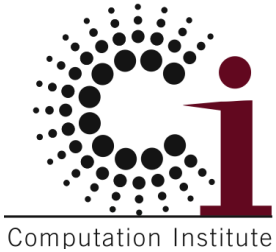
FDTD Summary



- Series of loops over components in time stepping loop
- Simple idea, complex in coding
- Fixed-side physical domain
- Usage model – set up simulation, run it, then examine output data
- Domain decomposition leads to static mapping to processors
- Tightly-coupled (alternating computation/communication)
- Load balancing is complex in practice

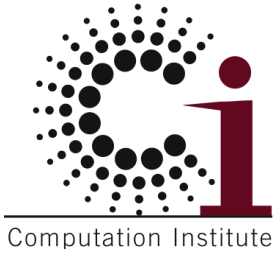
- Common to use MPI now

Outline



- Electromagnetics (FDTD): Sequential -> Parallel
- Astronomy (Montage): Parallel -> Grid
- Clouds

Montage



- An astronomical image mosaic service for the National Virtual Observatory
- Project web site - http://montage.ipac.caltech.edu/
- Core team at JPL (NASA's Jet Propulsion Laboratory) and Caltech (IPAC - Infrared Processing and Analysis Center, CACR - Center for Advance Computing Research)
- Grid architecture developed in collaboration with ISI - Information Sciences Institute

- Attila Bergou - *JPL*
- Nathaniel Anagnostou - *IPAC*
- Bruce Berriman - *IPAC*
- Ewa Deelman - *ISI*
- John Good - *IPAC*
- Joseph C. Jacob - *JPL*
- Daniel S. Katz - *JPL*
- Carl Kesselman - *ISI*
- Anastasia Laity - *IPAC*
- Thomas Prince - *Caltech*
- Gurmeet Singh - *ISI*
- Mei-Hui Su - *ISI*
- Roy Williams - *CACR*

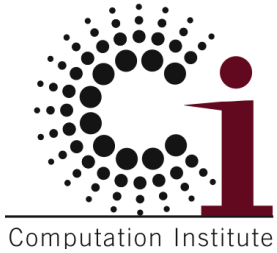
What is Montage?

Computation Institute

- Delivers custom, science grade image mosaics
  - An image mosaic is a combination of many images containing individual pixel data so that they appear to be a single image from a single telescope or spacecraft
  - User specifies projection, coordinates, spatial sampling, mosaic size, image rotation
  - Preserve astrometry (to 0.1 pixels) & flux (to 0.1%)
- Modular, portable “toolbox” design
  - Loosely-coupled engines for image reprojection, background rectification, co-addition
    - Control testing and maintenance costs
    - Flexibility; e.g custom background algorithm; use as a reprojection and co-registration engine
  - Each engine is an executable compiled from ANSI C
- Public service deployed
  - Order mosaics through web portal

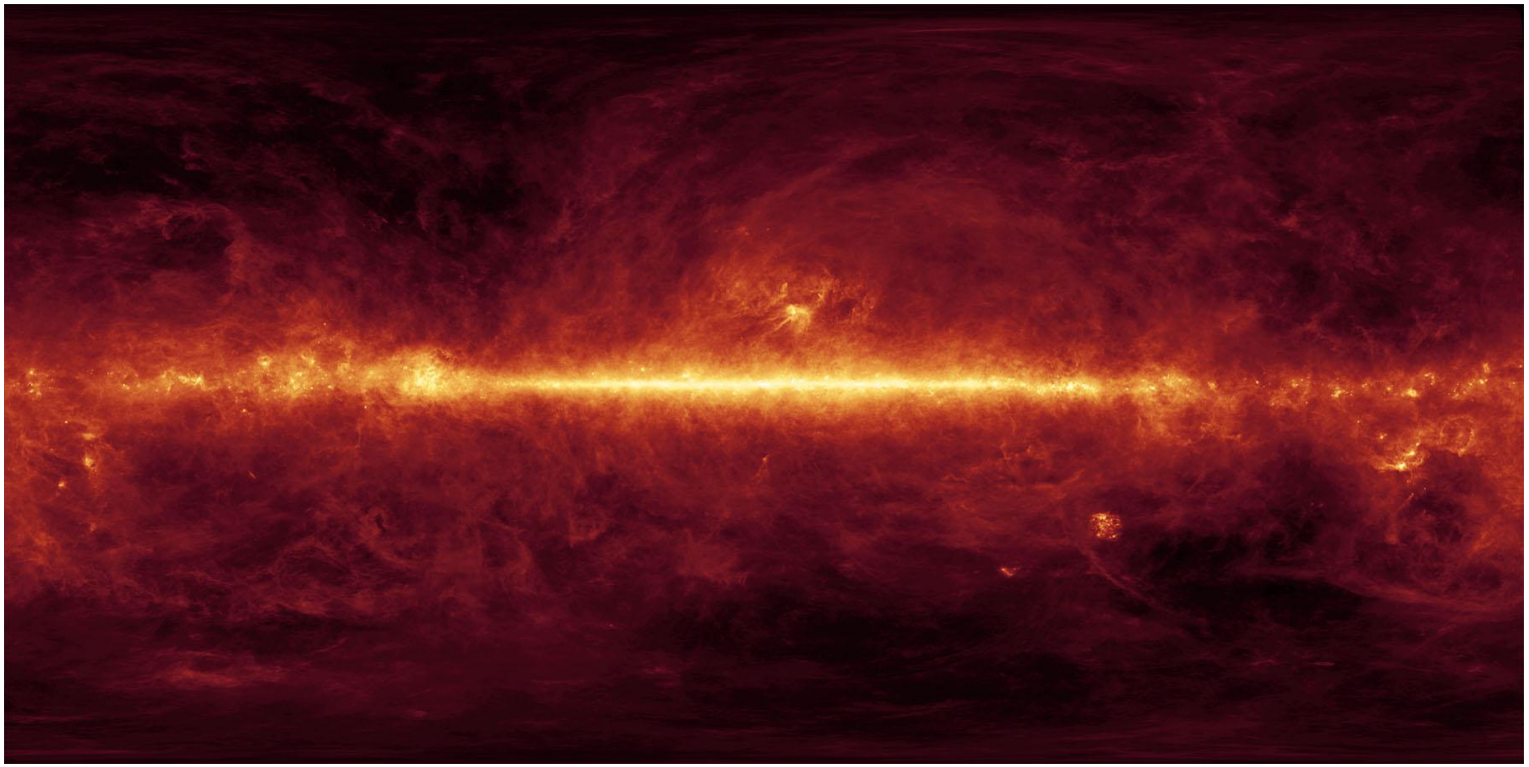
David Hockney *Pearblossom Highway* 1986

Use of Montage

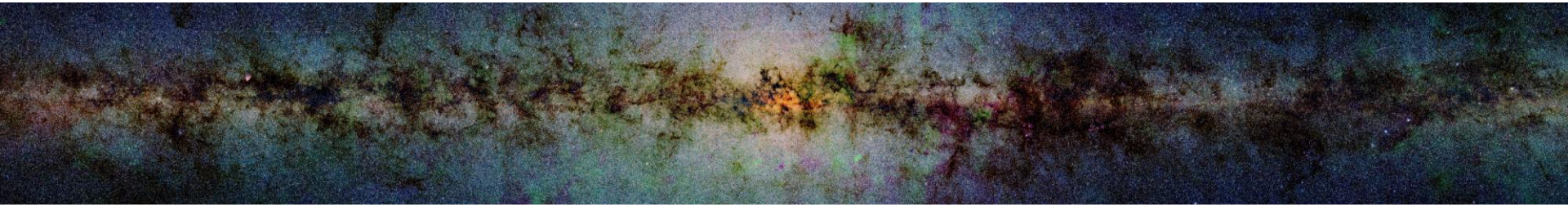


- Scientific Use Cases
  - Structures in the sky are usually larger than individual images
  - High signal-to-noise images for studies of faint sources
  - Multiwavelength image federation
    - Images at different wavelengths have differing parameters (coordinates, projections, spatial samplings, . . .)
    - Place multiwavelength images on common set of image parameters to support faint source extraction

Montage Use by Spitzer E/PO Group

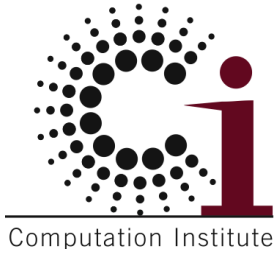


100 µm sky; aggregation of COBE and IRAS maps (Schlegel, Finkbeiner and Davis, 1998) 360˚ x 180˚, CAR projection



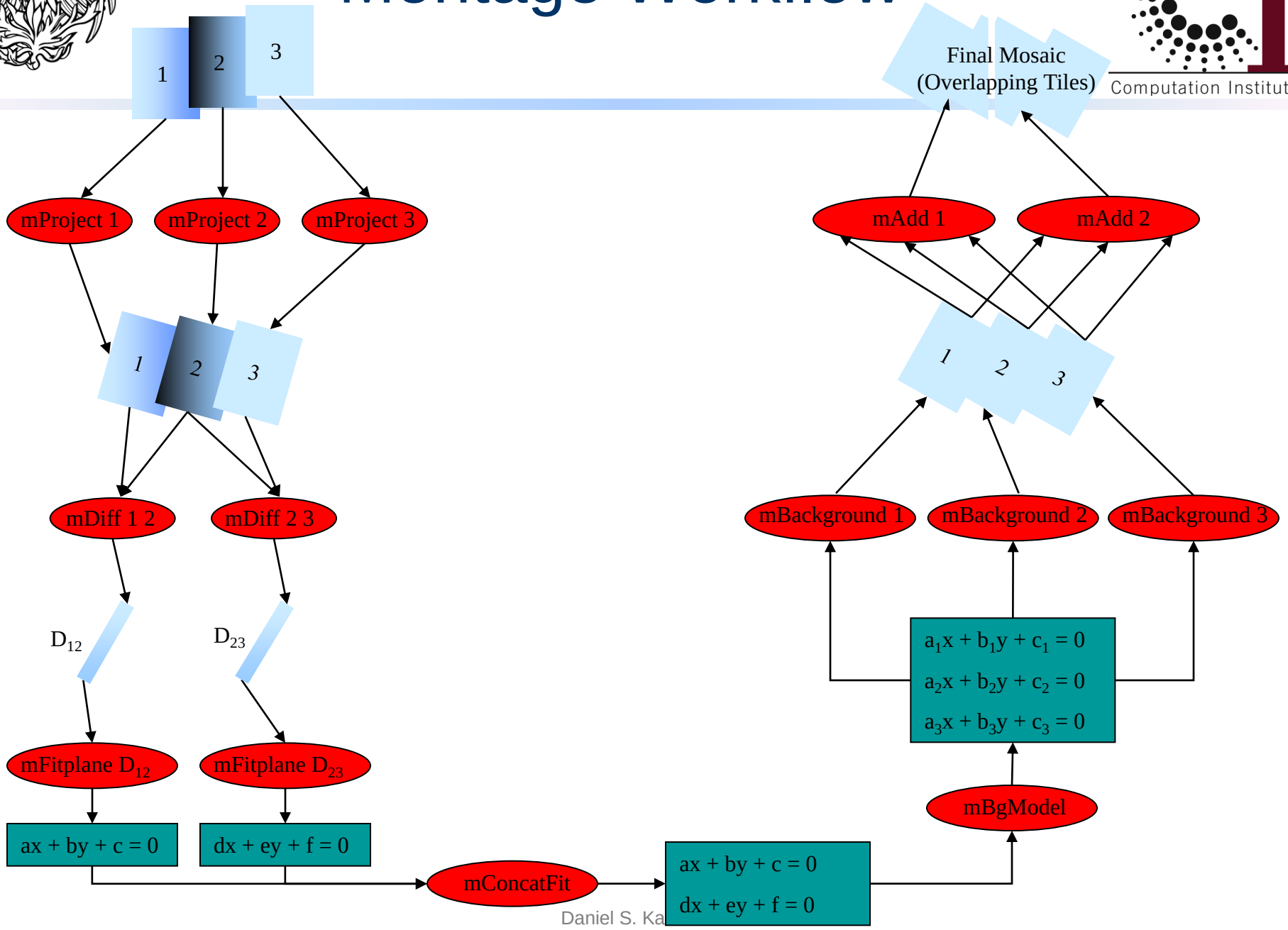
Panoramic view of galactic plane as seen by 2MASS, 44˚ x 8˚, 158,400 x 28,800 pixels; covers 0.8% of sky

Montage Versions

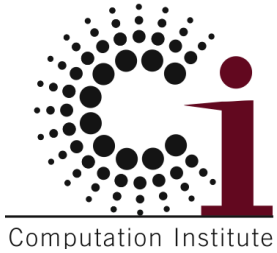


- Montage version 1.0 emphasized accuracy in photometry and astrometry
  - Images processed serially
  - Extensively tested and validated on 2MASS 2IDR images on Red Hat Linux 8.0 (Kernel release 2.4.18-14) on a 32-bit processor
- Montage version 2.2
  - More efficient reprojection algorithm: up to 30x speedup
  - Improved memory efficiency: capable of building larger mosaics
  - Enabled for parallel computation with MPI
  - Enabled for processing on TeraGrid using standard grid tools
- Montage version 3.0
  - Utilities and bug fixes
- Code and User's Guide available for download at *http://montage.ipac.caltech.edu/*

Montage Workflow



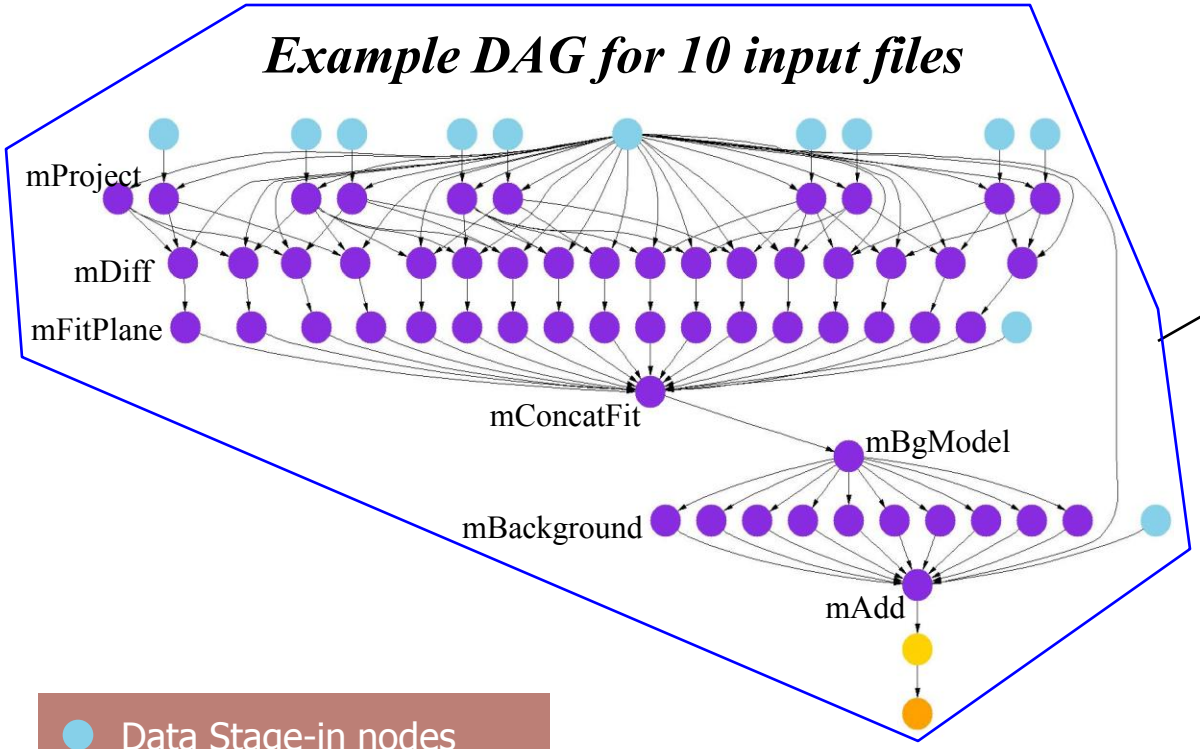
Montage on a Grid



- "Grid" is an abstraction
  - Array of processors, grid of clusters, …
- Use a methodology for running on any "grid environment"
  - Exploit Montage's modular design in an approach applicable to any grid environment
    - Describe flow of data and processing (in a Directed Acyclic Graph - DAG), including:
      - Which data are needed by which part of the job
      - What is to be run and when
  - Use standard grid tools to exploit the parallelization inherent in the Montage design
- Build an architecture for ordering a mosaic through a web portal
  - Request can be processed on a grid
- This is just one example of how Montage could run on a grid

Montage on the Grid Using Pegasus

Computation Institute



Maps an abstract workflow to an executable form

Pegasus

http://pegasus.isi.edu/

Grid Information Systems

Information about available resources, data location

Condor DAGMan

Executes the workflow

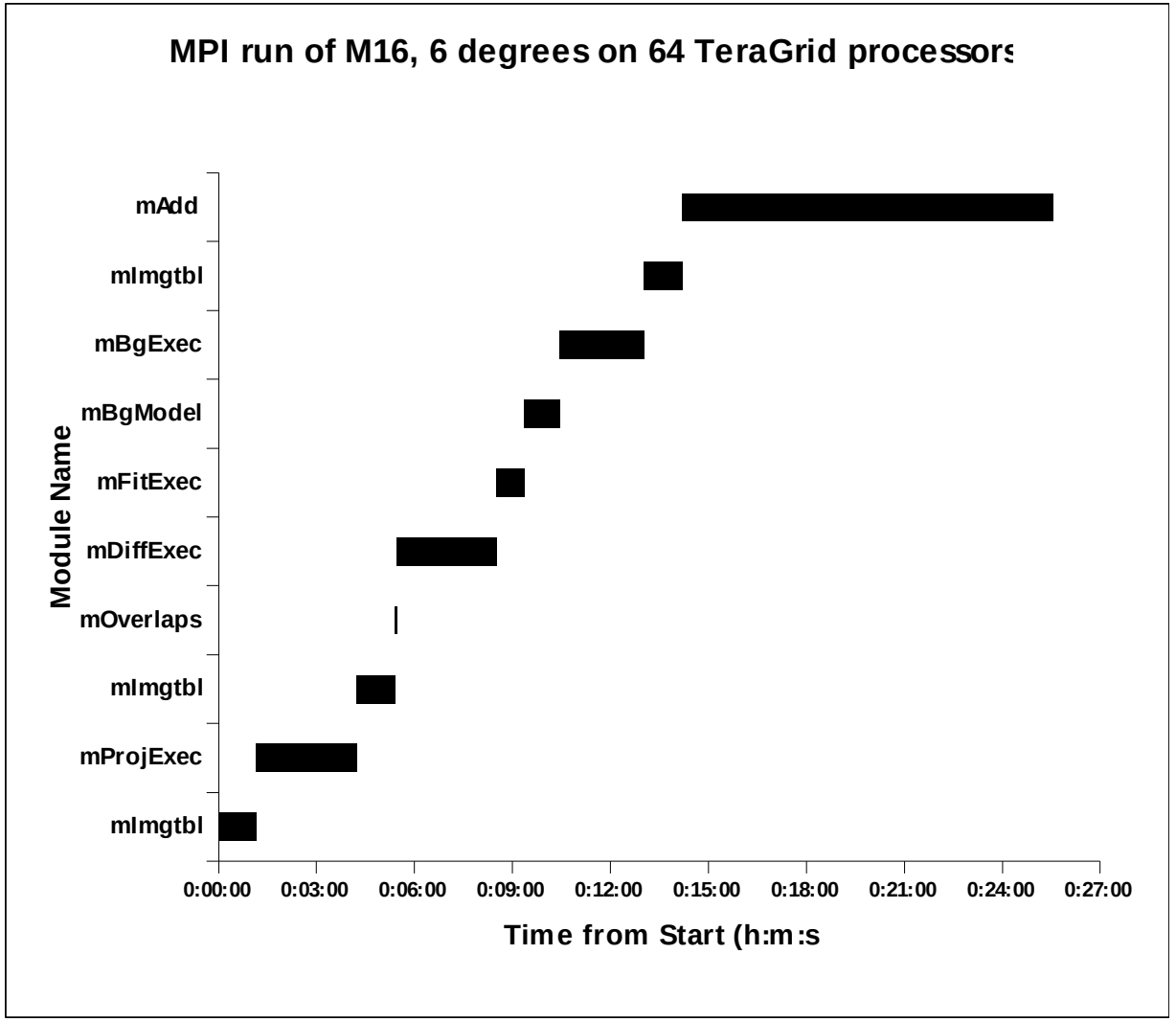
MyProxy

Grid

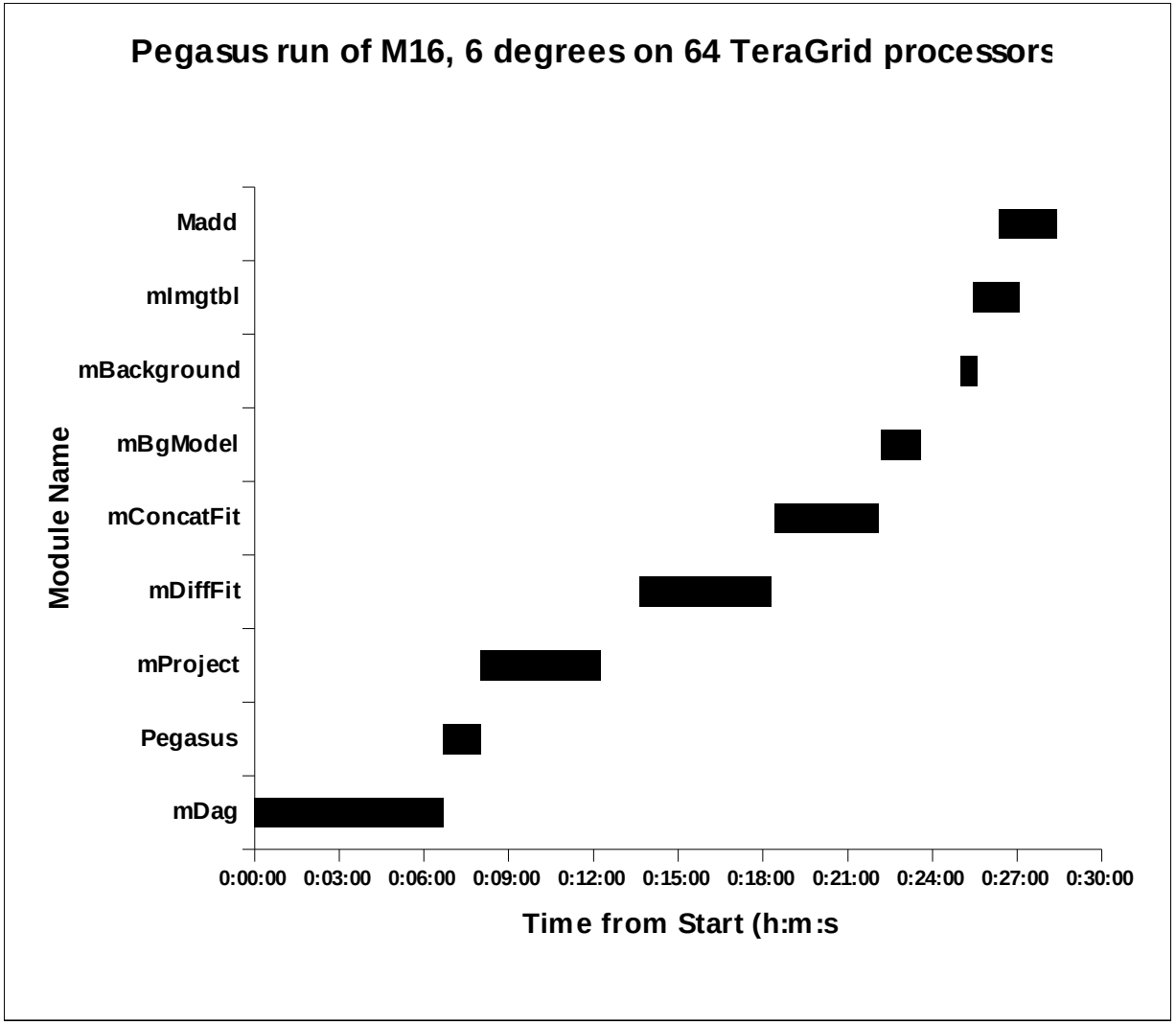
User's grid credentials

- Data Stage-in nodes
- Montage compute nodes
- Data stage-out nodes
- Registration nodes

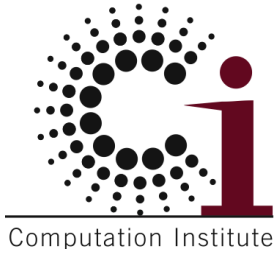
Montage Performance on Large Problem



Montage Performance on Large Problem

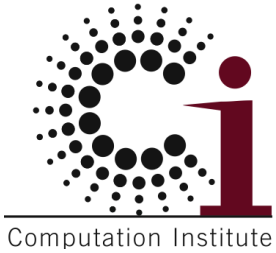


Timing Discussion



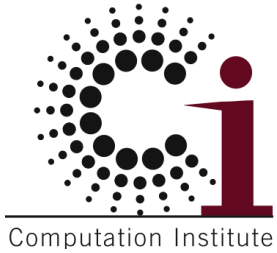
- Both MPI and Pegasus timings ignore time to start job (queuing delay)
  - MPI - script is placed in queue
  - Pegasus - Condor Glide-in is used to allow single processor jobs to work on pool
    - For efficiency, jobs are clustered and each cluster is submitted to the pool
    - Condor overhead for each item submitted is between 1 and 5 seconds
- Tasks are different
  - MPI - mImgtbl, mProjExec, mImgtbl, mOverlaps, mDiffExec, mFitExec, mBgModel, mBgExec, mImgtbl, mAdd
    - *Exec tasks are parallel tasks, others are sequential
    - Flow is dynamic, based on resulting files from previous stages
  - Pegasus - mDag, Pegasus, mProject(s), mDiffFit(s), mConcatFit, mBgModel, mBackground(s), mImgtbl, mAdd
    - *(s) tasks are multiple, clustered by Pegasus/Condor
    - Flow is fixed, based on output of mDag
- Gaps between tasks - not important, tasks are long compared to gaps
- Accuracy is very uncertain, as the parallel file system is being hit harder
- Overall
  - MPI - job finishes in 00:25:33
  - Pegasus - job finishes in 00:28:25

Newer Montage Work



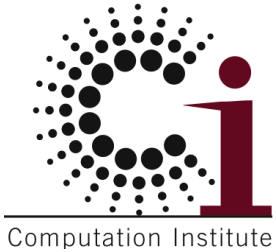
- C. Hoffa, G. Mehta, E. Deelman, T. Freeman, K. Keahey, B. Berriman, J. Good, “On the Use of Cloud Computing for Scientific Workflows,” SWBES08: Challenging Issues in Workflow Applications, 2008
  - Ran Montage on virtual and physical machines, including a private cloud-like system
- Montage used as prototype application by teams involved in ASKALON, QoS-enabled GridFTP, SWIFT, SCALEA-G, VGrADS, etc.

Montage Summary



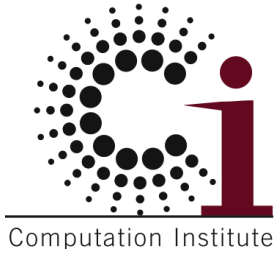
- Montage is a custom astronomical image mosaicking service that emphasizes astrometric and photometric accuracy
- Public release, available for download at the Montage website: http://montage.ipac.caltech.edu/
- MPI version of Montage:
  - Best performance
  - Requires a set of processors with a shared file system
- Pegasus/DAGman version of Montage:
  - Almost equivalent performance for large problems
  - Built-in fault tolerance
  - Can use multiple sets of processors
- Grid version works: flexible, efficient
- Local usage is still easier, mixed mode is common
  - Some processors local for known work, grid for excess/unknown work

Outline



- Electromagnetics (FDTD): Sequential -> Parallel
- Astronomy (Montage): Parallel -> Grid
- Clouds

Clusters, Grid, and Clouds

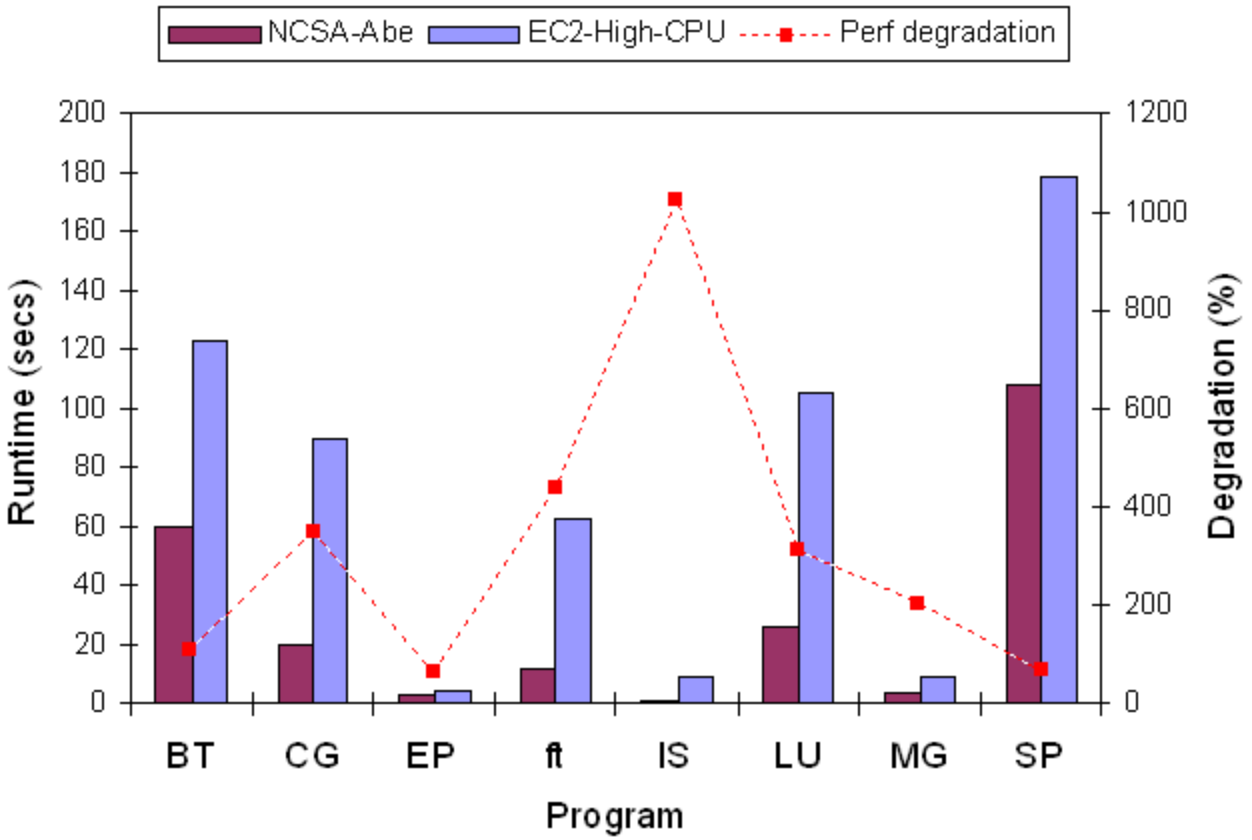


	Cluster	Grid	Cloud
Queue	yes	yes	no
(Resources	scarce	scarce	abundant)
Coupling	tight	loose	loose
Dynamic	no	no/yes	yes (but no?)
OS/tools	physical	physical	virtual

Not clear that these are intrinsic to clouds, but seem to be correct for current commercial clouds, such as Amazon EC2; maybe different for private clouds (w/ Eucalyptus, Nimbus, etc.)

MPI benchmarks on Clouds

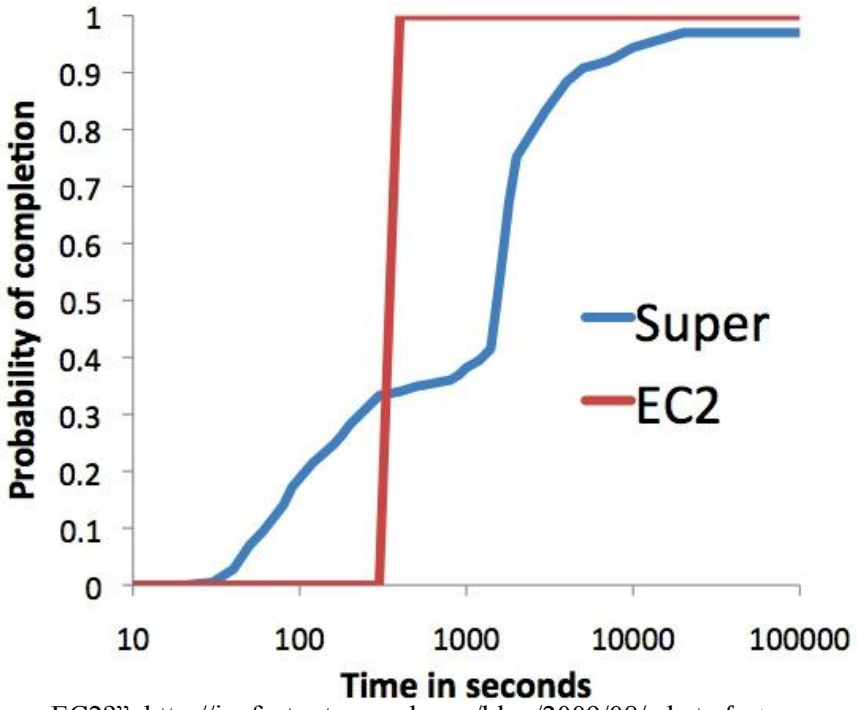
- NAS Parallel Benchmarks, MPI, Class B



E. Walker, "Benchmarking Amazon EC2 for High-Performance Scientific Computing," *;login:*, 2008.

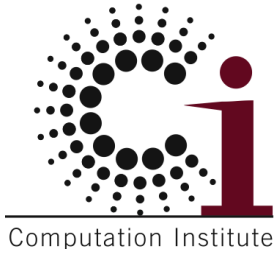
What about Queues

- Prediction for completion of LU (runtime = 25 sec on cluster, 100 sec on EC2)
- Queue time = ?? on cluster, 300 sec on EC2



I. Foster, "What's faster--a supercomputer or EC2?", http://ianfoster.typepad.com/blog/2009/08/whats-fastera-supercomputer-or-ec2.html, August 5, 2009

NSF Clouds



- FY08 – Cluster Exploratory (CluE) program: cloud-based software services supported by Google and IBM
  - Linux, Hadoop, PaaS
- and access to another cluster supported by HP, Intel, and Yahoo housed at the University of Illinois at Urbana-Champaign
  - Linux, Hadoop, IaaS & PaaS
- FY09 – Data-intensive Computing Program: explore new ways to design, develop, use, and evaluate large cluster platforms and systems, especially to support data-intensive applications that require very large-scale clusters
- FY10 – Access to Windows Azure platform
  - Windows, Azure, PaaS

DOE Magellan: Where do clouds fit?

Computation Institute

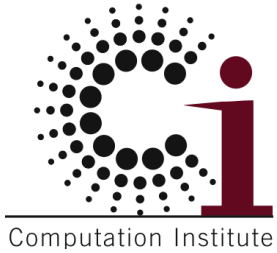
- Extreme-scale platforms fit extreme-scale problems
- Need a handful of nodes? That small cluster down the hall is perfect
- What about the mid-range?
  - Unique and customized software stacks?
  - Data-intensive computing?
  - Infrequent big runs

This supercomputer is too big!

This cluster is too small!

Credit: Pete Beckman, ANL

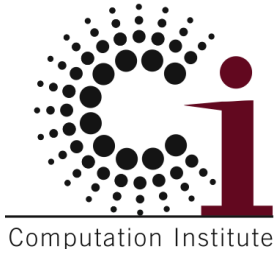
Unique Characteristics of ALCF Magellan



- High-speed, low-latency interconnect
  - QDR Infiniband connection to all nodes
- High-performance storage
  - Solid-state storage
  - High-performance parallel file system
- High-bandwidth wide area networking
  - Direct connection to 20-Gbps ESnet, eventually 100-Gbps
- Tuned middleware and scientific software

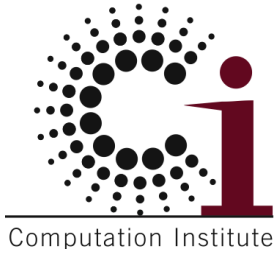
Credit: Pete Beckman, ANL

MPI Clouds



- Penguin on Demand
  - “HPC as a Service”
  - A virtualized, scalable cluster available on demand that operates and has the same performance characteristics as a physical HPC cluster located in a machine room
    - Tries to group processes to take advantage of interconnects
  - Includes support with HPC expertise
- SGI Cyclone
  - HPC as a Service
  - Either SaaS (technical apps/support) or IaaS (clusters w/ accelerators)

Montage Cloud Challenges



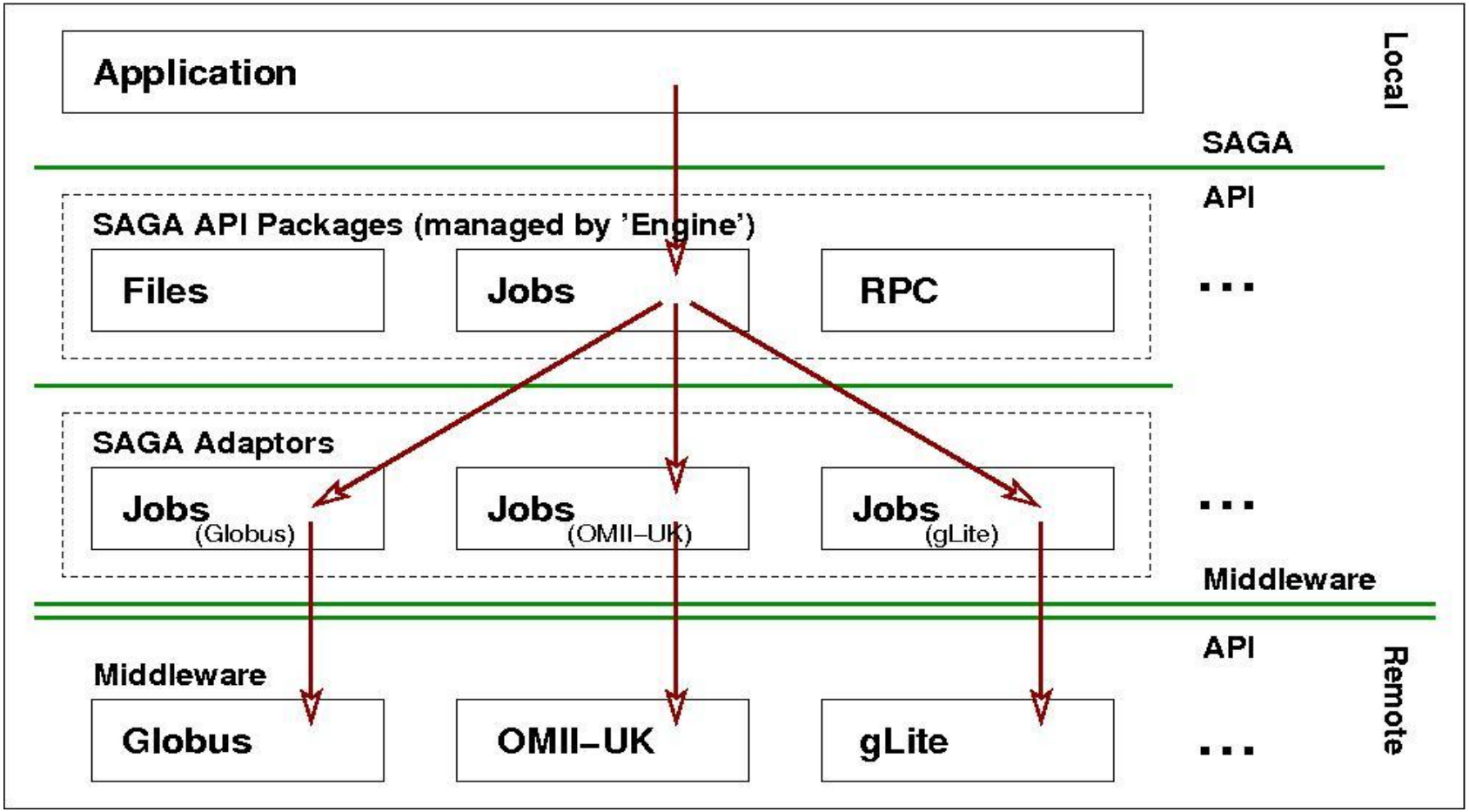
- Implementation and tools are not general
  - Development - could have been simpler
    - mDAG is not a simple code
    - Could have used Pegasus DAX API, but didn't seem any simpler
    - No way to make runtime decisions based on data
  - Deployment and Execution
    - Want to use other infrastructures, such as clouds
    - Want to make runtime decisions based on resources
      - Provide better fault tolerance than rescue DAG
    - Want to control resources (e.g., networks)
- Started looking at these – led to: A. Merzky, K. Stamou, S. Jha, D. S. Katz, "A Fresh Perspective on Developing and Executing DAG-Based Distributed Applications: A Case-Study of SAGA-based Montage," *Proceedings of 5th IEEE International Conference on e-Science*, 2009

Distributed Applications (Montage) Development Objectives

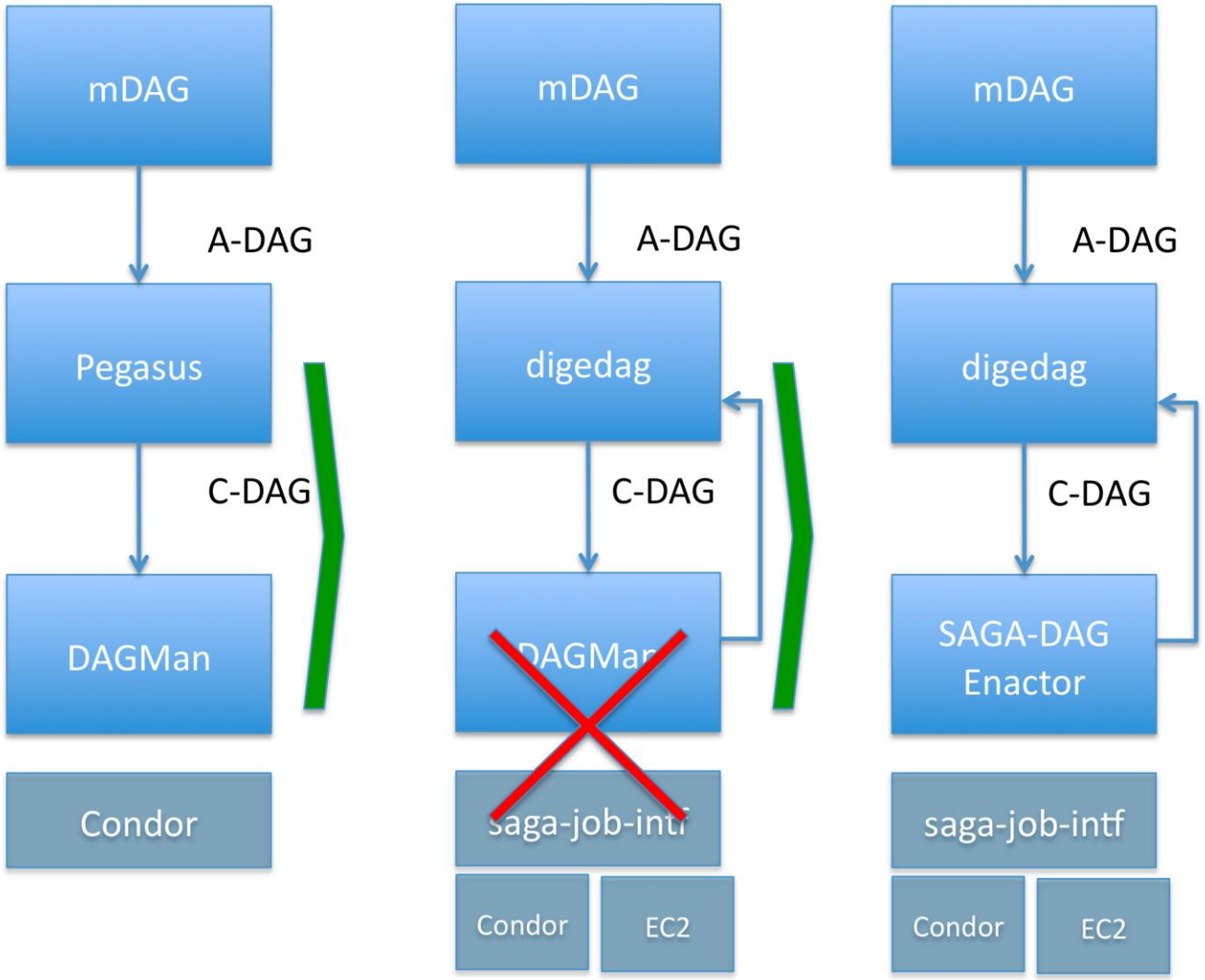
Computation Institute

- eSI theme – Distributed Programming Abstractions
  - Jha, Katz, Parashar, Rana, Weissman, “Critical Perspectives on Large-Scale Distributed Applications and Production Grids,” Proceedings of Grid 2009
  - Question: What are the main objectives for developing, deploying, and executing distributed applications?
    - Interoperability: Ability to work across multiple distributed resources
    - Distributed Scale-Out: The ability to utilize multiple distributed resources concurrently
    - Extensibility: Support new patterns/abstractions, different programming systems, functionality & Infrastructure
    - Adaptivity: Response to fluctuations in dynamic resource and availability of dynamic data
    - Simplicity: Accommodate above distributed concerns at different levels easily…
  - Potential answer to IDEAS: Frameworks, including SAGA, the Simple API for Grid Applications?
    - Use Montage to explore

SAGA: Job Submission
Role of Adaptors (middleware binding)



DAG-based Workflow Applications: Extensibility Approach

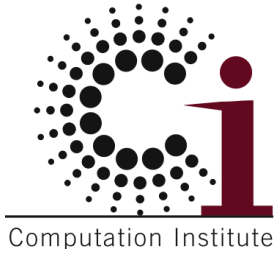


Application Development Phase

Generation & Exec. Planning Phase

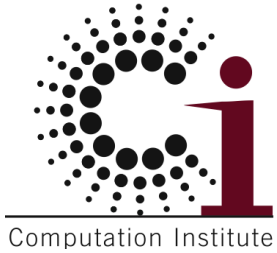
Execution Phase

digedag



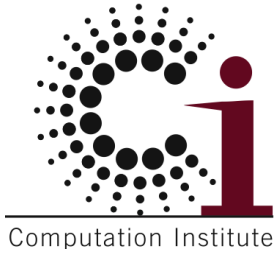
- digedag - prototype implementation of a SAGA-based workflow package, with:
  - an API for programatically expressing workflows
  - a parser for (abstract or concrete) workflow descriptions
  - an (in-time workflow) planner
  - a workflow enactor (using the SAGA engine)
    - this will eventually be separated from digedag, but will continue to use SAGA
- Can accept mDAG output, or Pegasus output
- Can move back and forth between abstract and concrete DAG

SAGA-Montage Testing



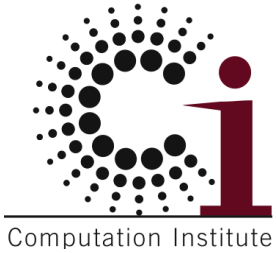
- Tests run
  - toy problem: m101 tutorial (0.2° x 0.2°)
  - But useful for trying things – functionality
- digedag used for planning
  - For this problem, takes about about 0.2 s – same as Pegasus
- Runs
  - Local submission using fork
  - Local submission using ssh/SAGA
  - Local submission using Condor/SAGA
    - Local submission using 2 of above 3 and 3 of above 3
  - Queen Bee submission using ssh/SAGA
  - EC2 submission using AWS/SAGA
    - Remote submission to Queen Bee and EC2 using both ssh/SAGA and AWS/SAGA
    - Local/remote submission to local, Queen Bee, and EC2 using fork, ssh/SAGA, and AWS/SAGA

Further Montage Cloud Work



- Goal: Develop distributed data-intensive scientific applications to utilize a broad range of distributed systems, without vendor lock-in, or disruption, yet with the flexibility and performance that scientific applications demand
  - Coordination of distributed data & computing
  - Runtime (dynamic) scheduling (including networks), placement, affinity
    - Including use of information systems – BQP on TG, etc.
  - Fault-tolerance
- Challenges
  - What are the components? How are they coupled? How is functionality expressed/exposed? How is coordination handled?
  - Layering, ordering, encapsulations of components
  - Tradeoff of costs and rewards
    - Balance user and system utility (time to solution vs. system utilization)

Conclusions



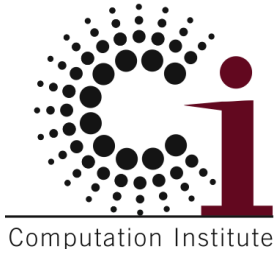
- Static (parallel) apps don't have much to gain from today's clouds
  - Emerging "HPC as a Service", and new private clouds such as Magellan might change this
  - Also research in removing VM overhead, such as pass-through communication and I/O
  - Recognize that the app may not want to change, the infrastructure has to change to support the app
- Other apps can clearly gain from today's clouds
  - Using PaaS is simple, for the apps that can be re-written to do so (DAG-based+)
- New apps can be the most flexible
  - If they throw out old assumptions
  - Use SAGA to make best use of clouds and grids together?

Conclusions (2)

Computation Institute

- NIST definition:
  - a computing capability that provides **an abstraction between the computing resource and its underlying technical architecture** (e.g., servers, storage, networks), enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction
- Applications are developed for specific underlying architectures
- What is the abstract architecture for clouds?

Credits



- FDTD material:
  - Allen Taflove, Northwestern, John Schneider, WSU, Tae-Woo Lee, LSU
- Montage
  - Attila Bergou, Nathaniel Anagnostou, Bruce Berriman, John Good, Joseph C. Jacob, Anastasia Laity, Thomas Prince, Roy Williams
- Grid Montage
  - Ewa Deelman, Carl Kesselman, Gurmeet Singh, Mei-Hui Su
- DPA Theme
  - e-Science Institute
  - Shantenu Jha, Manish Parashar, Omer Rana, Jon Weissman
- SAGA
  - SAGA Team – http://saga.cct.lsu.edu/
- SAGA Montage
  - Andre Luckow, Andre Merzky, Katerina Stamou, Shantenu Jha