GLOBAL cyberbridges

BRIDGING THE DIVIDE BETWEEN THE INFORMATION TECHNOLOGY AND SCIENCE COMMUNITIES

Workshop 03A - Global CyberBridges: A Model Global Collaboration Infrastructure for E-Science Between the United States and International Partners

Educause Learning Initiative (ELI) Annual Meeting—Orlando, FL

Date: January 20, 2009

FIU

CIARA

NSF

NSF AWARD N°. 0537464

Slides

- http://users.cs.fiu.edu/~sadjadi/Presentations/EDUCAUSE-ELI-2009-GCB/
- http://evo.caltech.edu

Agenda

- Why HPC?
- What is HPC anyway?
- Scaling OUT vs. Scaling UP!
- Example Grid Enabling Projects
- Collaboration Tools

Words of Wisdom

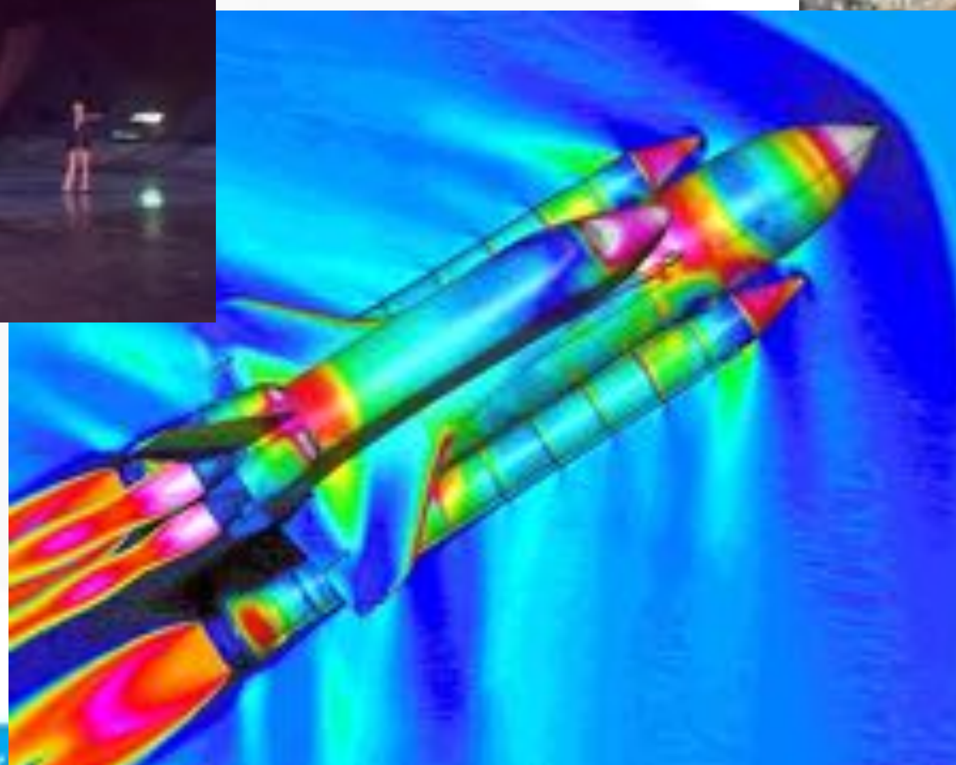
- "Four or five computers should be enough for the entire world until the year 2000."
  - T.J. Watson, Chairman of IBM, 1945.
- "640KB [of memory] ought to be enough for anybody."
  - Bill Gates, Chairman of Microsoft,1981.
- You may laugh at their vision today, but …
  - Lesson learned: Don't be too visionary and try to make things work!

Evolution of Science

- Traditional scientific and engineering disciplines:
  - Do theory or paper design
  - Perform experiments or build system
- Limitations:
  - Too difficult -- build large wind tunnels
  - Too expensive -- build a throw-away airplane
  - Too slow -- wait for climate or galactic evolution
  - Too dangerous -- weapons, drug design, climate experiments
- Solution:
  - Use HPC to simulate the phenomenon

Computational Fluid Dynamics (CFD)

Replacing NASA's Wind Tunnels with Computers



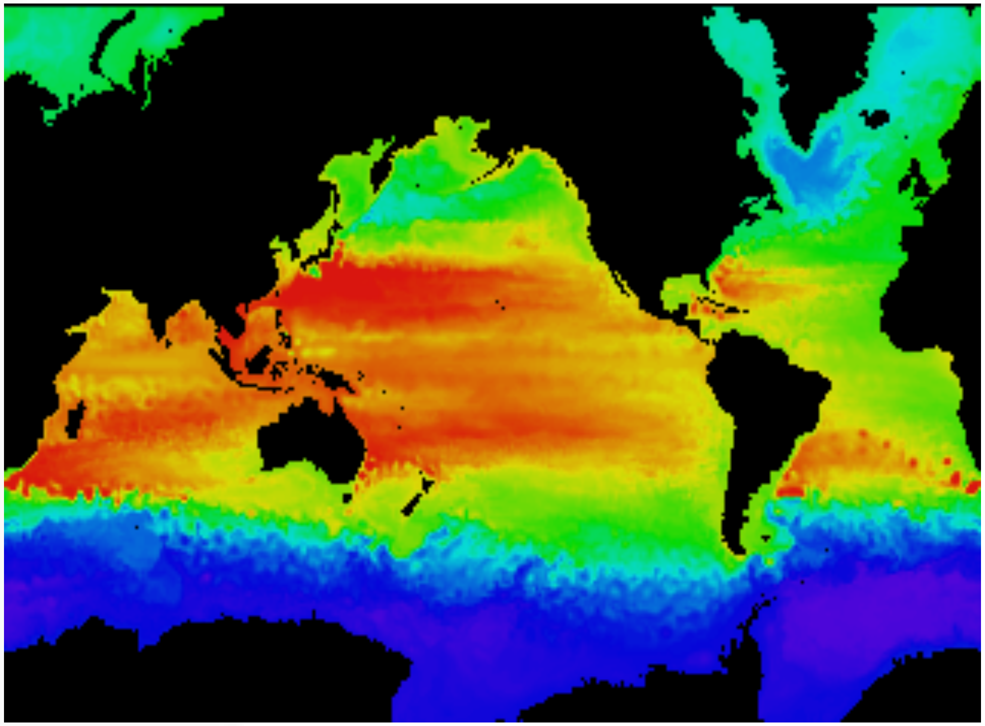
Why High-Performance Computing?

- Science
  - Global climate modeling &Hurricane Modeling
  - Astrophysical modeling
  - Biology: genomics; protein folding; drug design
  - Computational Chemistry
  - Computational Material Sciences and Nanosciences
- Engineering
  - Crash simulation
  - Semiconductor design
  - Earthquake and structural modeling
  - Computation fluid dynamics (airplane design)
  - Combustion (engine design)
- Business
  - Financial and economic modeling
  - Transaction processing, web services and search engines
- Defense
  - Nuclear weapons -- test by simulation
  - Cryptography

Global Climate

- Problem is to compute:
  f (latitude, longitude, elevation, time) →
  temperature, pressure, humidity, wind velocity

- Approach:
  - *Discretize* the domain measurements for points every 10 km, for example
  - Devise an algorithm to predict weather at time t+1 given t



Source: http://www.epm.ornl.gov/chammp/chammp.html

Global Climate Requirements

- One piece is modeling the fluid flow in the atmosphere
  - Solve Navier-Stokes problem
    - Roughly 100 Flops per grid point with 1 minute timestep
- Computational requirements:
  - To match real-time, need 5×10^{11} flops in 60 seconds = 8 Gflop/s
  - Weather prediction (7 days in 24 hours) → 56 Gflop/s
  - Climate prediction (50 years in 30 days) → 4.8 Tflop/s
  - Policy negotiations (50 years in 12 hours) → 288 Tflop/s
- Let's make it even worse!
  - To 2x grid resolution, computation is > 8x
  - State of the art models require integration of atmosphere, ocean, sea-ice, land models, plus possibly carbon cycle, geochemistry and more
- Current models are coarser than this!

Agenda

- Why HPC?
- What is HPC anyway?
- Scaling OUT vs. Scaling UP!
- Example Grid Enabling Projects
- Collaboration Tools

High Performance Computing?

- Difficult to define - it’s a moving target.
  - In 1980s:
    - a “supercomputer” was performing 100 Mega FLOPS
    - FLOPS: **FL**oating point **O**perations **P**er **S**econd
  - Today:
    - a 2G Hz desktop/laptop performs a few Giga FLOPS
    - a “supercomputer” performs tens of Tera FLOPS (Top500)
- High Performance Computing:
  - loosely an order of 1000 times more powerful than the latest desktops?
- Super Computing:
  - Computing on top 500 machines?

What is a computer?

- The term "computer" has been subject to varying interpretations over time.
  - Originally, referred to a person who performed numerical calculations (a human computer), often with the aid of a mechanical calculating device.
- A ***computer*** is a machine that manipulates data according to a list of instructions.
- A ***machine*** is any device that perform or assist in performing some work.
- ***Instructions*** are sequence of statements and/or declarations written in some human-readable computer programming language.

History of Computers!

- The history of the modern computer begins with two separate technologies
  - Automated calculation
  - Programmability
- Examples
  - 2400 BC, abacus was used.
  - In 1801, Jacquard added punched paper cards to textile loom.
  - In 1837, Babbage conceptualized and designed a fully programmable mechanical computer, “The Analytical Engine”.

Early Computers!

- Large-scale automated data processing of punched cards was performed for the U.S. Census in 1890 by ***tabulating machines*** designed by Herman Hollerith and manufactured by the Computing Tabulating Recording Corporation, which later became IBM.

- During the first half of the 20th century, many scientific computing needs were met by increasingly sophisticated analog computers, which used a direct mechanical or electrical model of the problem as a basis for computation.

Five Early Digital Computers

Computer	First operation	Place
Zuse Z3	May 1941	Germany
Atanasoff–Berry Computer	Summer 1941	USA
Colossus	December 1943 / January 1944	UK
Harvard Mark I – IBM ASCC	1944	USA
ENIAC	1944	USA
	1948	USA

The IBM Automatic Sequence Controlled Calculator (ASCC), called the Mark I by Harvard University.

Mark I was devised by Howard H. Aiken, created at IBM, and was shipped to Harvard in 1944.

Supercomputers?

- A ***supercomputer*** is a computer that is considered, or was considered at the time of its introduction, to be at the frontline in terms of processing capacity, particularly speed of calculation.
- The term "***Super Computing***" was first used by New York World newspaper in 1929 to refer to large custom-built tabulators IBM made for Columbia University.
  - ***Computation*** is a general term for any type of information processing that can be represented mathematically.
  - ***Information processing*** is the change (processing) of information in any manner detectable by an observer.

Supercomputers History!

- Supercomputers introduced in the 1960s were designed primarily by ***Seymour Cray*** at Control Data Corporation (CDC), and led the market into the 1970s until Cray left to form his own company, Cray Research.
  - The top spot in supercomputing for five years (1985–1990).

- Cray, himself, never used the word "supercomputer“; he only recognized the word "computer".

The Cray-2 was the world's fastest computer from 1985 to 1989.

The Cray-2 was a vector supercomputer made by Cray Research starting in 1985.

Supercomputer market crash!

- In the 1980s a large number of smaller competitors entered the market (in parallel to the creation of the minicomputer market a decade earlier), but many of these disappeared in the mid-1990s "supercomputer market crash".

- Supercomputers were typically ***one-of-a-kind custom designs*** produced by "traditional" companies such as IBM and HP, who had purchased many of the 1980s companies to gain their experience.

Supercomputer History!

- Note that yesterday's supercomputers have become today’s normal computers.

- CDC's early machines were simply very fast ***scalar processors***, some ten times the speed of the fastest machines offered by other companies.

- In the 1970s most supercomputers were dedicated to running a ***vector processor***, and many of the newer players developed their own such processors at a lower price to enter the market.

Scalar and Vector Processors?

- A ***processor*** is a machine that can execute computer programs.
- A ***scalar processor*** is the simplest class of computer processors that can process one data item at a time (typical data items being integers or floating point numbers).
- A ***vector processor***, by contrast, can execute a single instruction to operate simultaneously on multiple data items.
  - Analogy: scalar and vector arithmetic.

Supercomputer History!

- The early and mid-1980s saw machines with a modest number of vector processors working in parallel become the standard.
  - Typical numbers of processors were in the range of four to sixteen.

- In the later 1980s and 1990s, attention turned from vector processors to ***massive parallel processing*** systems with thousands of "ordinary" CPUs, some being off the shelf units and others being custom designs.
  - ***the attack of the killer micros.***

Supercomputer History!

- Today, parallel designs are based on "off the shelf" server-class ***microprocessors***, such as the PowerPC, Itanium, or x86-64, and most modern supercomputers are now highly-tuned ***computer clusters*** using ***commodity processors*** combined with custom interconnects.
- ***Commercial, off-the-shelf (COTS)*** is a term for software or hardware, generally technology or computer products, that are ready-made and available for sale, lease, or license to the general public.

Parallel Processing & Computer Cluster

- ***Parallel processing*** or ***parallel computing*** is the simultaneous use of more than one CPU to execute a program.
  - Note that parallel processing differs from ***multitasking***, in which a single CPU executes several programs at once.
- A ***computer cluster*** is a group of loosely coupled computers that work together closely so that in many respects they can be viewed as though they are a single computer.
  - The components of a cluster are commonly, but not always, connected to each other through fast local area networks.

Grid Computing

- ***Grid computing*** or ***grid clusters*** are a technology closely related to cluster computing.
- The key differences between grids and traditional clusters are that grids connect collections of computers which do not fully trust each other, or which are geographically dispersed.
- Grids are thus more like a computing utility than like a single computer.
- Grids typically support more heterogeneous collections than are commonly supported in clusters.

Ian Foster's Grid Checklist

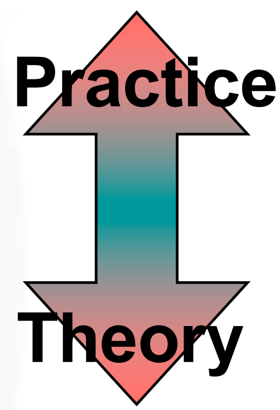
- A Grid is a system that:
  - Coordinates resources that are not subject to centralized control
  - Uses standard, open, general-purpose protocols and interfaces
  - Delivers non-trivial qualities of service

History Summary!

- 1960s: Scalar processor
  - Process one data item at a time
- 1970s: Vector processor
  - Can process an array of data items at one go
- Later 1980s: Massively Parallel Processing (MPP)
  - Up to thousands of processors, each with its own memory and OS
- Later 1990s: Cluster
  - Not a new term itself, but renewed interests
  - Connecting stand-alone computers with high-speed network
- Later 1990s: Grid
  - Tackle collaboration; Draw an analogue from Power grid

High Performance Computing

- What should we care about?
  - "How do we make computers to compute ***bigger*** problems ***faster***?"
- Three main issues
  - Hardware: How do we build faster computers?
  - Software: How do we write faster programs?
  - Hardware and Software: How do they interact?
- Many perspectives
  - architecture
  - systems
  - programming
  - modeling and analysis
  - simulation
  - algorithms and complexity



Agenda

- Why HPC?
- What is HPC anyway?
- Scaling OUT vs. Scaling UP!
- Example Grid Enabling Projects
- Collaboration Tools

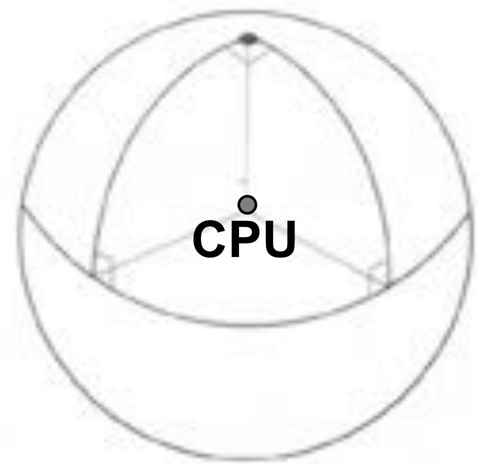
Parallelism & Parallel Computing

- The key techniques for making computers compute "bigger problems faster" is to use multiple computers at once
  - Why?
- This is called parallelism
  - It takes 1000 hours for this program to run on one computer!
    - Well, if I use 100 computers maybe it will take only 10 hours?!
  - This computer can only handle a dataset that's 2GB!
    - If I use 100 computers I can deal with a 200GB dataset?!
- Different flavors of parallel computing
  - shared-memory parallelism
  - distributed-memory parallelism
  - hybrid parallelism

Let's try to build a 10 TFlop/s CPU?

- Question?
  - Can we build a single CPU that delivers 10,000 billion floating point operations per second (10 TFlops), and operates over 10,000 billion bytes (10 TByte)?
  - Representative of what many scientists need today.
- Assumptions
  - data travels from MEM to CPU at the speed of light
  - CPU is an "ideal" sphere
  - CPU issues one instruction per cycle
    - The clock rate must be 10,000GHz
  - Each instruction will need 8 bytes of mem

– The distance between the memory and the CPU must be $r < c / 10^{13} \sim 3x10^{-6}$ m



Let's try to build a 10 TFlop/s CPU?

- Then we must have 10^{13} bytes of memory in
  - $4/3\pi r^3 = 3.7e^{-17}$ m^3
- Therefore, each word of memory must occupy
  - $3.7e^{-30}$ m^3
- This is 3.7 $Angstrom^3$
- Or the volume of a very small molecule that consists of only a few atoms

- Current memory densities are $10GB/cm^3$,
  - or about a factor 10^{20} from what would be needed!

- **Conclusion**: It's not going to happen until some scientific breakthrough happens $\rightarrow$ Cluster & Grid Computing

HPC Related Technologies

1. *Computer architecture*
   – CPU, memory, VLSI
2. *Compilers*
   – Identify inefficient implementations
   – Make use of the characteristics of the computer architecture
   – Choose suitable compiler for a certain architecture
3. *Algorithms*
   – For parallel and distributed systems
   – How to program on parallel and distributed systems
4. *Middleware*
   – Grid computing technology
   – Application $\rightarrow$ middleware $\rightarrow$ operating system
   – Resource discovery and sharing

Many connected “areas”

- Computer architecture
- Networking
- Operating Systems
- Scientific Computing
- Theory of Distributed Systems
- Theory of Algorithms and Complexity
- Scheduling
- Internetworking
- Programming Languages
- Distributed Systems
- High Performance Computing

Units of Measure in HPC

- High Performance Computing (HPC) units are:
  - Flops: floating point operations
  - Flop/s: floating point operations per second
  - Bytes: size of data (double precision floating point number is 8)

- Typical sizes are millions, billions, trillions…

Mega	Mflop/s = 10^6 flop/sec	Mbyte = 10^6 byte (also 2^{20} = 1048576)
Giga	Gflop/s = 10^9 flop/sec	Gbyte = 10^9 byte (also 2^{30} = 1073741824)
Tera	Tflop/s = 10^{12} flop/sec	Tbyte = 10^{12} byte (also 2^{40} = 10995211627776)
Peta	Pflop/s = 10^{15} flop/sec	Pbyte = 10^{15} byte (also 2^{50} = 1125899906842624)
Exa	Eflop/s = 10^{18} flop/sec	Ebyte = 10^{18} byte

Metric Units

- The principal metric prefixes.

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	Kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	Mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

Agenda

- Why HPC?
- What is HPC anyway?
- Scaling OUT vs. Scaling UP!
- Example Grid Enabling Projects
- Collaboration Tools

Hurricane Mitigation Project

- **FIU:**
  - Masoud Sadjadi, Steve Luis, Hugh Willoughby, Ping Zhu, Selim Kalayci, Juan Carlos Martinez, David Villegas, Javier Delgado, Javier Figueroa, Marlon Bright, and others
- **CNIC CAS (China):**
  - Wendy Zhao and Yunxia Hao
- **IBM T. J. Watson:**
  - Liana Fong, Norman Babroff, and Grace Yanbin
- **IBM IRL:** Gargi Dasgupta and Balaji Viswanathan
- **BSC:** Rosa Badia, Julita Corbalan, Ivan Rodero
- **SDSU:** Onyeka Ezenwoye **UNF:** Pat Welsh

HURRICANE KATRINA
MOST DESTRUCTIVE HURRICANE EVER TO STRIKE THE U.S.

On August 28, 2005, Hurricane Katrina was in the Gulf of Mexico, powered up to a Category 5 storm, packing winds estimated at 175 mph.

Do We Need More Accuracy?!

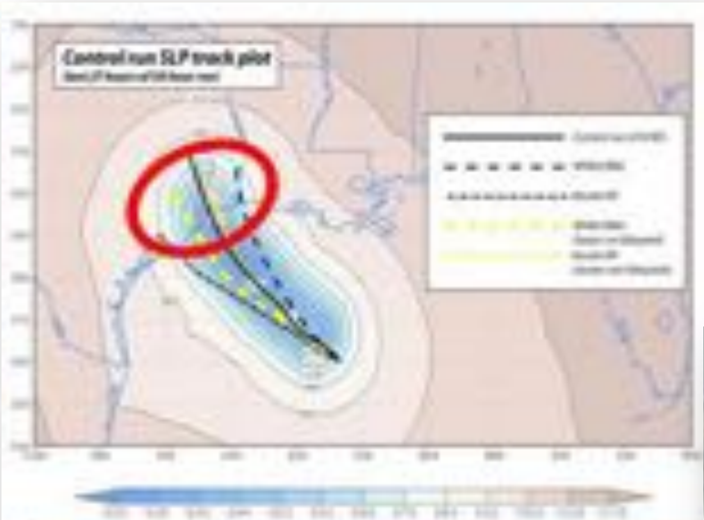


Image Source: http://mls.jpl.nasa.gov

Ike, Cuba 2008

Hurricane Andrew, Florida 1992

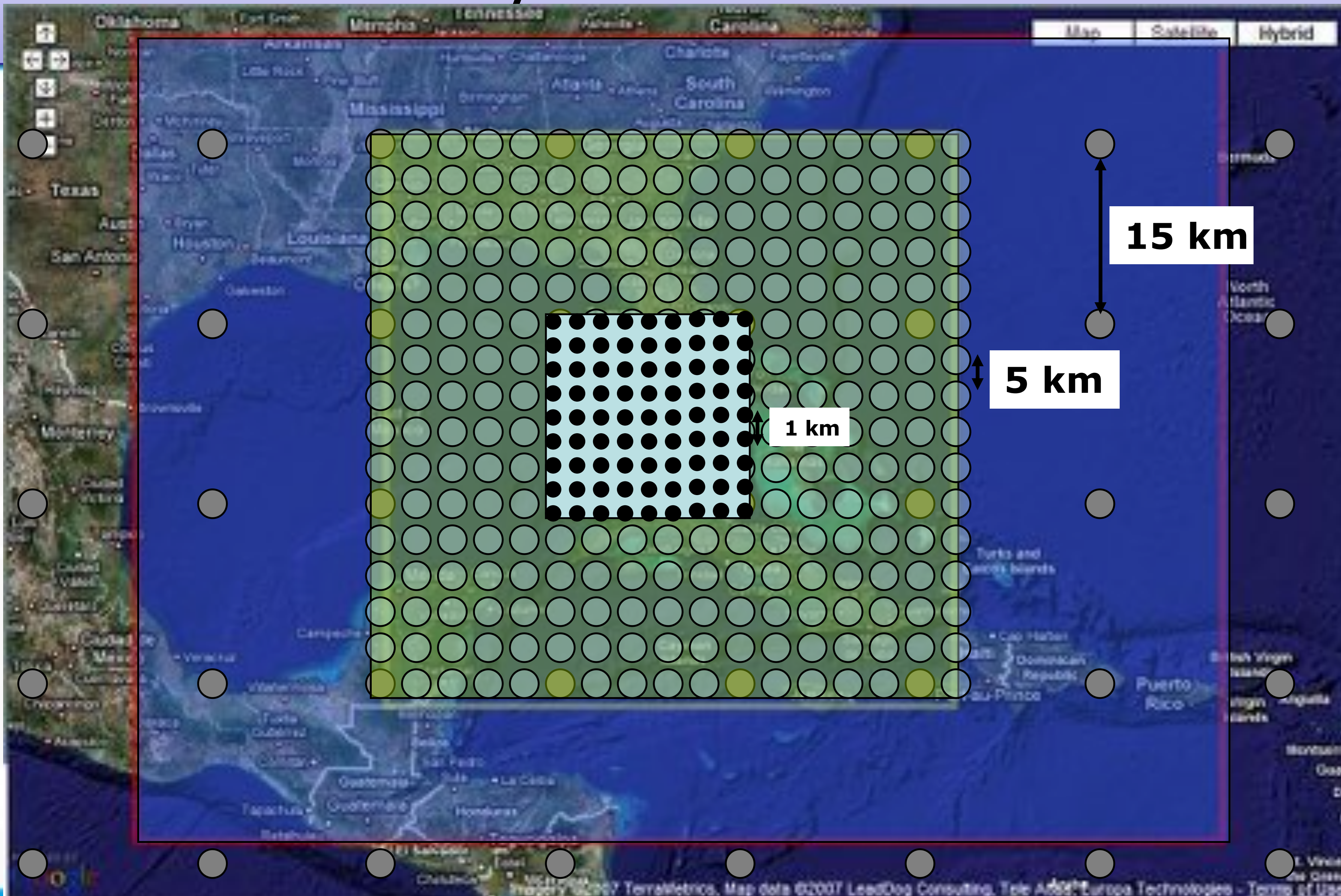
Katrina, New Orleans 2005

Project Goals

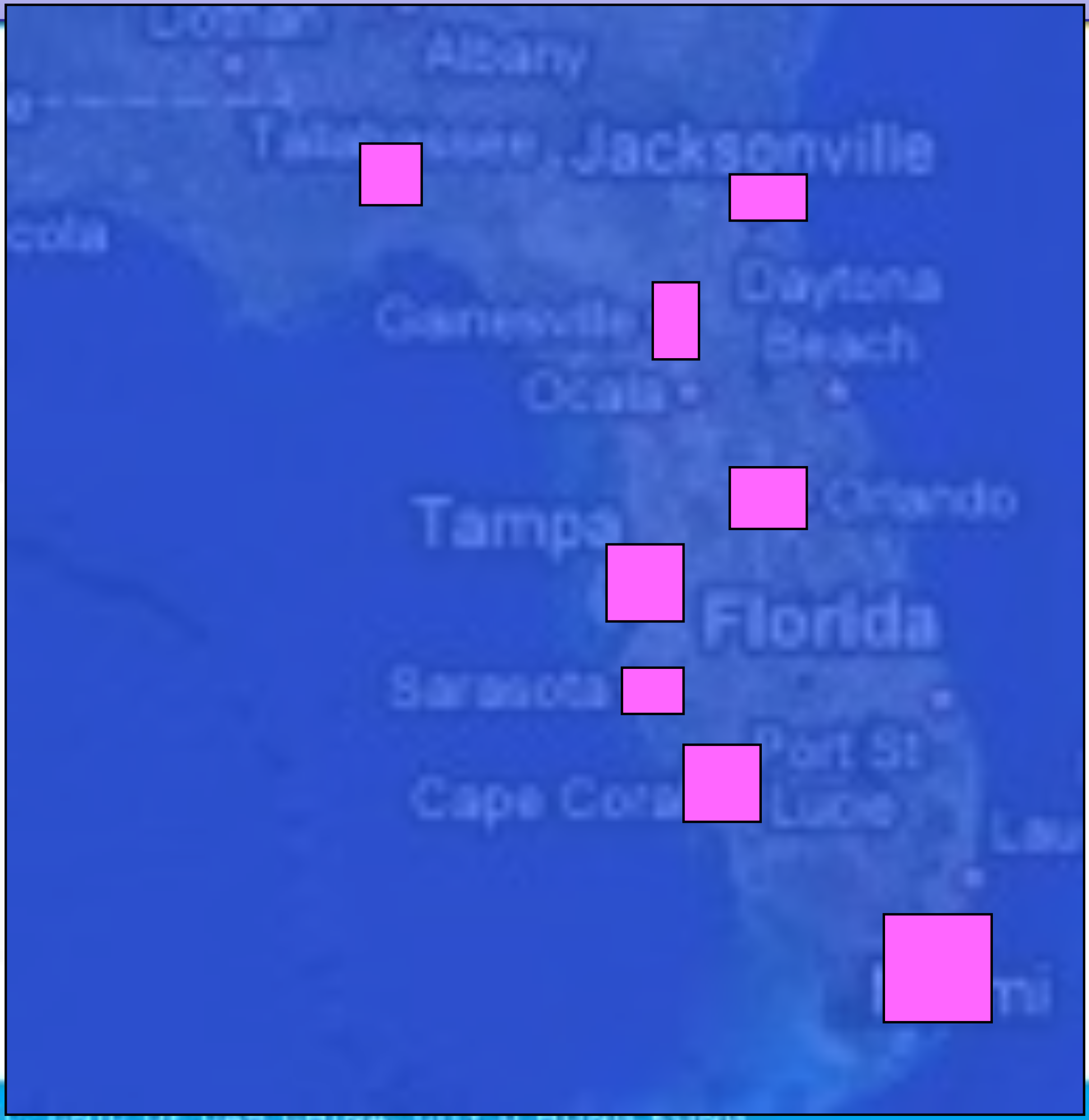
- High-resolution forecasts with guaranteed simulation execution times
- Human-friendly portal
- High-resolution visualization modality

Three-Layer Nested Domain

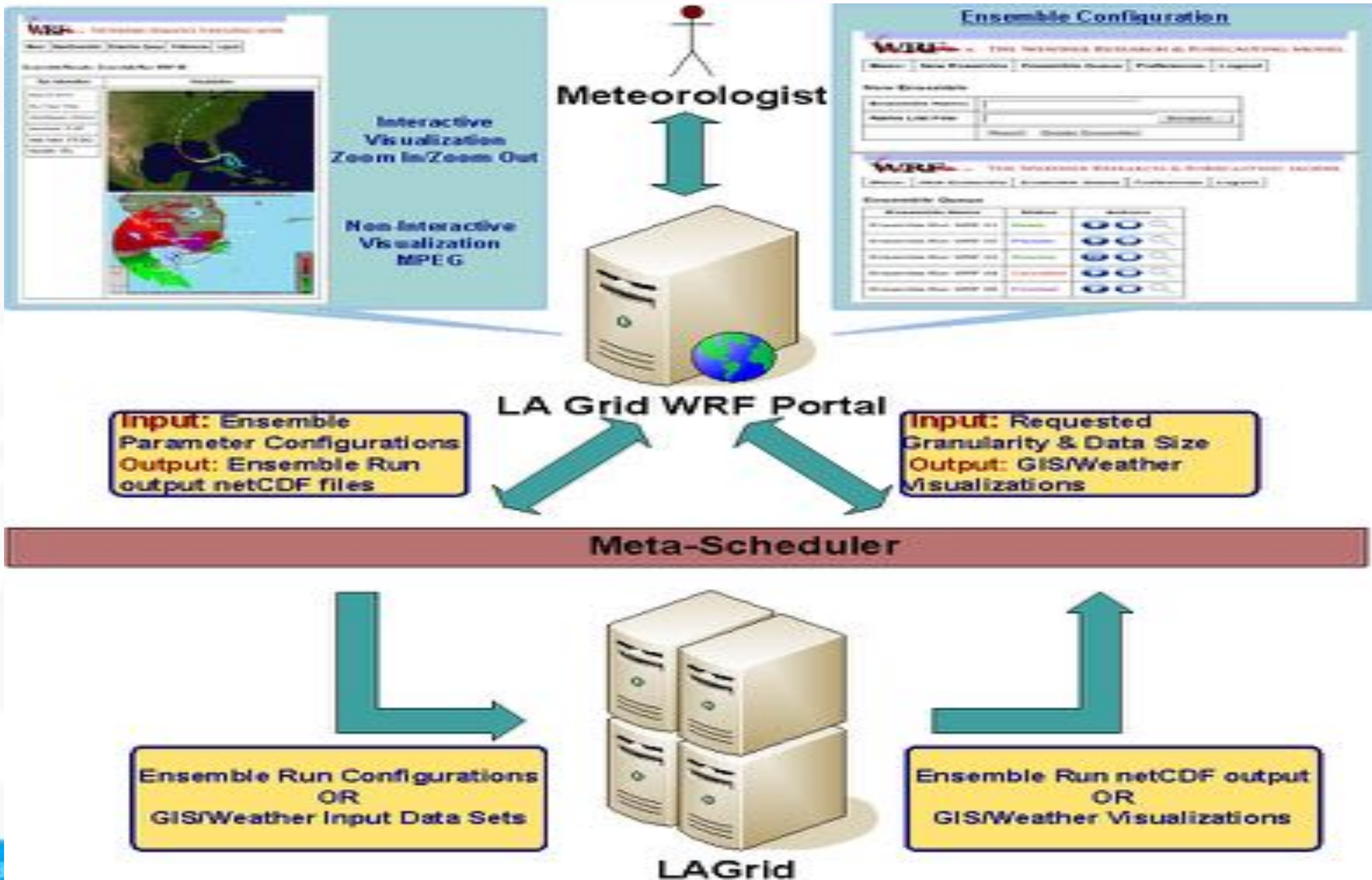
Three-Layer Nested Domain



Three-Layer Nested Domain

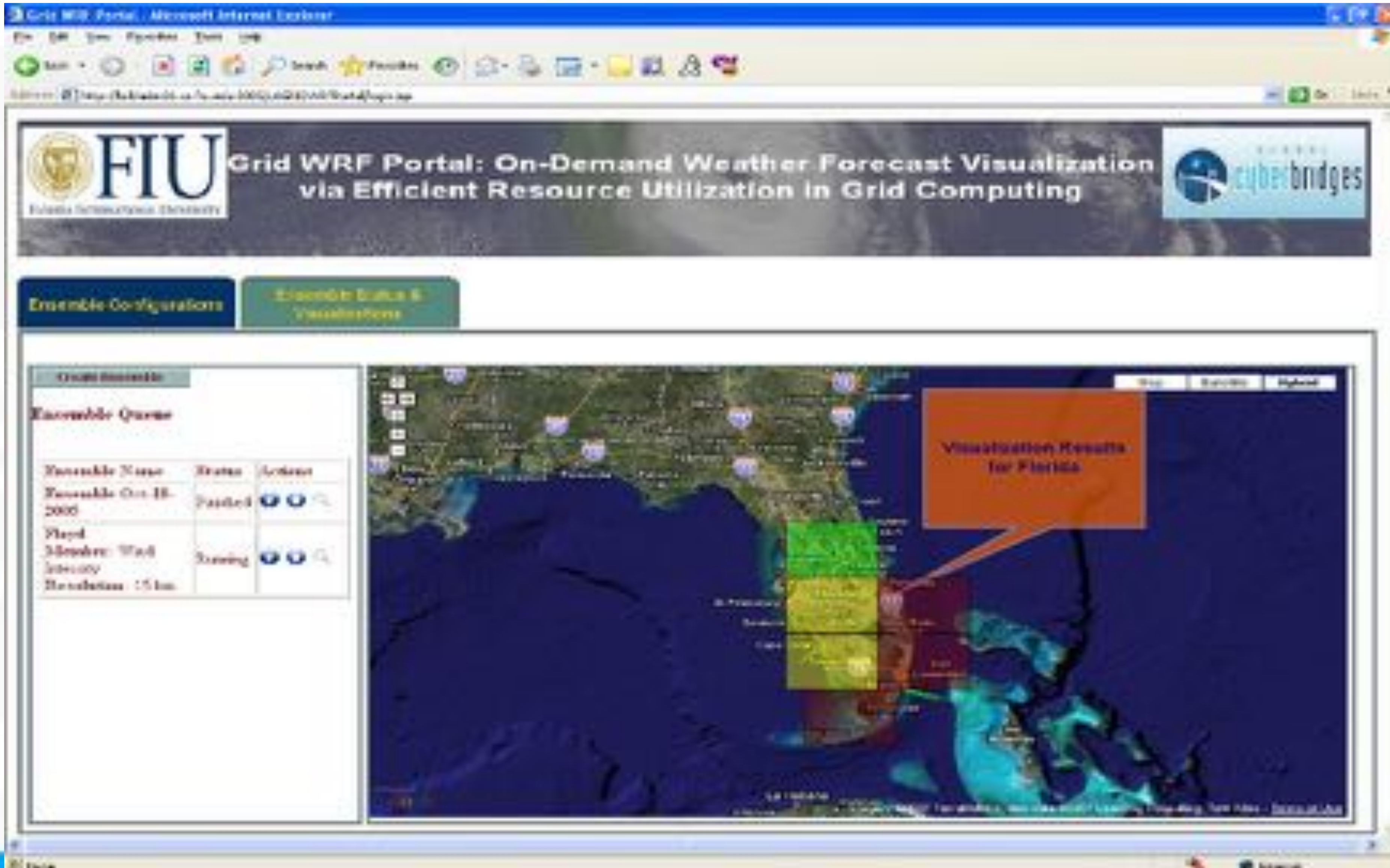


Deployment

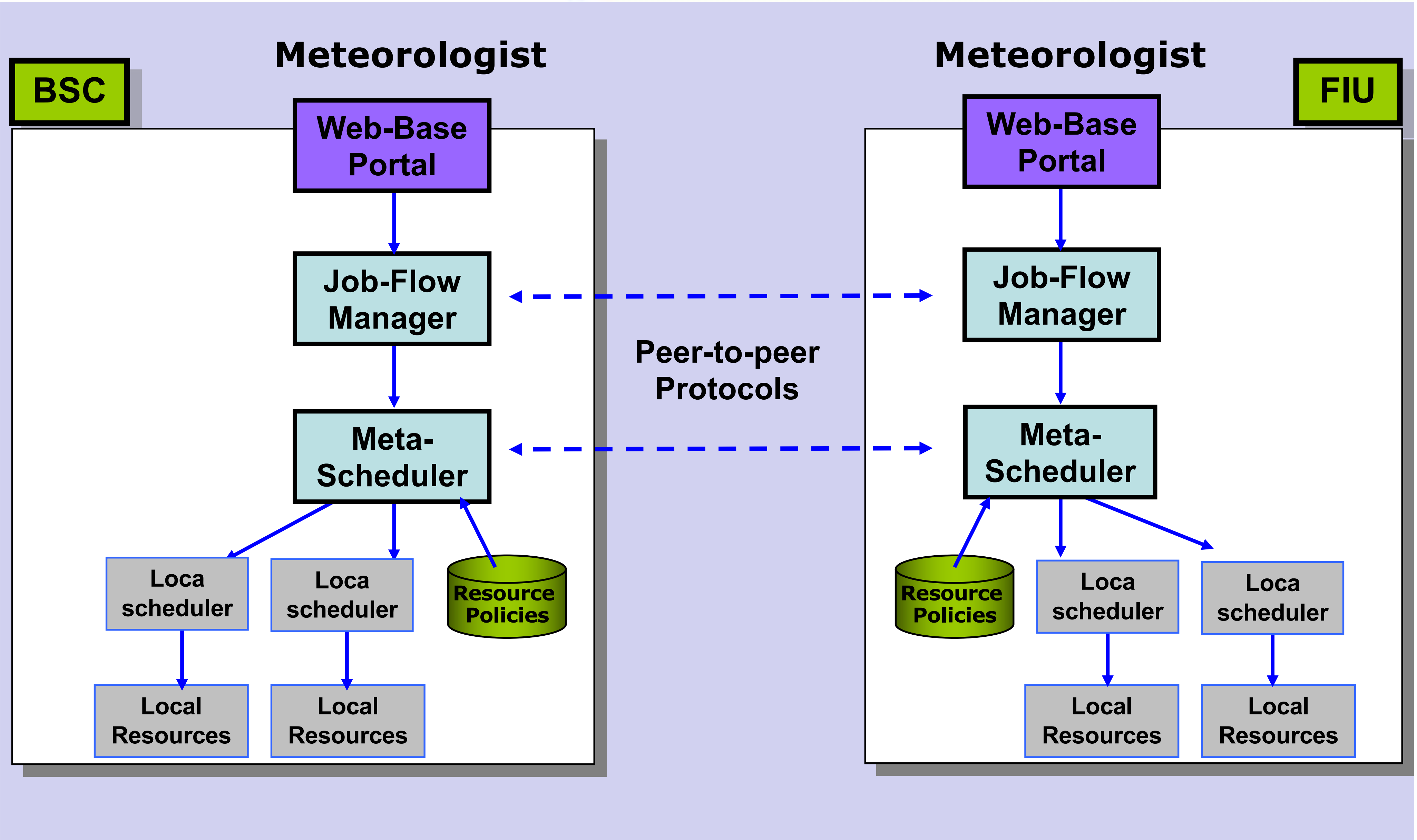


Interactive Visualization Zoom In/Zoom Out
Non-Interactive Visualization MPEG
Meteorologist
Ensemble Configuration
LA Grid WRF Portal
Input: Ensemble Parameter Configurations
Output: Ensemble Run output netCDF files
Input: Requested Granularity & Data Size
Output: GIS/Weather Visualizations
Meta-Scheduler
Ensemble Run Configurations OR GIS/Weather Input Data Sets
LAGrid
Ensemble Run netCDF output OR GIS/Weather Visualizations

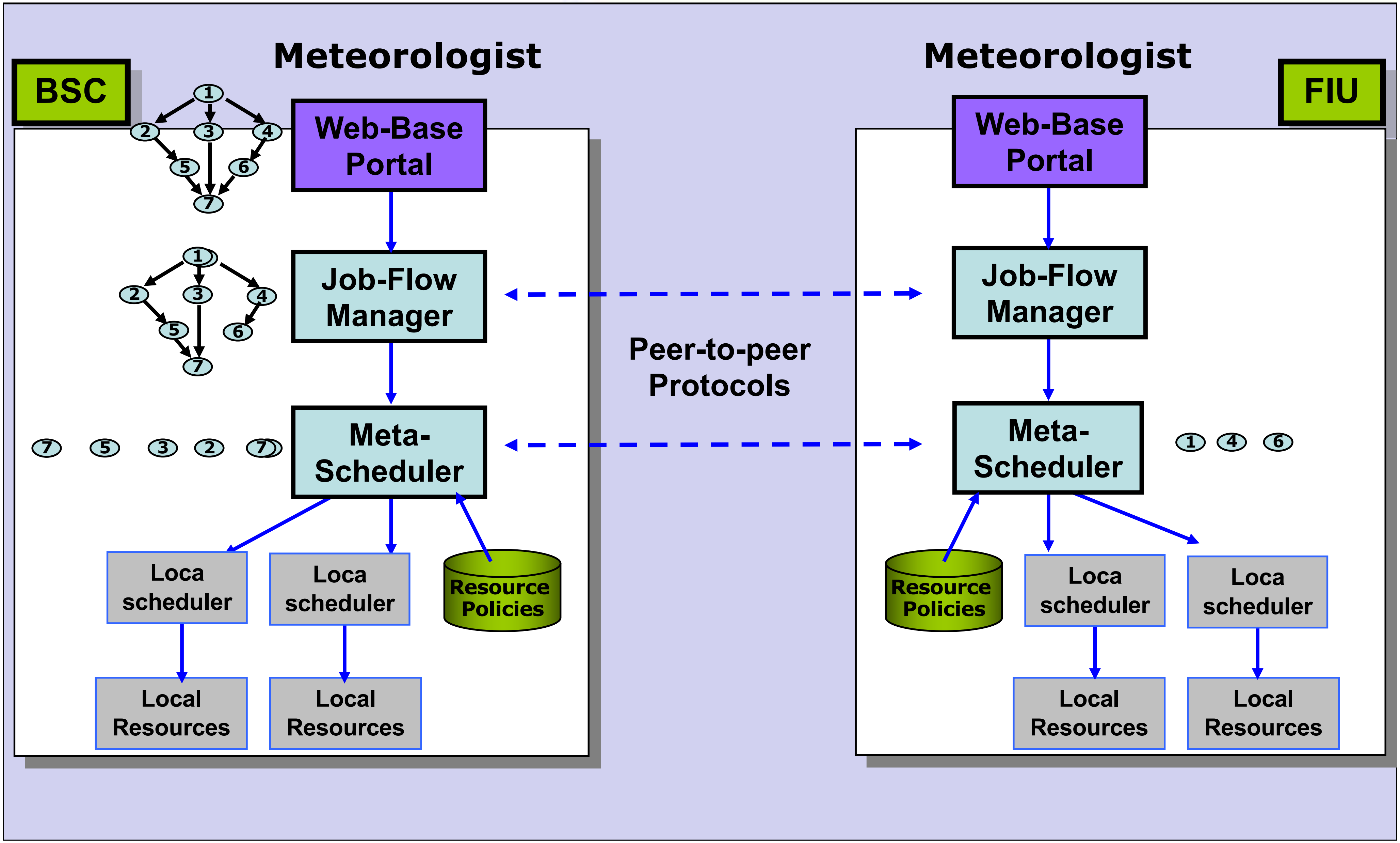
Web-Based Portal

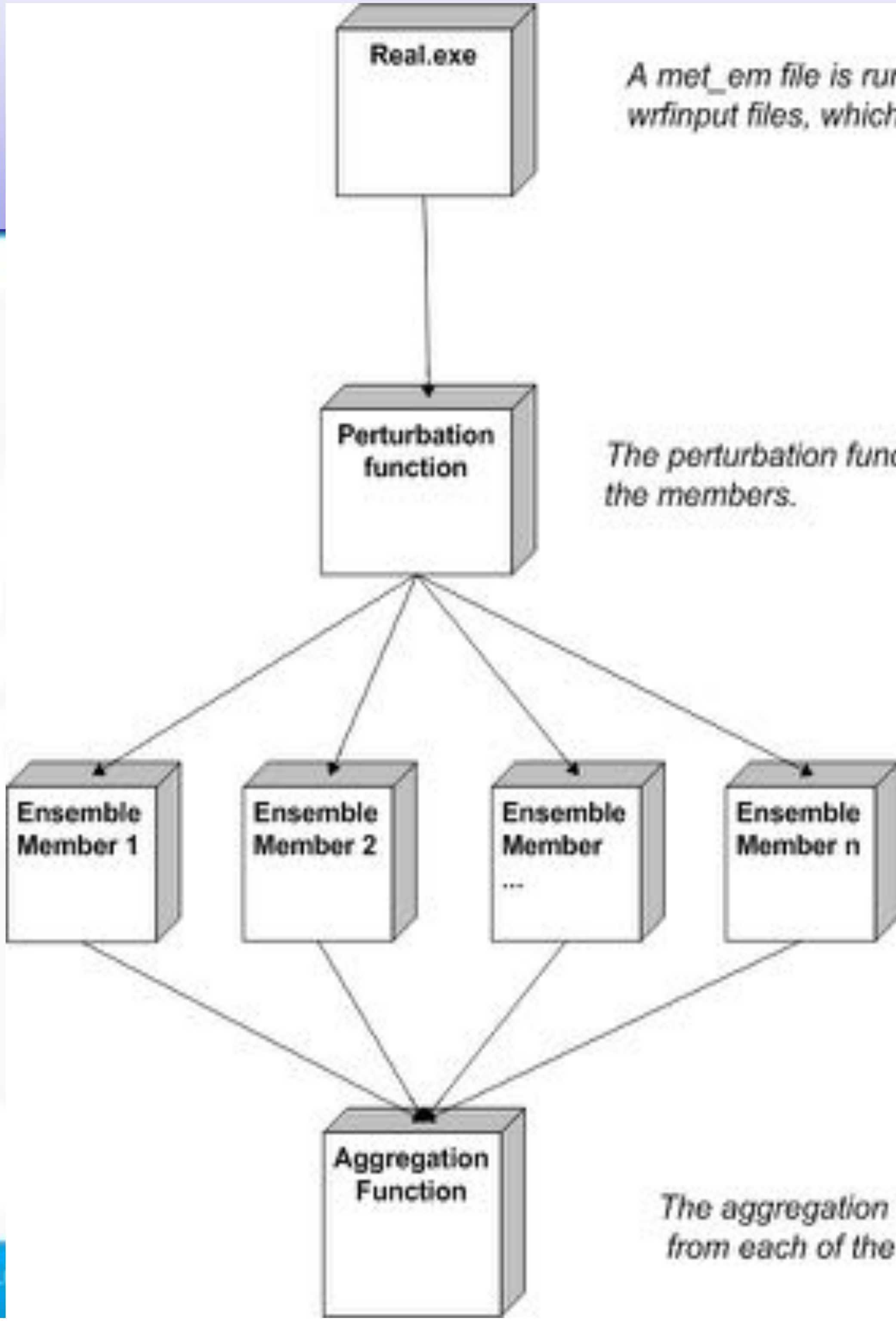

FIU
Grid WRF Portal: On-Demand Weather Forecast Visualization
via Efficient Resource Utilization in Grid Computing
cyberbridges
Ensemble Configurations
Ensemble Queue
Visualization Results
for Florida

Peer-to-Peer Inter-Domain Interactions



Peer-to-Peer Inter-Domain Interactions





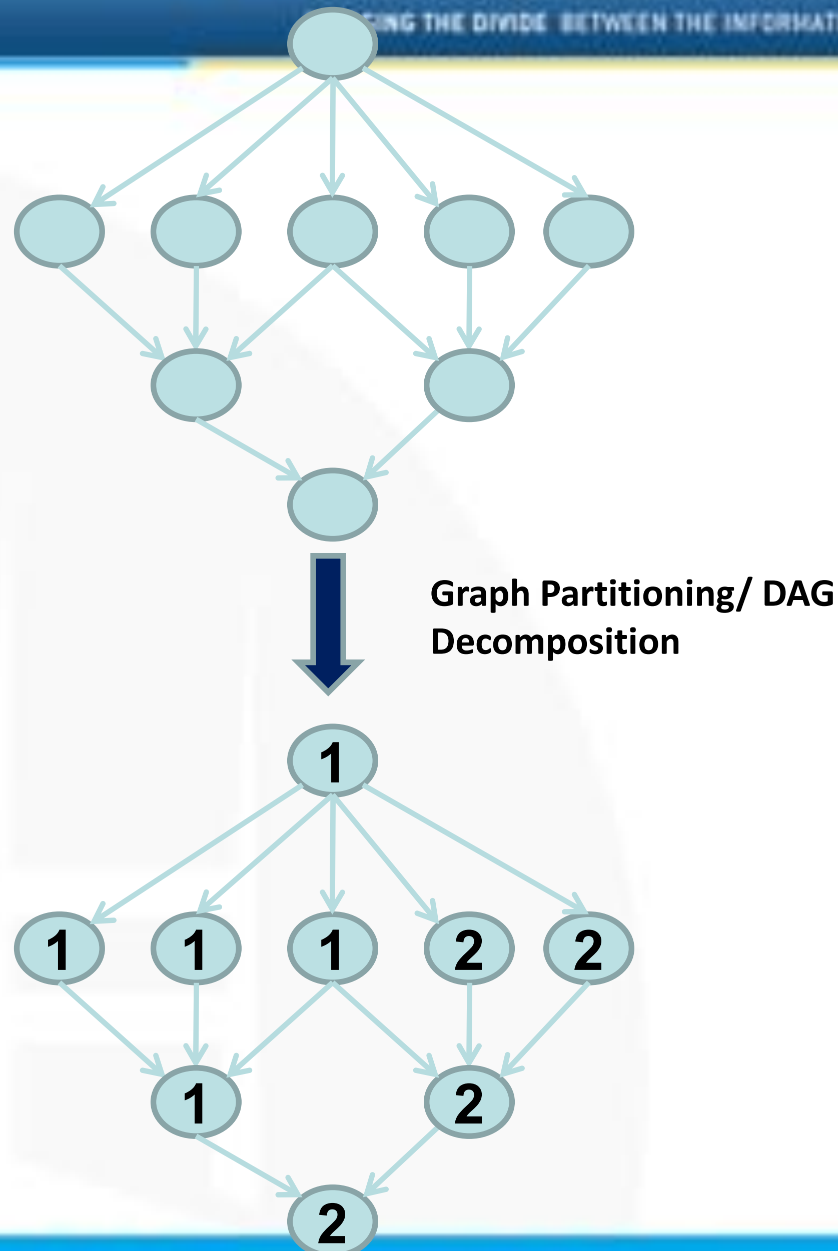
A met_em file is run on Real.exe which produces wrfbdy and wrfinput files, which are then included in an ensemble run.

The perturbation function will make the variations in each of the members.

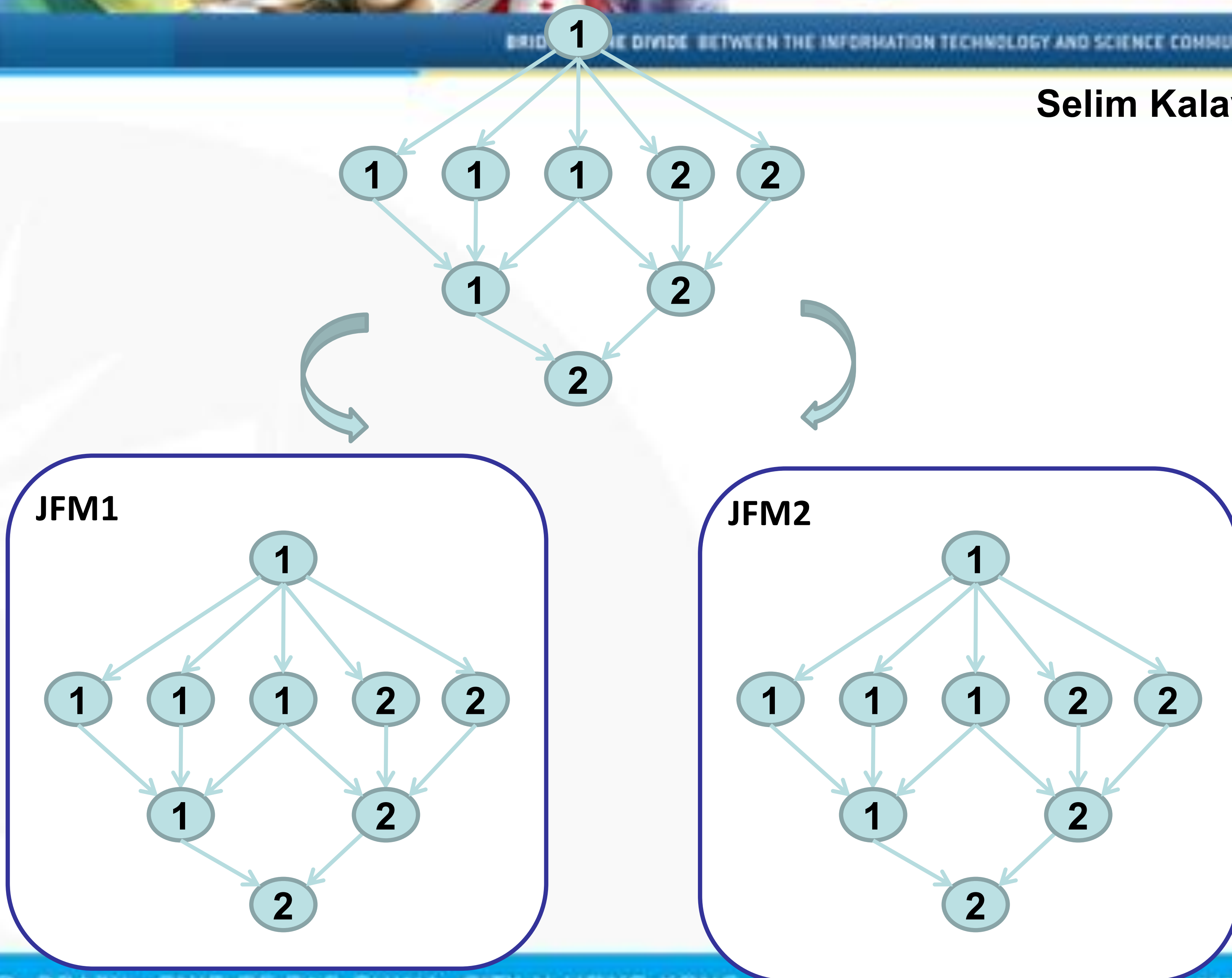
Each Ensemble Member is an instance of WRF. The amount of ensemble members is defined during the creation of the ensemble.

The aggregation function will combine the data obtained from each of the ensemble members into one data set.

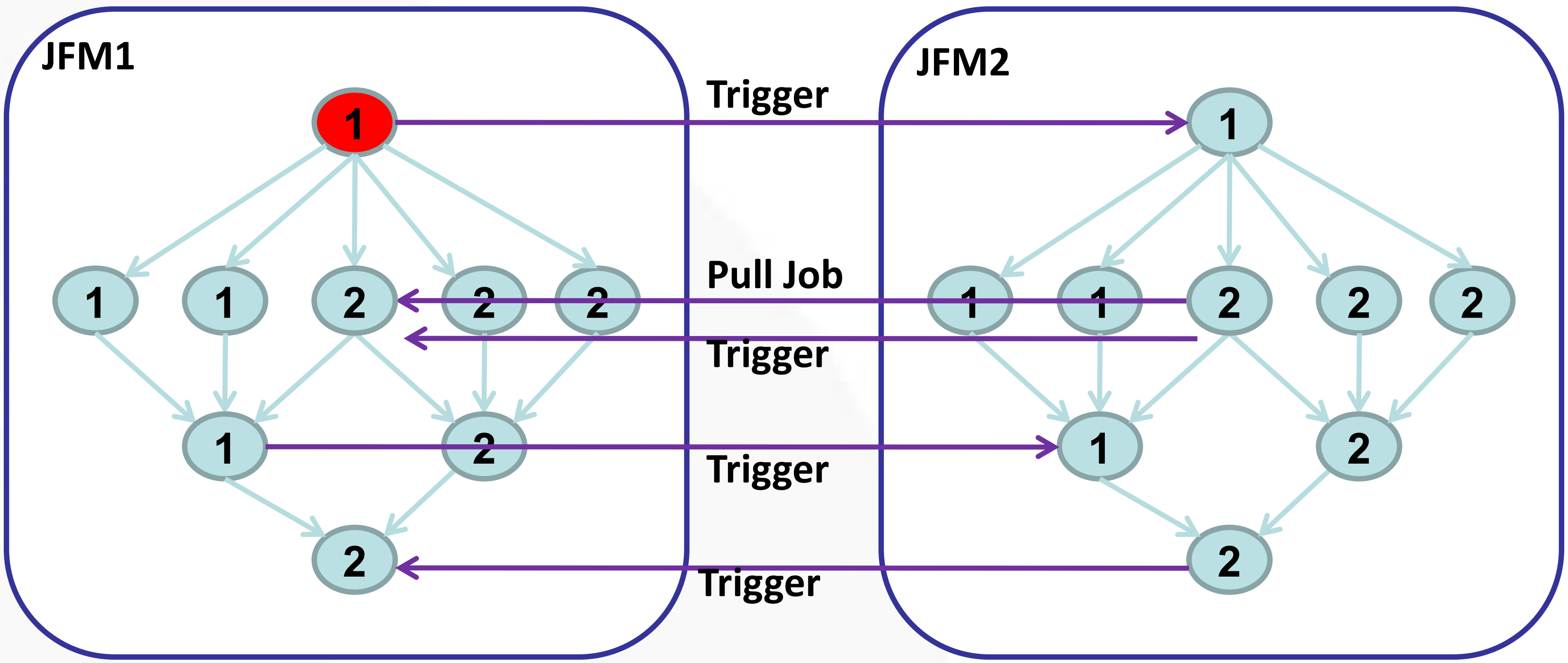
Selim Kalayci



Selim Kalayci



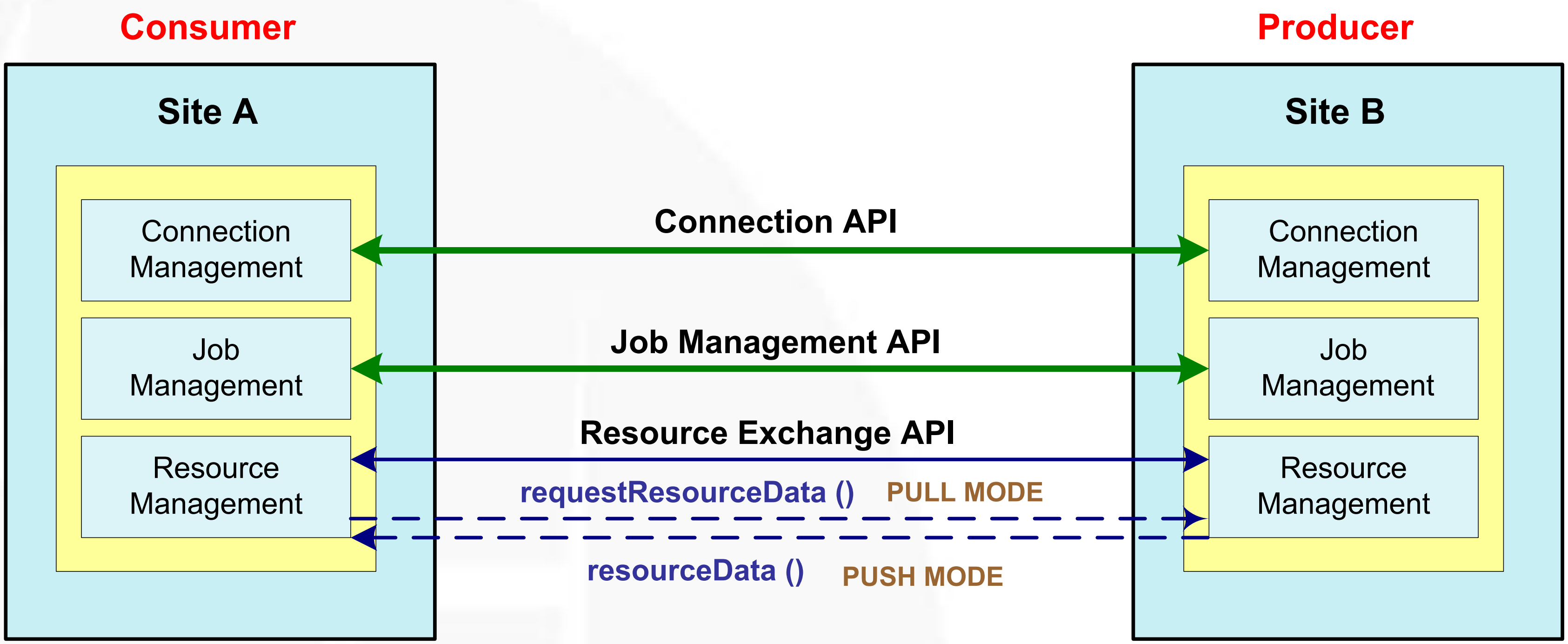
Selim Kalayci



1 = Task finished executing on JFM1

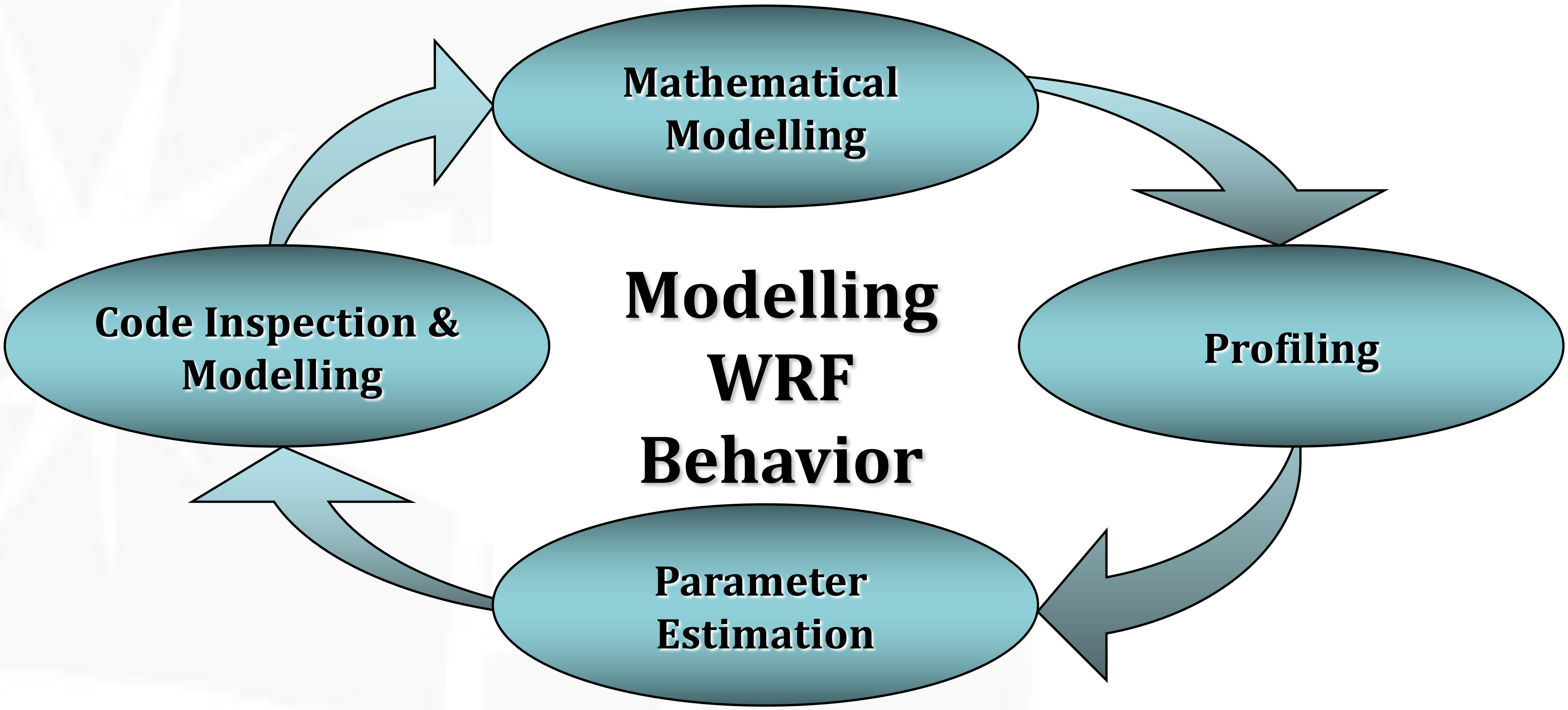
1 = Task still running on JFM1

The Meta-Scheduler P2PProtocol



David Villegas

WRF Behaviour Modelling



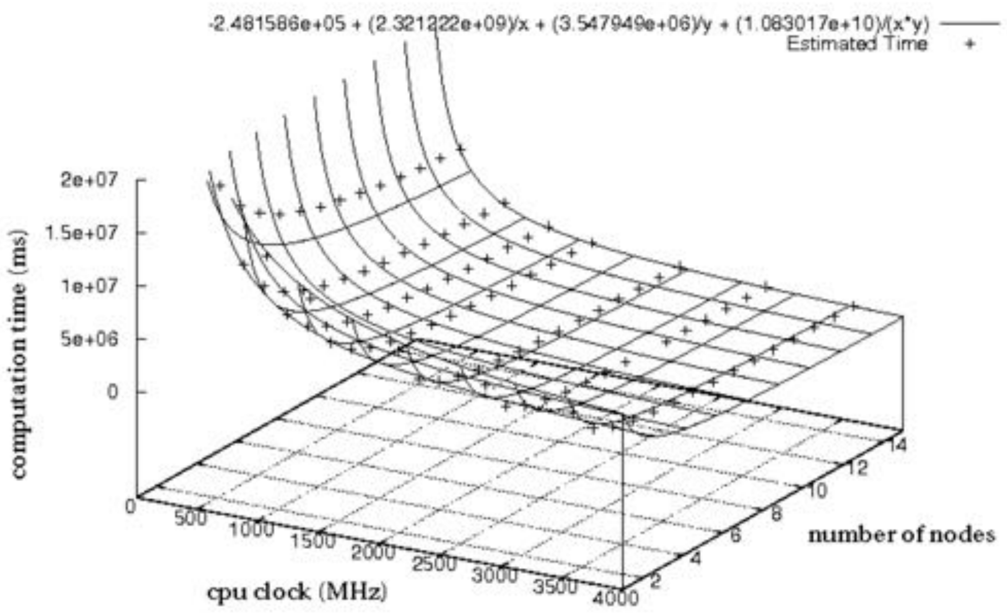
Javier Delgado

Results

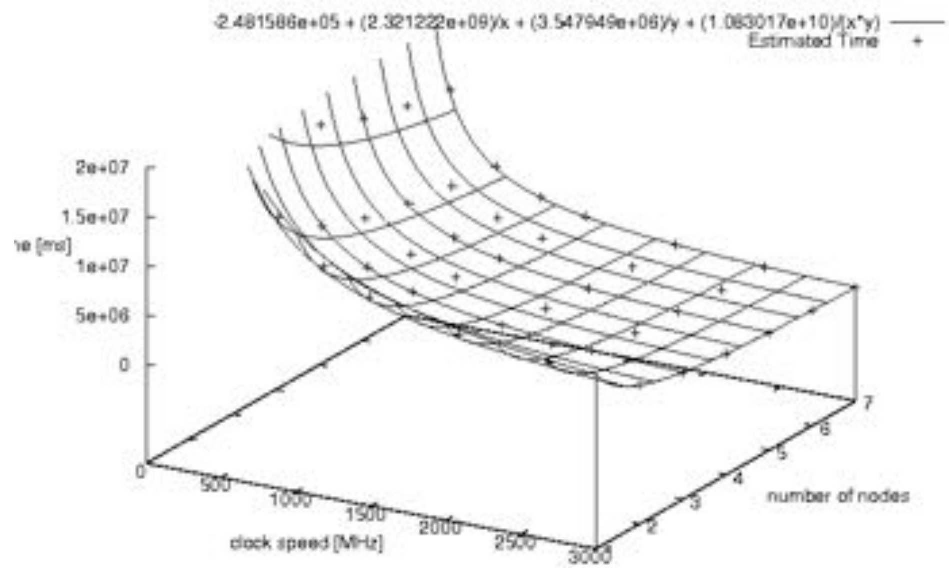
Javier Delgado

Intra-Cluster Error Rate

Mind



GCB



Mean Error: 5.66%
Median Error: 3.80%

Mean Error: 5.34%
Median Error: 5.86%

High-resolution Visualization Project

- **FIU:**
  - Masoud Sadjadi, Heidi Alvarez, David Villegas, Javier Delgado, Javier Figueroa, and Marlon Bright.
- **CNIC CAS (China):**
  - Wendy Zhao and Bi Shuren.
- **UFF and USP (Brazil):**
  - Silvio Luiz Stanzani and Mark Eirik Scortegagna Joselli

- High-resolution Visualization
  - Built on top of the Scalable Adaptive Graphics Environment (SAGE)

4x5 SAGE Display Wall at CNIC

SAGE Overview

- Scalable
  - Hundreds of Screens can be used
  - Built with high-performance applications in mind

- Extensible
  - Provides API for creating custom SAGE applications
  - But this is also a problem
    - Porting an application is not trivial
    - There's a lot of applications out there!

Enhancements

- Porting the Mozilla Firefox Web browser
  - Many emerging applications are web-based
  - The web browser is the platform
  - Native SAGE Web Browser would give optimal performance
- Remote Desktop Enhancement
  - A responsive remote desktop modality is essential for collaboration and e-Learning
  - Users can share their display for all collaborators to see
  - Non-portable applications can be displayed also

Enhancements

- Wii Remote input interface
  - A traditional mouse makes it difficult to work with a large display

SAGE Display Wall at FIU

Agenda

- Why HPC?
- What is HPC anyway?
- Scaling OUT vs. Scaling UP!
- Example Grid Enabling Projects
- Collaboration Tools

The EVO (Enabling Virtual Organizations) System

- Go to http://evo.caltech.edu
- Click on the “Register” button
- Carefully fill out the registry form
- Check you email for the confirmation email
- Follow the instruction in your email to complete the registration process
- Go back to http://evo.caltech.edu
- Click on the “Start” button
- Follow the instructions
- Join the EDUCAUSE Meeting

Acknowledgements

- NSF Award No. 0537464
- CIARA Staff
- Henri Casanova. Thank You for your slides!
  - Principles of High Performance Computing
  - http://navet.ics.hawaii.edu/~casanova
  - henric@hawaii.edu
- Some of the definitions provided in this lecture are based on those in Wikipedia. Thank You!
  – http://en.wikipedia.org/wiki/Main_Page

Questions?

LambdaVision 100-Megapixel display and SAGE (Scalable Adaptive Graphics Environment) software developed by the Electronic Visualization Laboratory at the University of Illinois at Chicago. Major funding provided by NSF.

Email info@cyberbridges.net

Website www.cyberbridges.net

Thank You!

Masoud Sadjadi, Ph.D.
Assistant Professor
School of Computing and Information Sciences
Florida International University
sadjadi@cs.fiu.edu
http://www.cs.fiu.edu/~sadjadi/