LEAD, Workflows and Virtual Grids

Dan Reed

Dan_Reed@unc.edu

Renaissance Computing Institute (RENCI)

University of North Carolina at Chapel Hill

VGrADS Site Visit

April 28, 2005

VGrADS Collaboration Credits

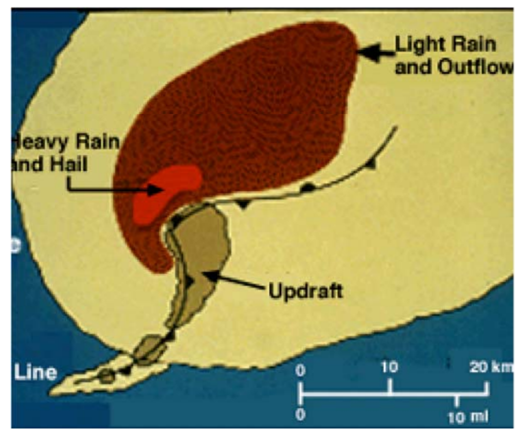
- UC Santa Barbara (NWS)
  - Graziano Obertelli
  - Rich Wolski
- UC San Diego (vgES)
  - Andrew Chien
- Rice University (Scheduling)
  - Ken Kennedy
  - Rob Fowler
  - Anirban Mandal
  - Ryan Zhang
- Illinois/NCSA (Scheduling)
  - Jay Alameda
  - Mark Straka
  - Bob Wilhelmson
- Tennessee (Fault Tolerance)
  - Jack Dongarra
  - Jeffery Chen
- UNC Chapel Hill
  - Emma Buneci
  - Kevin Gamiel
  - Min Lim
  - Lavanya Ramakrishnan
  - Mark Reed
  - Brad Viviano
  - Ying Zhang

Presentation Outline

- **Linked Environments for Atmospheric Discovery (LEAD)**
  - computer science research drivers
    - static/dynamic workflow management and scheduling
    - reliability and performance optimization
  - virtual grid validation and assessment
- **Workflow scheduling and validation**
  - Rice scheduler, NWS/HAPI measurement and LEAD
- **NWS/HAPI integration**
  - distributed system assessment for reliability
- **Qualitative behavioral classification**
  - see also Emma Buneci's poster
    - "A Framework for Reasoning About the Temporal Behavior of Scientific Applications"

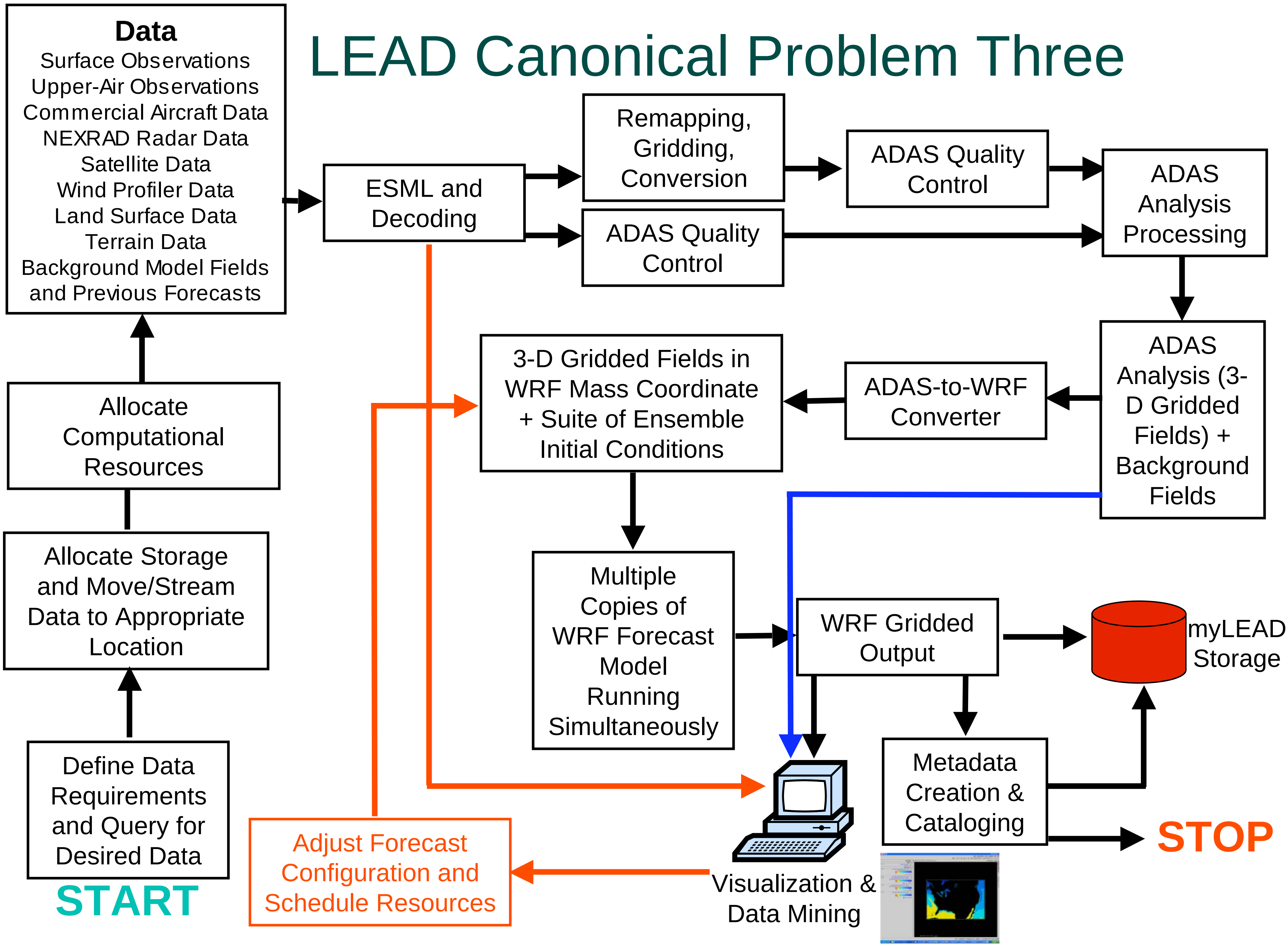
Linked Environments for Atmospheric Discovery

- **Rationale**
  - Each year, mesoscale weather – floods, tornadoes, hail, strong winds, lightning, hurricanes and winter storms – causes hundreds of deaths, routinely disrupts transportation and commerce, and results in annual economic losses in excess of $13B.
- **LEAD participants**
  - Oklahoma, UNC, Indiana, NCAR, Alabama, Illinois, Millersville St, …
- **From "offline" to "online" forecasting**
  - data assimilation and adaptive evaluation



Unique LEAD Attributes

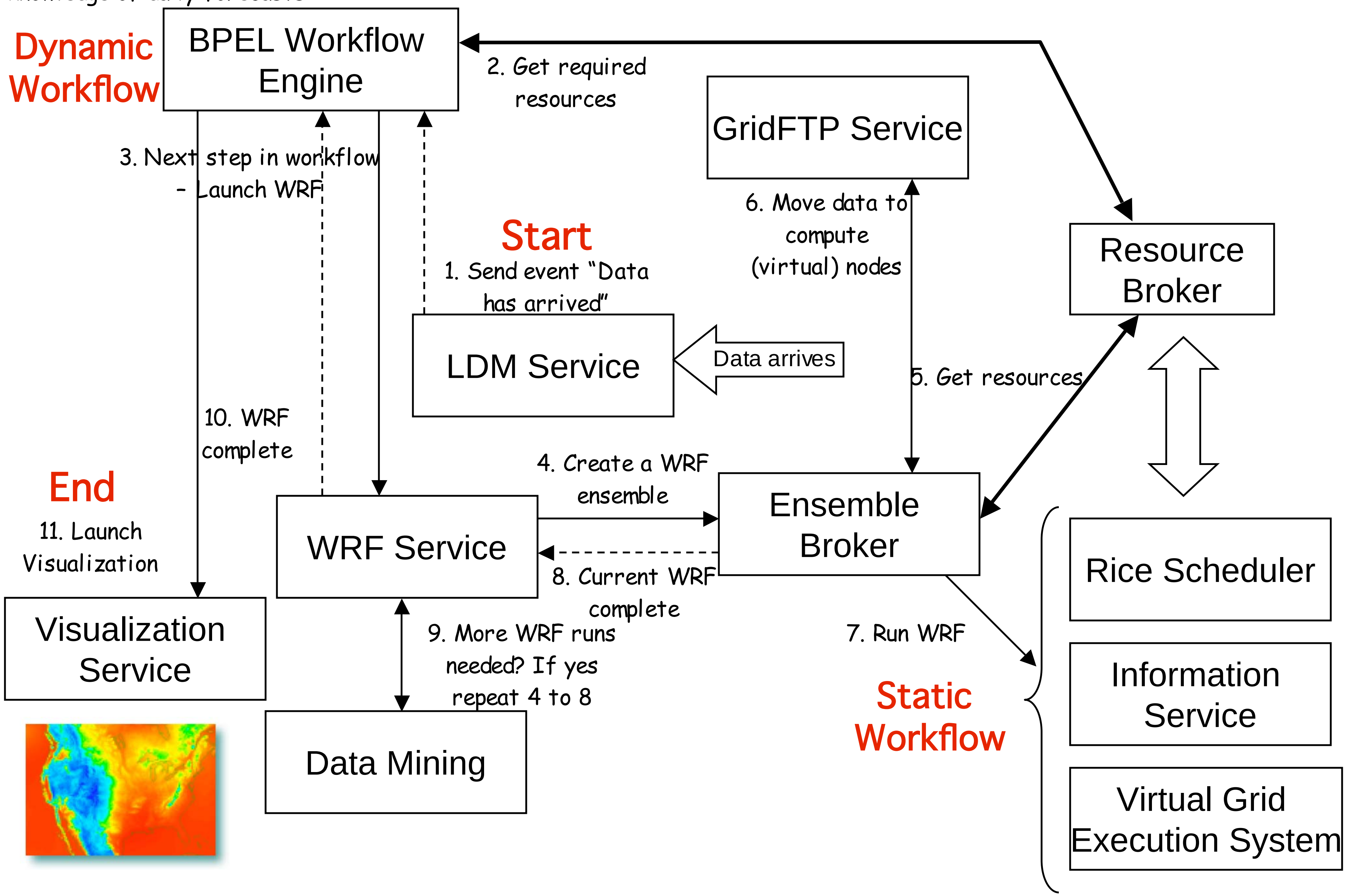
- **Couple analysis and assimilation tools, forecast models, and data repositories as dynamically adaptive, on-demand services to**
  - change configuration rapidly and automatically in response to weather
  - continually be steered by unfolding weather
  - respond to decision-driven inputs from users
  - initiate other processes automatically
    - dynamic, data driven workflows
  - steer remote observing technologies
    - to optimize data collection for the target problem
- **From VGrADS perspective**
  - application driver with characteristics different than e.g., EMAN
    - steaming data, multilevel workflow, adaptation, …
- **Canonical LEAD problem three**
  - produce high-resolution, nested WRF ensemble forecasts
  - respond dynamically to prevailing and predicted weather conditions



LEAD Canonical Problem Three
Data
Surface Observations
Upper-Air Observations
Commercial Aircraft Data
NEXRAD Radar Data
Satellite Data
Wind Profiler Data
Land Surface Data
Terrain Data
Background Model Fields and Previous Forecasts
ESML and Decoding
Remapping, Gridding, Conversion
ADAS Quality Control
ADAS Quality Control
ADAS Analysis Processing
ADAS Analysis (3-D Gridded Fields) + Background Fields
ADAS-to-WRF Converter
3-D Gridded Fields in WRF Mass Coordinate + Suite of Ensemble Initial Conditions
Multiple Copies of WRF Forecast Model Running Simultaneously
WRF Gridded Output
myLEAD Storage
Metadata Creation & Cataloging
STOP
Visualization & Data Mining
Adjust Forecast Configuration and Schedule Resources
Allocate Computational Resources
Allocate Storage and Move/Stream Data to Appropriate Location
Define Data Requirements and Query for Desired Data
START

0. Possible advanced reservation. Resources for the services based on knowledge of daily forecasts

LEAD Workflow Implementation

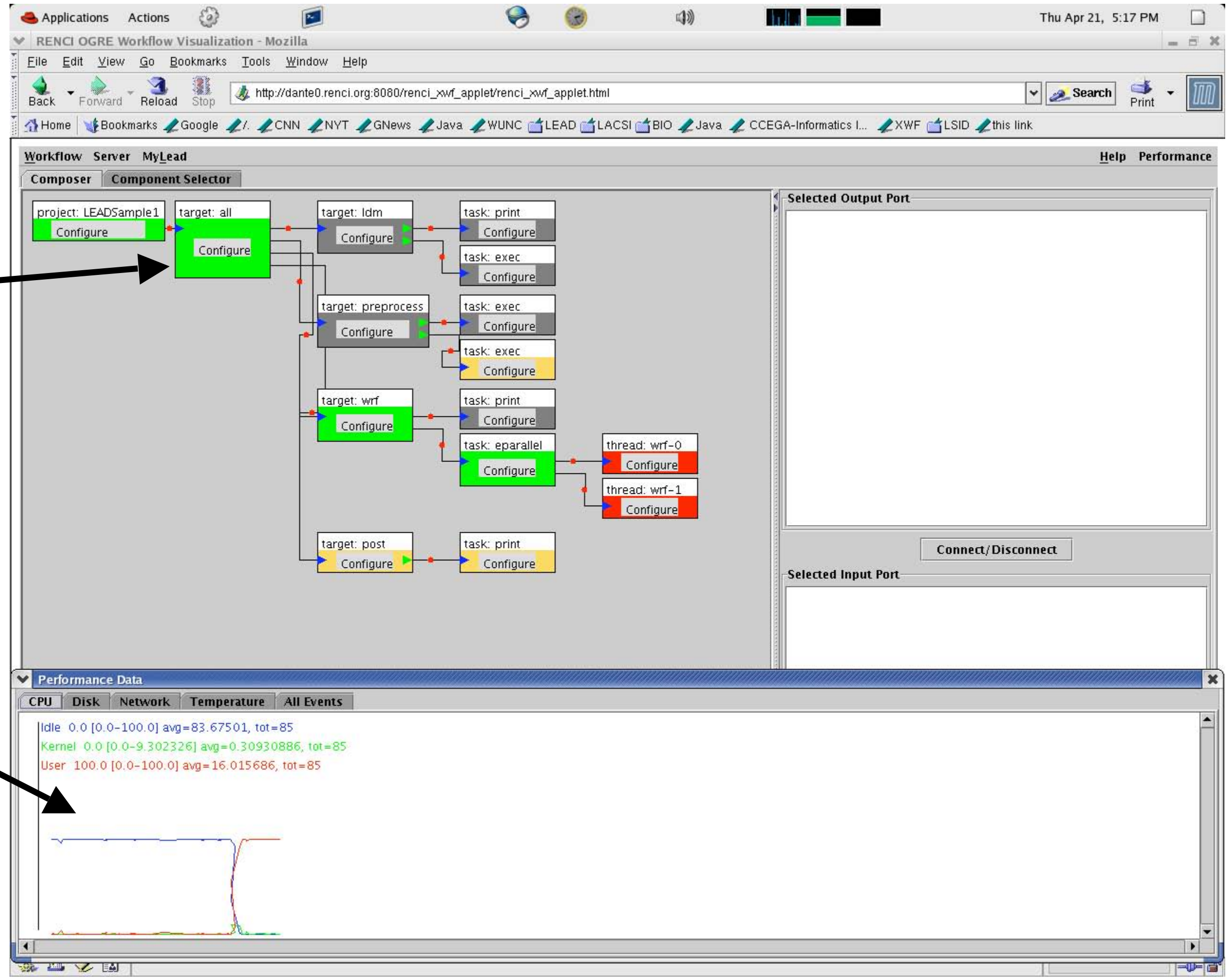


LEAD Orchestration Interface (LEAD Funded)

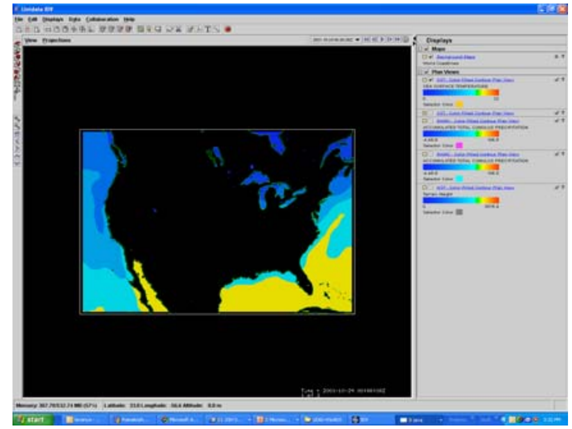
Color coded with execution status

Performance and reliability metrics

(Autopilot, NWS and HAPI)



Computing Research Drivers from LEAD



Unidata IDV

- **Complex services and virtualization**
  - complexity management and amelioration
  - virtual grid interfaces and mechanisms
  - multilevel workflow management
- **Measurement and monitoring techniques**
  - performance and system health
    - large, parametric studies can run for weeks

      "spring runs" underway now at PSC
- **Prediction and classification mechanisms**
  - failure indicators and long-term reliability
    - redundancy and recovery
  - application temporal classification and combination
- **Integrated management policies**
  - performance, fault tolerance, power management, …

vgDL Specification for LEAD Workflow

LEADSpec = LDMNode = {isLDMNode} // Note the loose coupling

far WRFNode = {memory >= 500MB, cpu > 2000, diskspace > 4GB}

far VizNode = {memory >=4GB, cpu > 4000}

vgidLEAD = vgCreateVG(vgESsrv.renci.org, LEADspec, 1000, null);

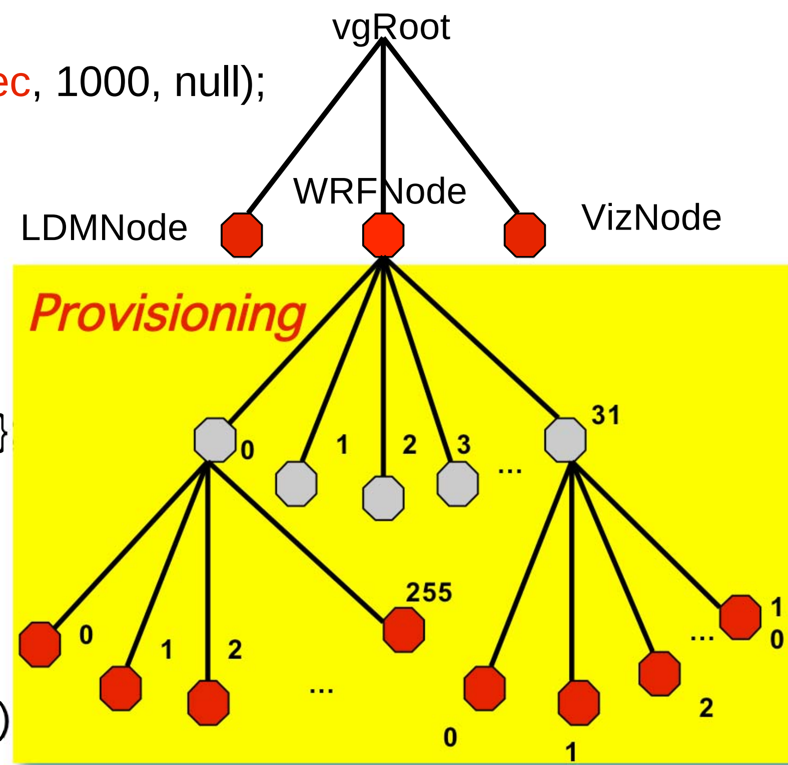
vgRoot = vgGetRoot(vgidLEAD);

WRFspec = LooseBagof<C> [1:32];

C = Clusterof<node>[4:256];

node = {node.memory > 500MB, node.cpu > 2000};

C = {C.hasSharedFileSystem = true}



vgidLEAD = vgGetMyVG(); wrfNode = vgGetMyNode()

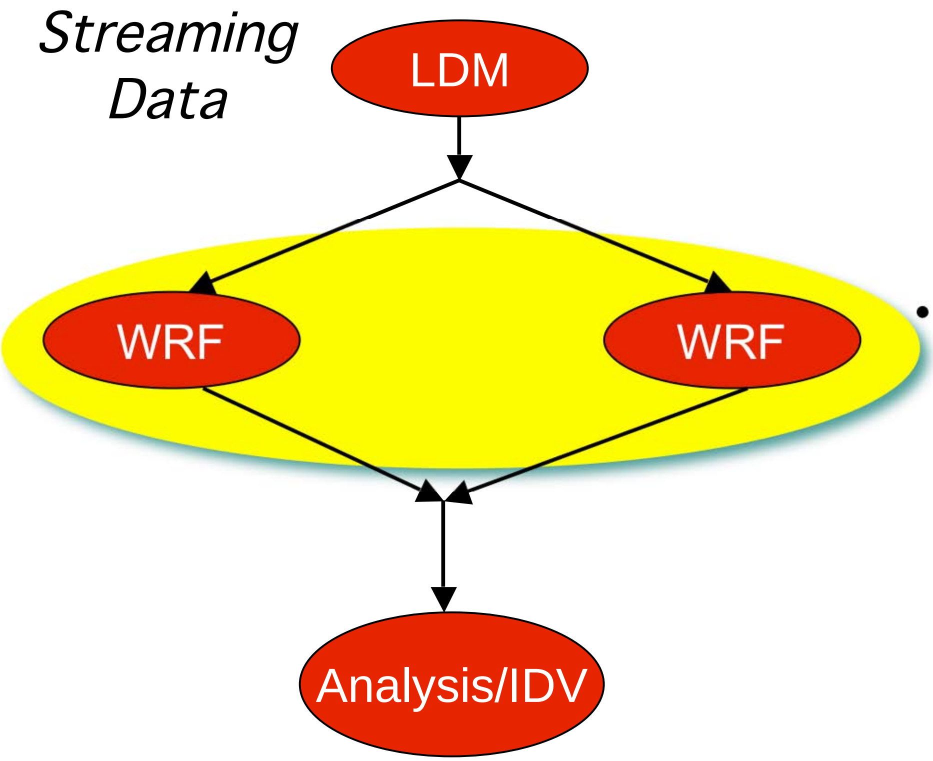
status = vgAddToVG(vgidLEAD, WRFspec, WRFNode,1000, init_WRF);

// Run WRF and await ensemble completion (*repeat as needed*)

LEAD Workflow and vgDL Implications

- **Multi-level workflow management**
  - static workflow (i.e., one instantiation of an execution)
    - Open Grid Computing Environments Runtime Engine (OGRE)
      extension of Apache Ant (Java makefile extension)
  - dynamic workflow (i.e., iterative ensemble executions)
    - Business Process Execution Language for Web Services (BPEL4WS)
- ***VGrADS research is workflow independent, however***
  - *EMAN and LEAD illustrate different implementation points for workflows*
- **Streaming data management**
  - Unidata Local Data Management (LDM) streams (e.g., NEXRAD2 data)
  - *fixed source locations that constrain task scheduling (research issue)*
    - *redundancy and reliability sites*
- **Long running application suite**
  - Weather Research and Forecast (WRF)
    - next generation weather code and multiple hours for each ensemble
  - multiscale fault tolerance
    - sites, clusters and data streams

Experimental LEAD Workflow

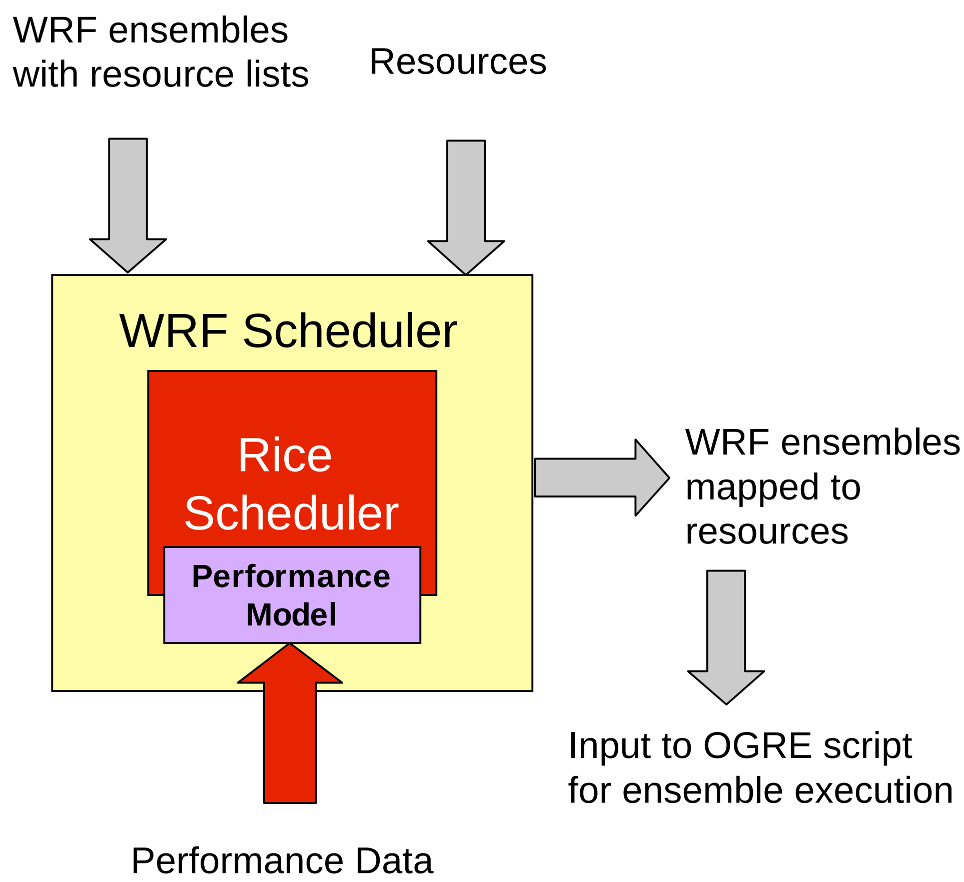


- Test problem (~4 hour execution)
  - 12 km resolution forecast
  - continental United States

- Simple workflow
  - simple LDM data push
  - two ensemble model runs
    - 8 and 16 nodes
  - offline visualization using IDV
- Execution configuration (dante)
  - 35 dual-processor compute nodes
    - Intel Xeon 3.2 GHz
    - 6 GB DRAM
    - ~60 GB local disk
  - 3 front-end nodes
  - interconnect
    - gigabit Ethernet and Infiniband
  - node software
    - ROCKS cluster distribution
    - Red Hat Linux 3.2.3-42

Rice Heuristic Scheduling Algorithm

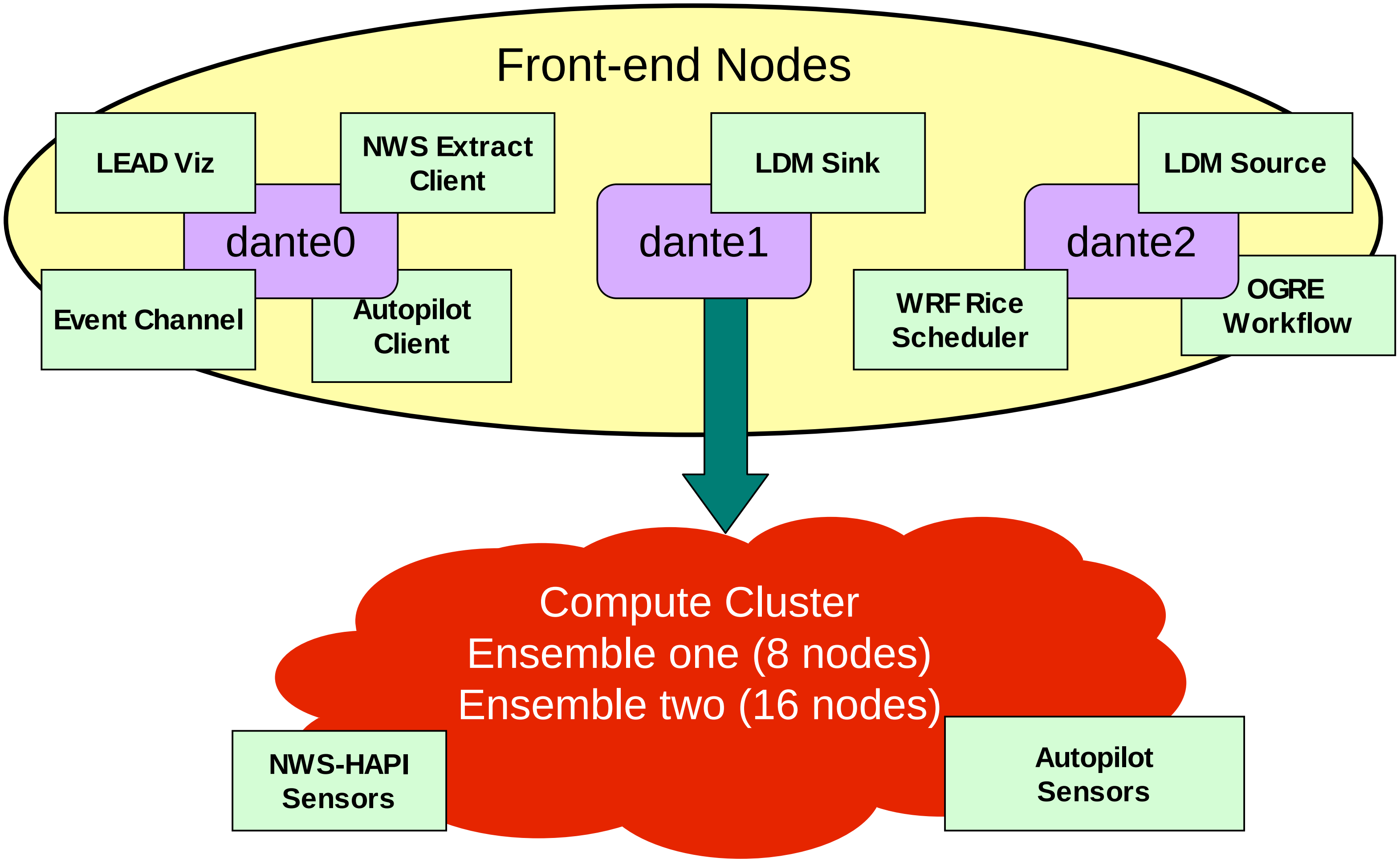
```
while all components not mapped do
    Find availComponents;
    Calculate the rank matrix;
    findBestSchedule(availComponents);
Endwhile

findBestSchedule(comps)
    while all comps not mapped do
        foreach Component, C do
            foreach Resource, R do
                ECT(C,R)=rank(C,R)+EAT(R);
            endforeach
            Find minECT(C,R) over all R;
            Find 2nd_minECT(C,R) over all R;
        endforeach
        j1* = j1 with min(minECT(j1,R));   //min-min
        j2* = j2 with max(minECT(j2,R));  //max-min
        j3* = j3 with min(2nd_minECT(j3,R)-minECT(j3,R));
          //sufferage
        Store mapping for jx* for each heuristic;
        Update EAT(R) and makespan for each heuristic;
    endwhile
    Select mapping with minimum makespan among three;
    Output selected mapping;
```



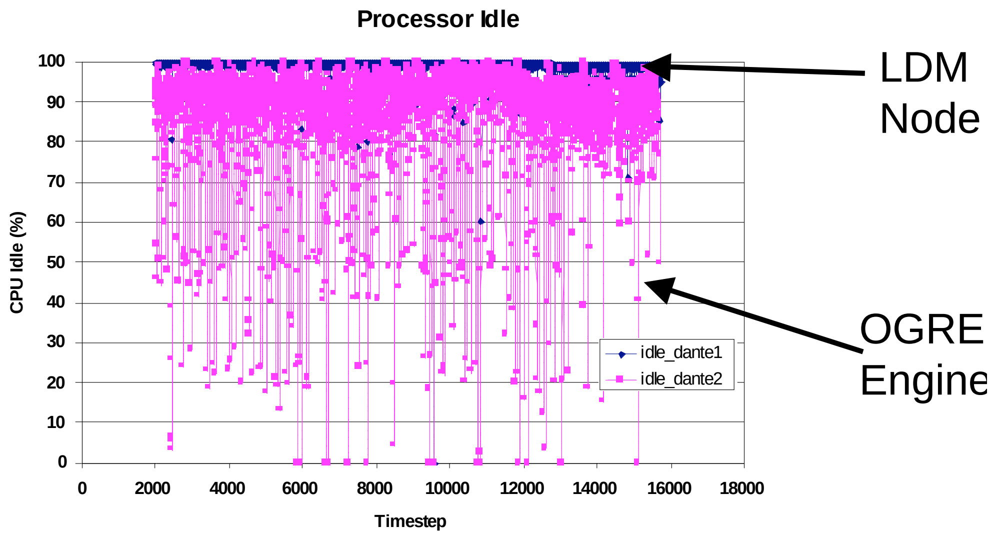
WRF Ensemble Scheduling and Execution

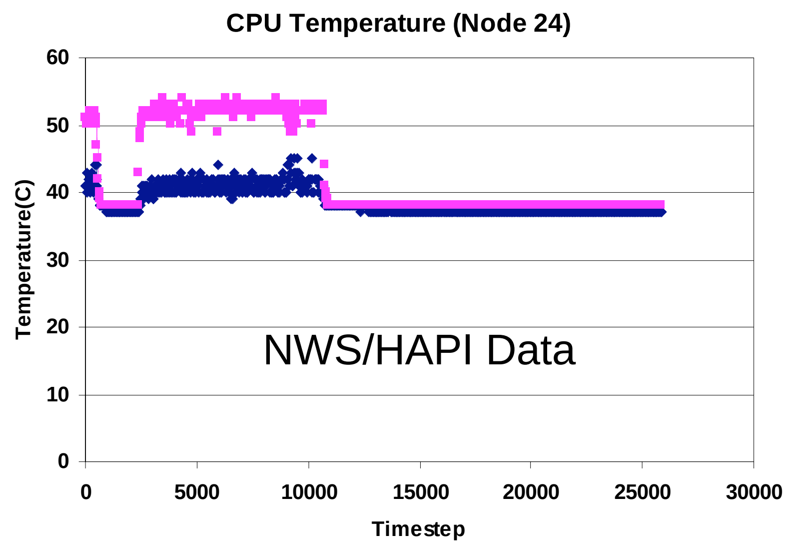
- **Execution time performance model f(x)**
  - as a function of number of resources (x)
  - conditioned by three factors
    - processor load
    - node temperature (reliability)
    - kernel type (uniprocessor or SMP)
- **Execution phases**
  - start all sensors
  - measure CPU utilization and temperature on the nodes
    - NWS, HAPI and/or Autopilot
  - invoke Rice scheduler
    - identify sets of systems for WRF execution
      - two sets in this example
  - execute OGRE script
    - pass system list to the script for job launch

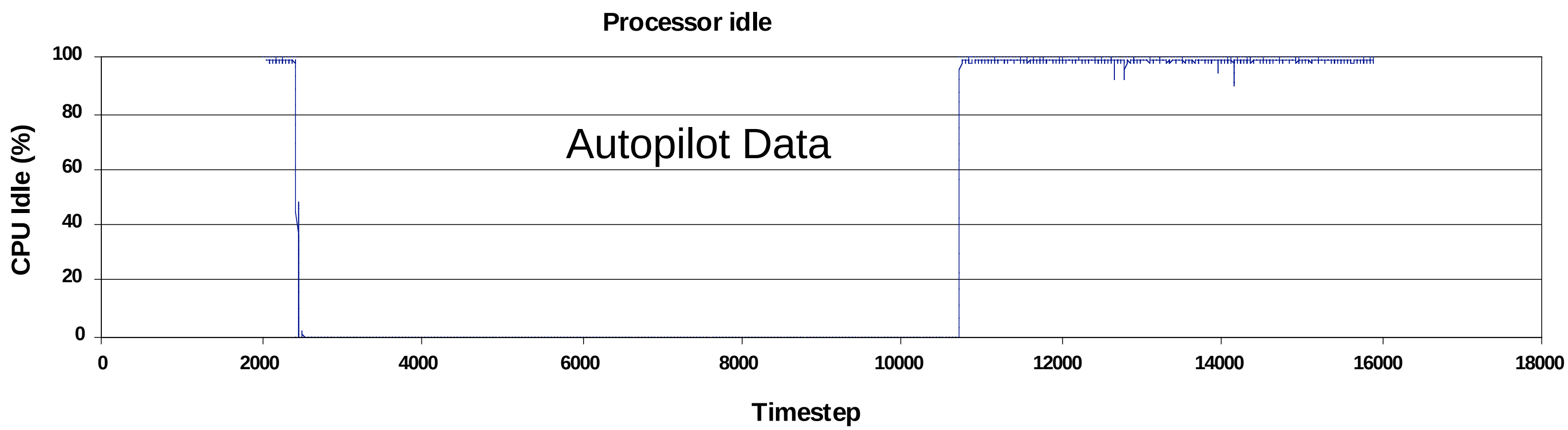
Experimental Configuration



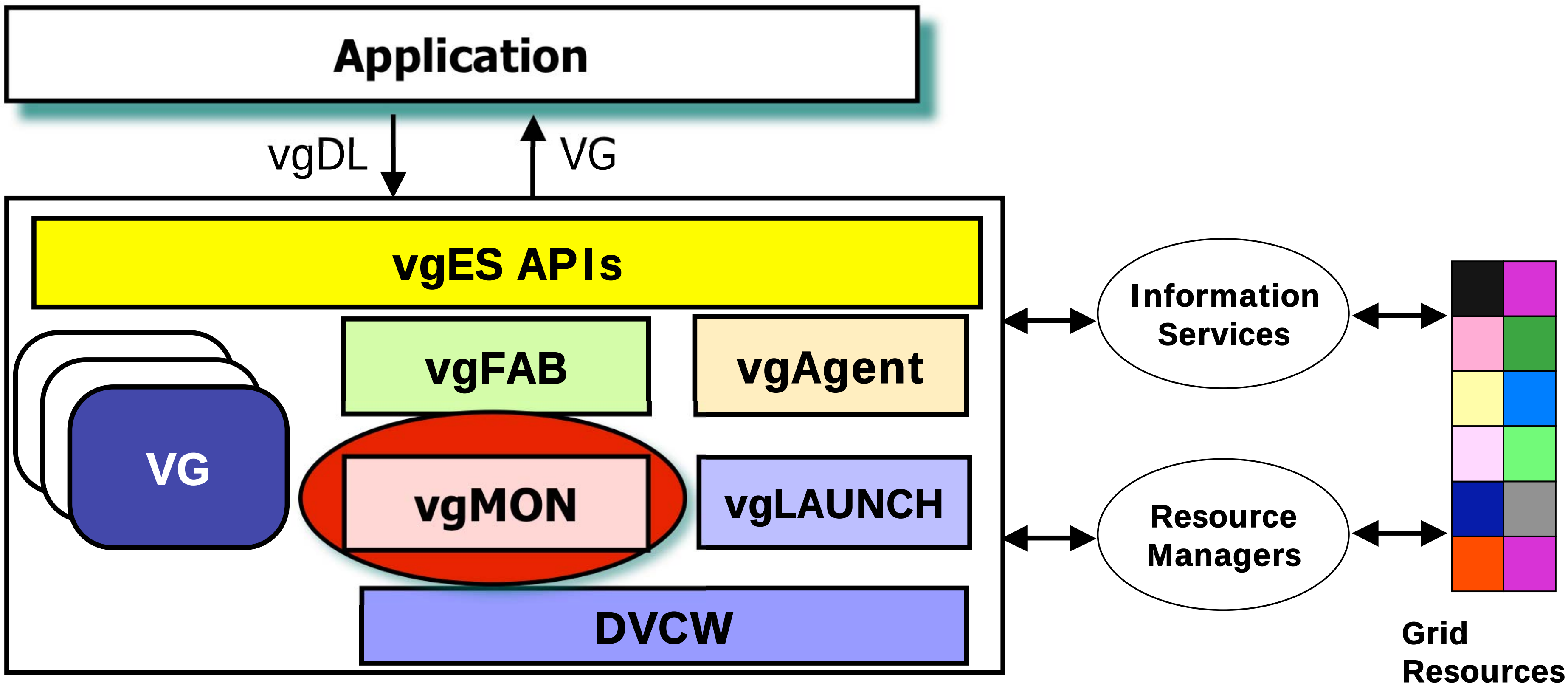
WRF Workflow Execution





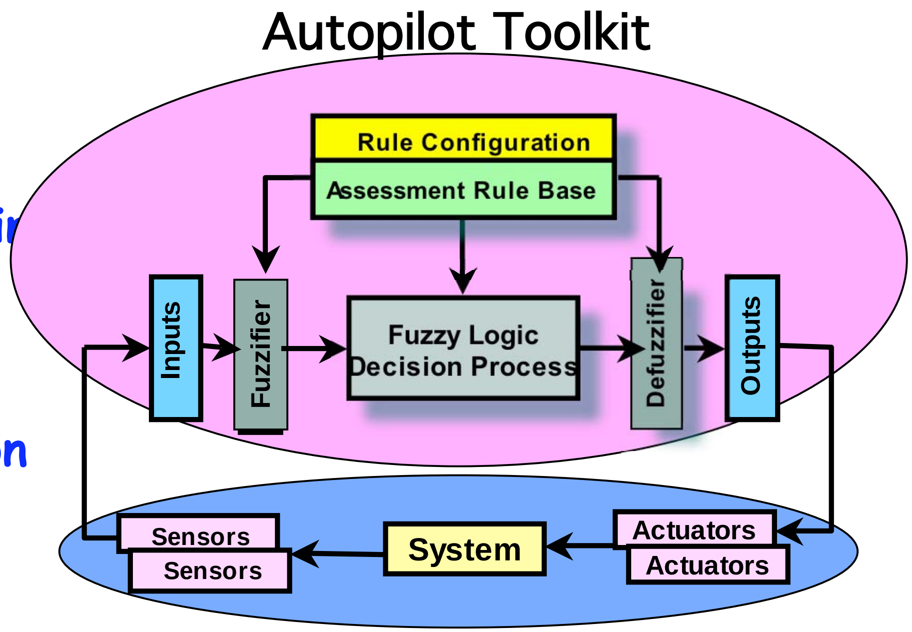


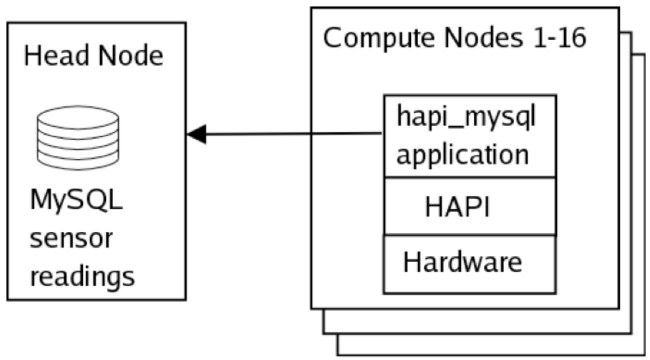
Virtual Grid Execution System



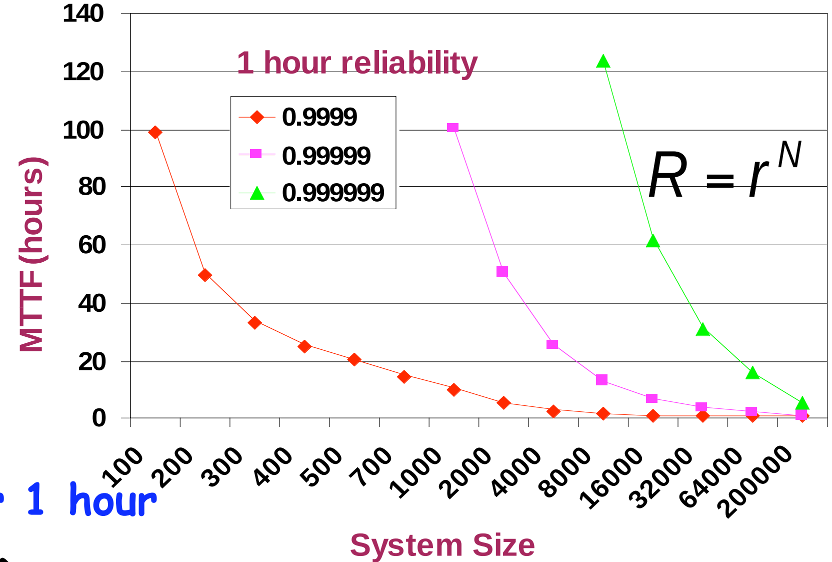
Adaptation: Measurement and Resilience

- **Intelligent monitoring**
  - Autopilot sensors
    - performance metrics
  - HAPI health and failure monitoring
    - SMART, ACPI, lm_sensors
- **Intelligent assessment**
  - failure prediction and remediation
    - macroscale trend analysis
    - microscale trends
  - performance optimization
- ***Virtual grid services (vgMON)***
  - *system health monitoring*
  - *fault tolerance*
    - *microscale and macroscale*
  - *performance measurement*





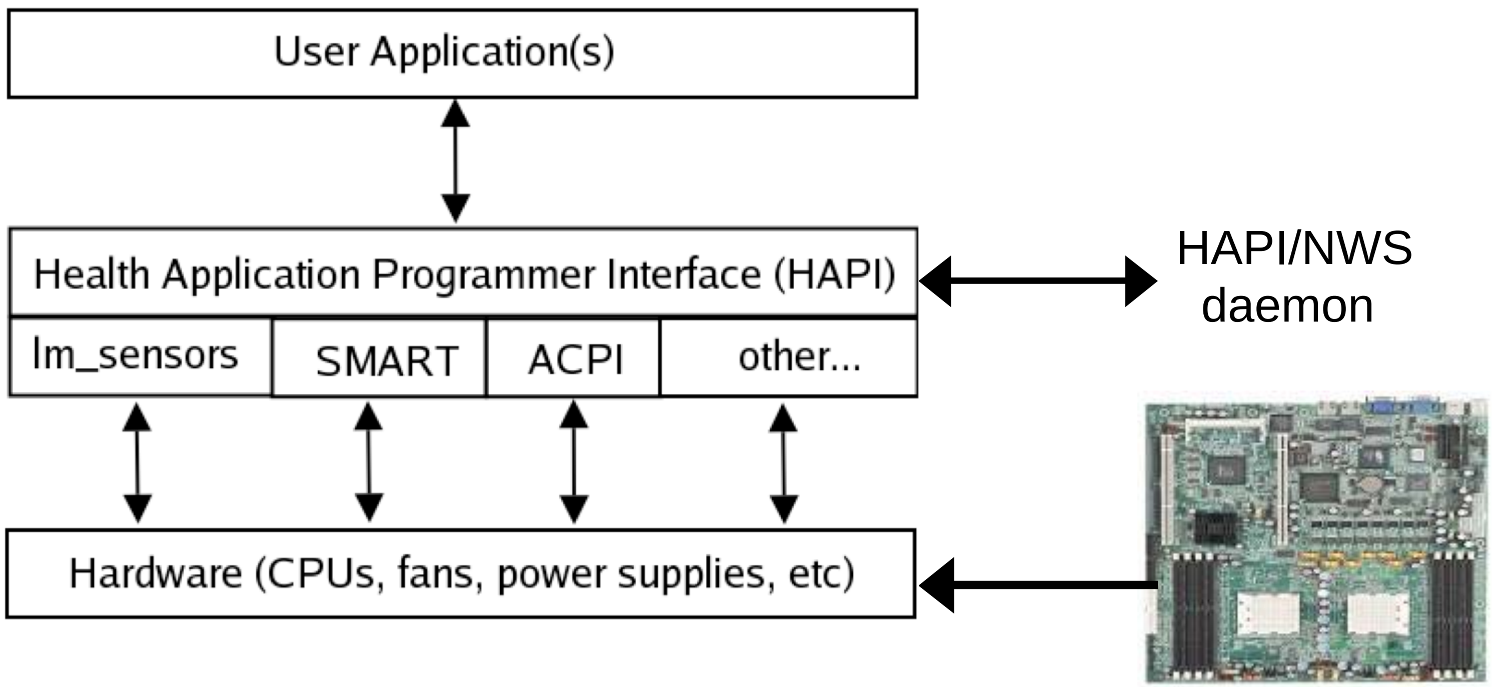
The Implications of Scale



- **Two important drivers**
  - overall system size (Grids and systems)
  - semiconductor component scaling
- **Overall system size (macroscale)**
  - assume independent component failures
    - an optimistic assumption
  - N: the number of processors
  - r: probability a component operates for 1 hour
- **Semiconductor components (microscale)**
  - static power leakage
    - temperature and reliability
  - software memory errors
- ***Applications are susceptible to failures from both***
  - *checkpointing is not enough!*
- ***Daniel Nurmi poster***
  - *Optimal Checkpoint Scheduling using Automatic Resource Characterization*

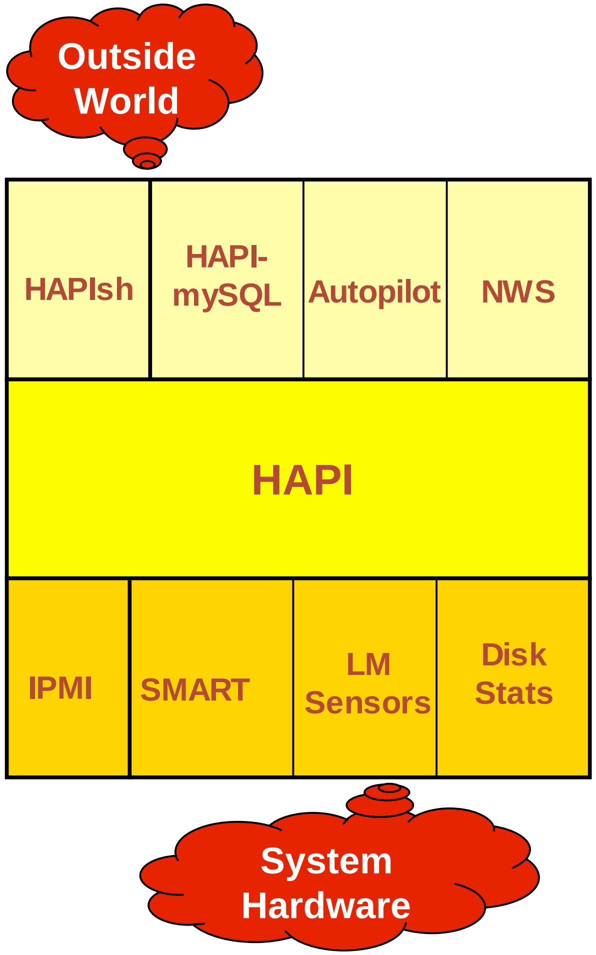
HAPI Failure Indicator Monitor

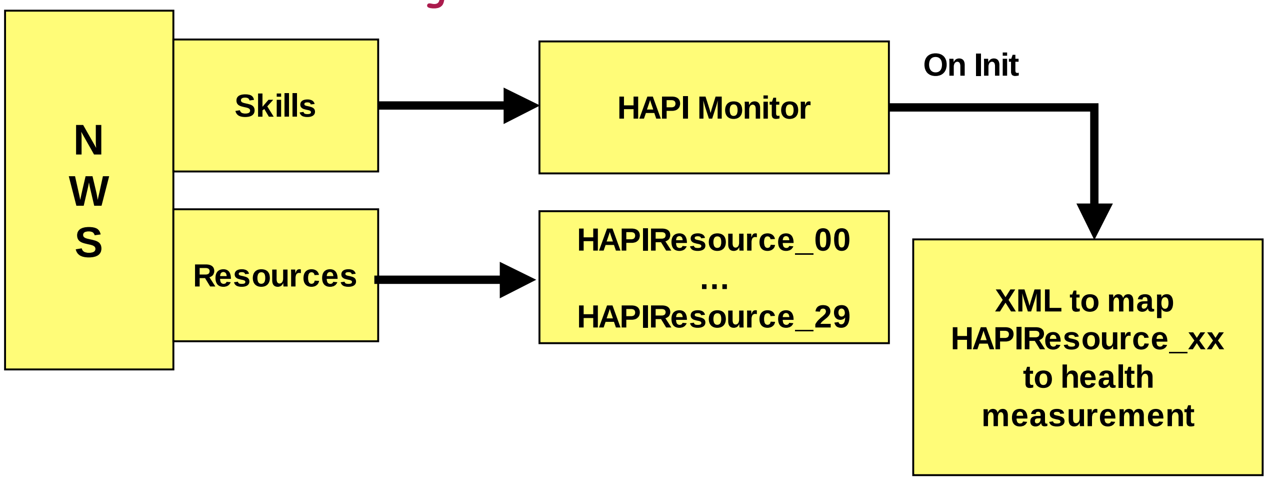
- **Health Application Programming Interface (HAPI): DOE leverage**
  - standard interface for health monitoring
    - Advanced Configuration and Power Management (ACPI)
    - Self Monitoring, Analysis and Reporting Technology (SMART)
    - Intelligent Platform Management Interface (IPMPI)



NWS/HAPI Integration

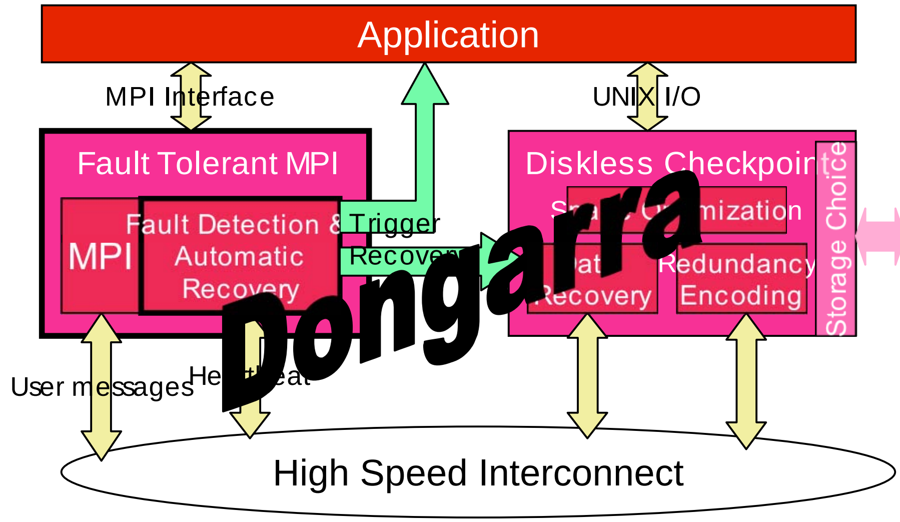
- **Rationale**
  - leverage extant NMI tool infrastructure
  - integrate performance and failure indicator data
  - support "performability" resource allocation in vgES
- **Mechanism**
  - HAPI appears as an NWS sensor with skills and resources
    - NWS statically defines skills and resources
    - HAPI adds dynamically defined health measurements
  - user defines measurement mapping to resources at runtime
    - HAPI ingests these via its API





Large Scale Adaptation Examples

- **Batch queue selection (Wolski)**
  - application fault sensitivity
  - predicted partition reliability
  - expected wait time
- **Checkpoint frequency**
  - application fault sensitivity
  - predicted "bag" reliability
- **Redundancy application**
  - spare nodes (within an application)
  - multiple application copies
- **Power aware code optimization**
  - tuning for power/performance/reliability
- **OS suicide hotline**
  - adaptive personality management



Qualitative Behavioral Classification

- **VGrADS principles**
  - hide unnecessary details (virtual grids)
    - physical resource locations and resource location
  - provide qualitative specifications (vgDL descriptions)
    - e.g., near, far, loose, tight, …
- **Behavioral application characterization goals**
  - similar, high level, qualitative descriptions for applications
    - steady, periodic, random, …
  - temporal specifications for long-running applications
    - differentiate persistent from transient behaviors
  - qualitative assessment of behavior for advertized resources
    - match resource capabilities and application behaviors
    - identify changes in behavioral expectations

Qualitative Behavioral Classification

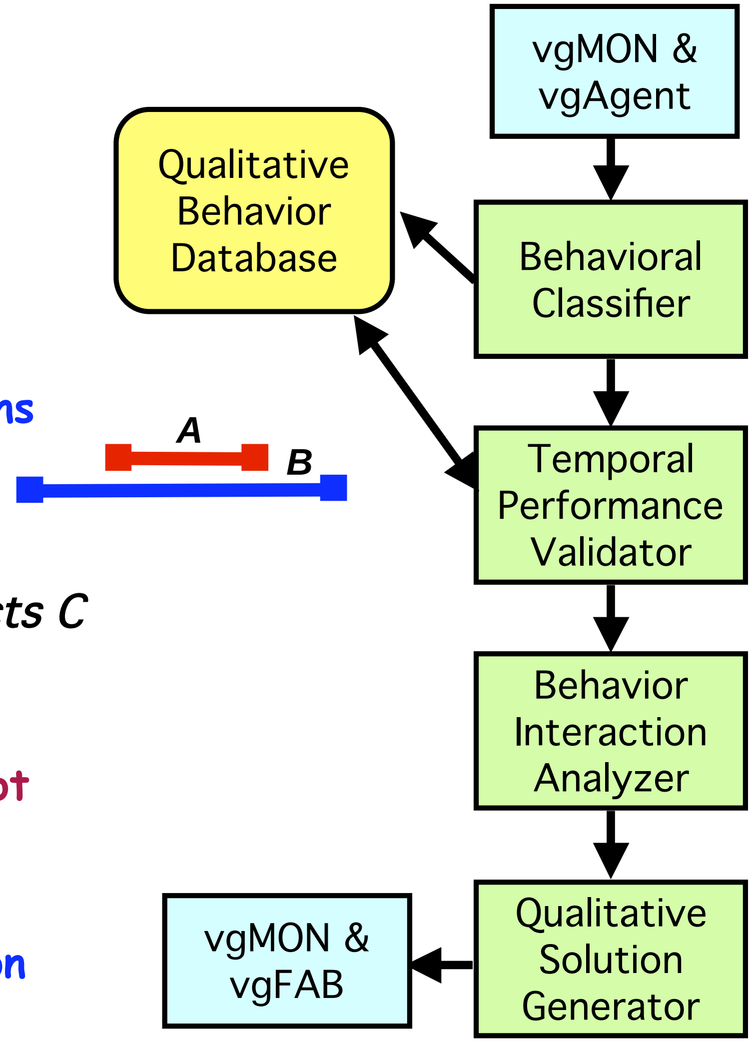
- **Three components**
  - classification of temporal behaviors
    - resource metric axes
      - processor, memory, network, disk
    - behaviors
      - steady, oscillatory, random
  - temporal algebra for application interactions
    - ordering, composition, …

A During B

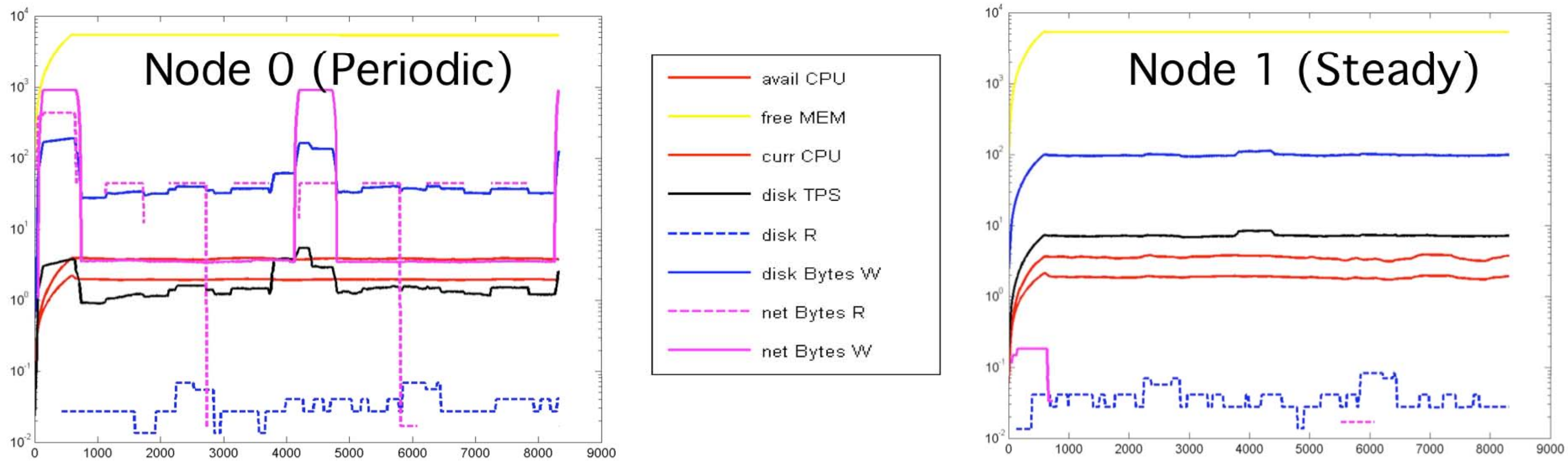
  - methodology for predicting interactions
    - resource overlap

(A Before B) Interacts C

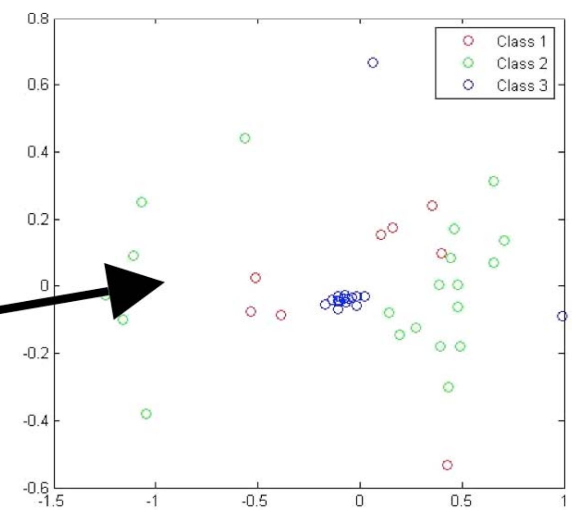
- **Methodology**
  - application stimuli and sensitivity analysis
    - metric data from NWS/HAPI/Autopilot
  - hierarchical/k-means clustering for phases
    - qualitative equivalence identification
  - online behavioral monitoring and remediation
    - virtual grid execution system



WRF Time Series Behavior



- **Clustering (for each metric)**
  - mean and variance
  - first auto-correlation coefficient
- **Behavioral classes**
  1. periodic
  2. steady with variability
  3. steady



Research Challenges and Goals

- LEAD workflow validation and testing
  - vgDL specification and execution
  - "real world" driver for virtual grid development
    - dynamic workflows, streaming data, …
- Multivariate execution system constraints (vgMON)
  - reliability and fault tolerance
    - failure prediction, over provisioning, performanbility
  - performance and power
  - microscale and macroscale management
- Tunable constraints and incomplete resource specification
  - balancing choices
- Behavioral classification (vgFAB)
  - resource selection
  - application behavioral validation

VGrADS Summary: A Holistic Approach

- **What justifies a Large ITR?**
  - community, no one institution covers everything
  - project vision
  - shared infrastructure
  - integration would not happen without a unified project

- **VGrADS**
  - Built on GrADS insights and experiences
    - community of leading researchers who work together effectively
    - broad coverage of requisite topics
  - vision for extremely simple application development interface
    - grid virtualization to hide complexity
  - shared software stack and testbed
    - vgES toolkit, policies and application drivers
  - many interrelated layers require integrated effort
    - program tools, provisioning, scheduling, measurement
    - prediction, fault tolerance, infrastructure, applications