Performance and Reliability Monitoring of the LEAD System

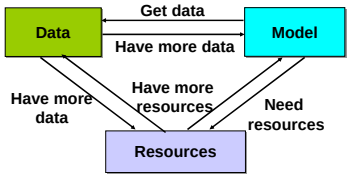
Lavanya Ramakrishnan, Ying Zhang, Kevin Gamiel, Min Yeol Lim, Brad Viviano, Daniel A. Reed

University of North Carolina, Chapel Hill

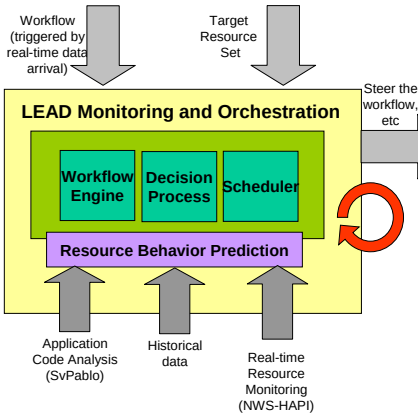
P7

Dynamic Adaptive LEAD System

- Coupled analysis and assimilation tools, forecast models, and data repositories as dynamic adaptive, on-demand services
  - to provide accurate and timely forecasts
    - change configuration rapidly and automatically in response to weather
    - respond to decision-driven user inputs
  - respond to performance variability and failure of resources
  - steer remote observing technologies to optimize data collection
- Monitoring and orchestration challenges
  - streaming data, multilevel workflow, etc
  - dynamic response and iteration based on model outputs



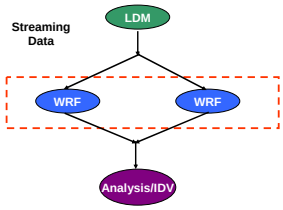
Monitoring and Orchestration Architecture



- Multilevel **monitoring**
  - resource (performance, reliability)
  - workflow status, priority
  - application code analysis
- Performance and reliability **contracts**
  - aggregation of individual levels
  - local guarantee per workflow
  - global optimization of number of workflows being serviced
- Proactive and reactive **adaptation**
  - Ensemble planning and iteration planning based on application and resource behavior
  - Multi-access pattern optimization
    - priorities, conflicting events

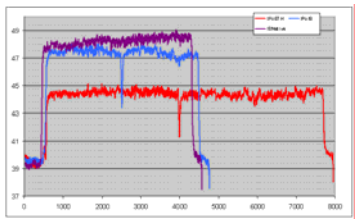
Resource Monitoring Example

Find the best set of resources for data streaming and subsequent WRF ensemble execution



Find the best resources for WRF runs

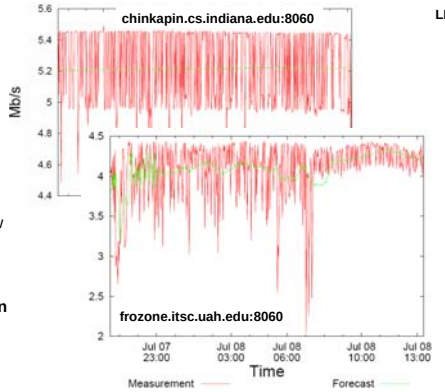
- Performance model
  - CPU utilization and temperature
  - machine characteristics (e.g., uniprocessor kernel, network connectivity)



LEAD Testbed at UNC: WRF runs on clusters with different connectivity. The graph shows the moving average trendline for temperatures with different interconnects

Rank data streaming hosts based on network capacity

- Network bandwidth and latency forecasting
  - parallel access to NWS to improve performance
  - User specifiable parameters
    - timeout, maximum returned hosts
    - bandwidth threshold



LEAD Testbed bandwidthTCP (MB/s) measurements from dante0.renci.org at UNC

WRF run on 16 nodes with different interconnects

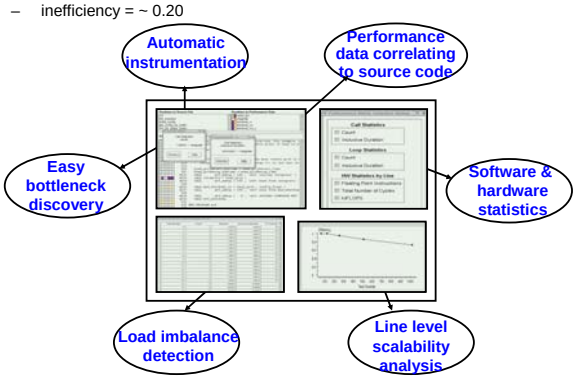
Infiniband Native: 2:18:44 (elapsed time)
IP over Infiniband: 2:32:59
IP over Ethernet: 4:37:46

Monitor the LEAD web services

- SOAP monitoring handler to gather service statistics
  - e.g. number of concurrent users, average service time

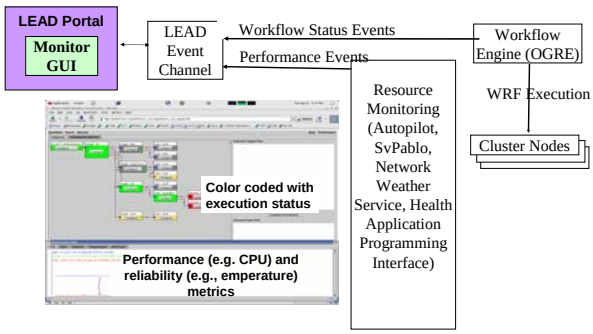
Performance and Scalability Analysis of Weather Research and Forecasting (WRF) using SvPablo

- Convenient graphical tool
  - easy detection of performance bottlenecks in WRF
    - performance statistics (count & durations) and hardware counter statistics at loop and function call level
  - detailed per processor performance data for load balance studies
    - WRF is mostly distributed evenly across processors
- Preliminary scalability analysis
  - on NERSC IBM SP3, 64 to 1024 processors; em_real, medium size case involving a 12 km resolution forecast over the continental US
  - inefficiency = ~ 0.20

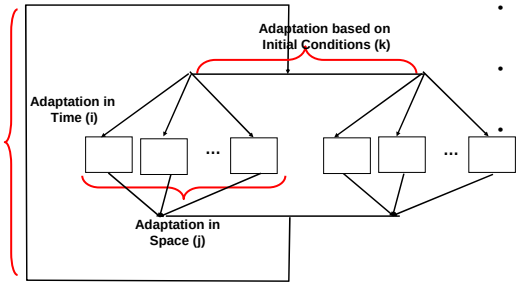


LEAD Monitoring Interface

- Challenges
  - Representation of workflow task to mapped resources
  - Large number of performance events
  - Visual representation of CPU, temperature, etc metrics and failure indication



Future Directions



- Correlate performance and reliability metrics to aid in resource selection
- Develop reliability and performance contracts for LEAD workflows
- Develop a higher level orchestration architecture
  - determine decision points in the workflow
  - estimating initial values for workflow dimensions (l,j,k)
  - adjust values for the workflow dimensions (i,j,k)

This work is supported primarily by the National Science Foundation under the following cooperative agreements: ATM03-31574, 31578, 31579, 31480, 31586, 31587, 31591, and 31594. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

Special thanks to the system administrators of the LEAD testbed, especially Brad Viviano, Marcus Christie, Michael McEniry, Suresh Marru