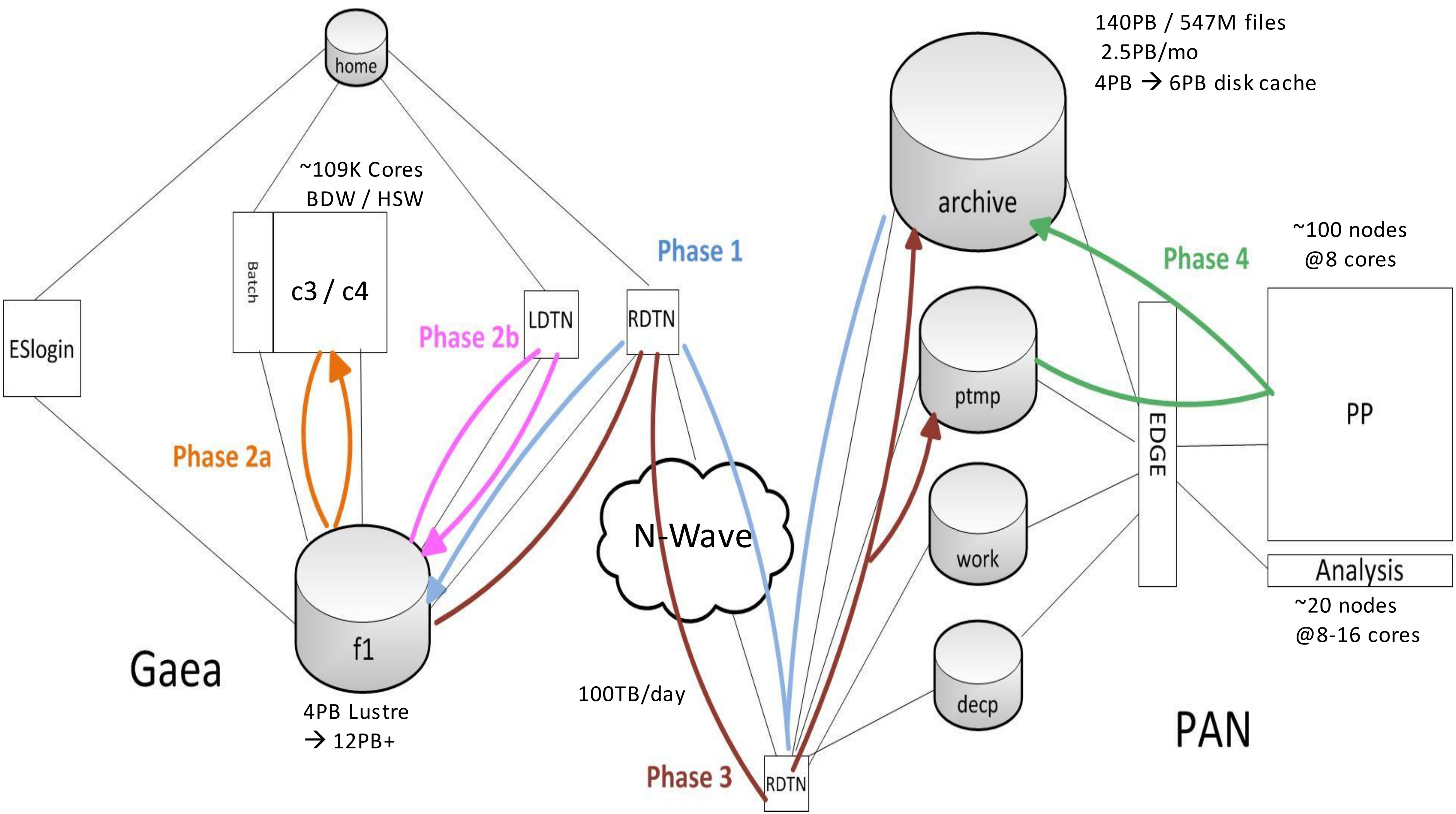
Towards HPC System Throughput Optimization

Jeff Durachta, PhD
Engineering Lead
Modeling Systems Division
Geophysical Fluid Dynamics Lab
Princeton, NJ

Overview

- The Setting: The GFDL Workflow
- Some Existing Tools
  - Just functional descriptions; no details
- Towards Throughput Optimization
  - A Workflow Database
- Conclusions
- Acknowledgments

Workflow



Driven by FMS Runtime Environment automated workflow manager

Some Speeds and Feeds

- Gaea
  - ~109K BDW/HSW cores; ~98K cores devoted to GFDL
  - 4PB -> 12PB Lustre File System
  - CM / ESM: 4-8K cores; 7-14 simYrs/compDay; Use ~100%
  - ~50-70TB/day data egress to GFDL
- GFDL
  - DMF Archive: 140PB @2.5PB/mo; 547MFiles; 6PB disk
  - ~100 post-processing nodes @8 cores
  - ~20 analysis nodes @8-16 cores
  - ~10-20K jobs per day

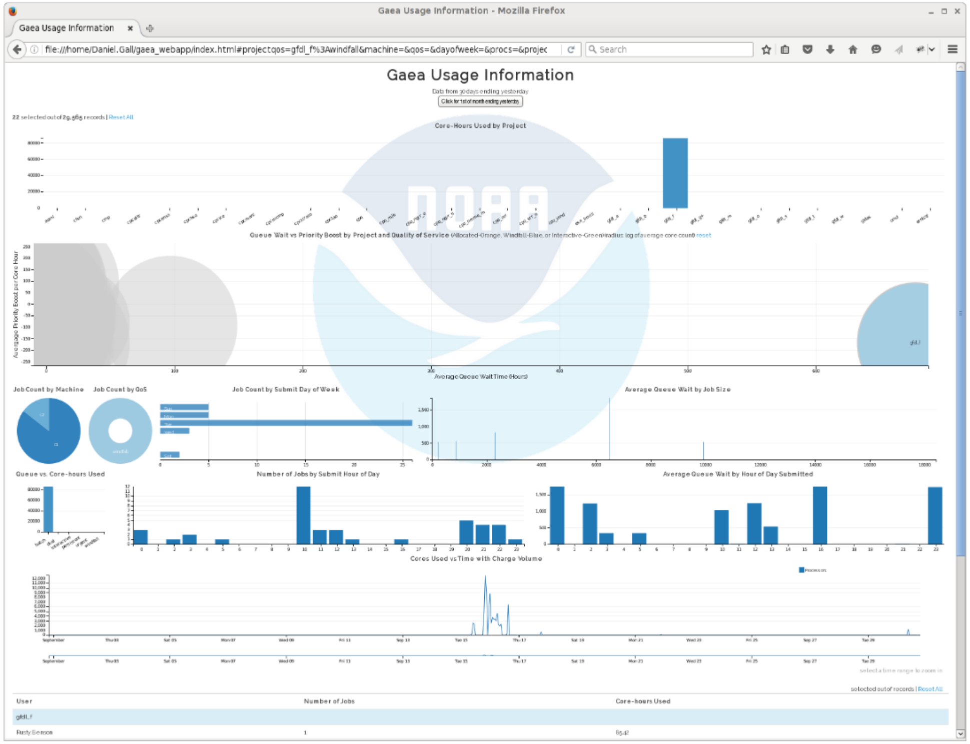
Workflow Data is Everywhere

- GFDL Curator / Model Dev DB Interface
  - (+) Detailed model provenance**
  - (+) Web interface to post-processed results
  - (-) Requires detailed input**
  - (-) Difficult to keep web interface relevant
- G(eneralized) CP (gcp)
  - (+) Centralized log of data transfers
  - (-) Only one aspect of workflow
  - (-) No systematic data mining

GFDL Scheduler Usage Data Visualizer[1]

How did my job fare against other jobs?

- Vendor-provided accounting reporting was inaccurate
- Aggregate data from many sources; validate scheduler and accounting correctness
- Find source of truth for scheduler components
  - and whether "truth" is a variable thing

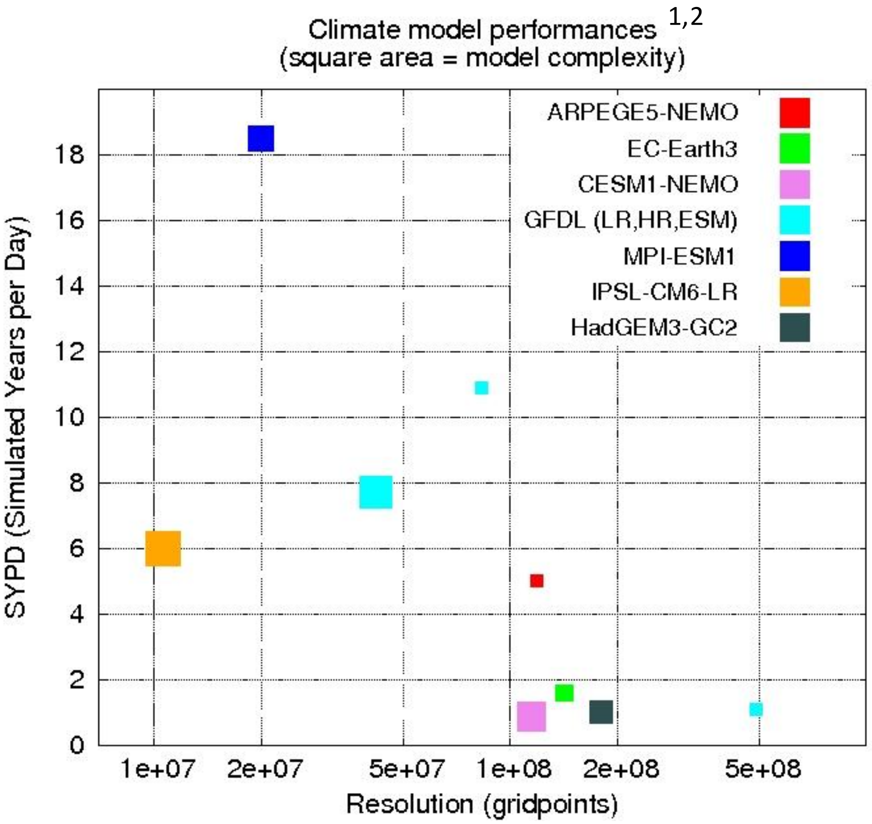


[1]Dan Gall, Engility Corporation

CPMIP

How much resource does my model use relative to its resolution and complexity[3]?

- "Measurements of real computational performance of Earth system models in CMIP6"
  - geosci-model-dev.net/10/19/2017/
  - (+) Evolving CV for CM/ESM Comp Perf & Resource Utilization
  - (-) Only summary aspect of the performance picture



[1] Figure courtesy of V. Balaji

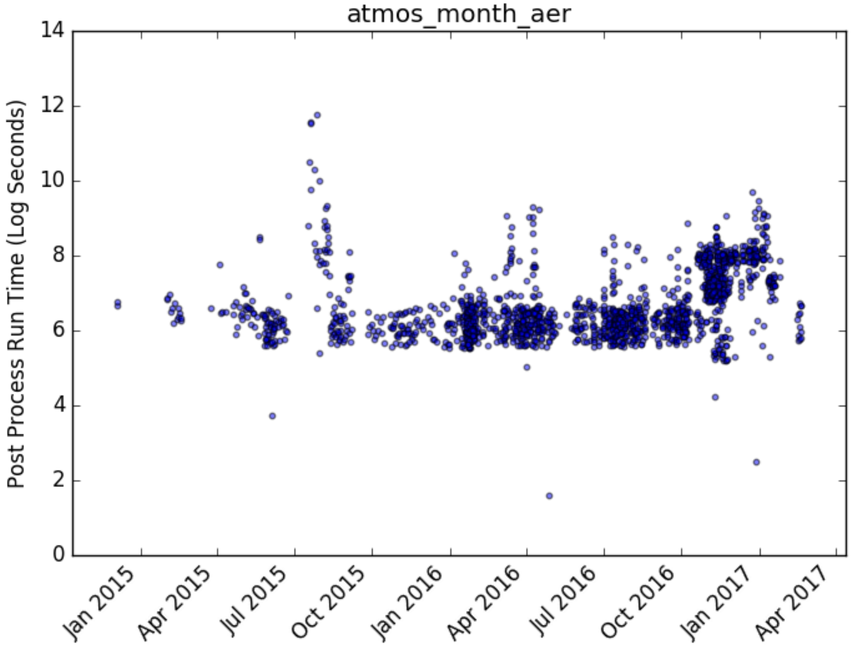
[2] GFDL data from Niki Zadeh, Engility Corp

[3] Complexity ≈ Num prognostic vars

GFDL Post-Process Job Log Scraping[1]

How long until I can see my results?

- Proof of concept
  - Examined 800GB of job logs
  - Goal: Build per sim diag component model of thruput
  - Given complexity of the system, descriptive stats alone insufficient
    - Learn to deal with data gaps
  - Begin to understand and address scaling issues
  - Develop some initial workload models



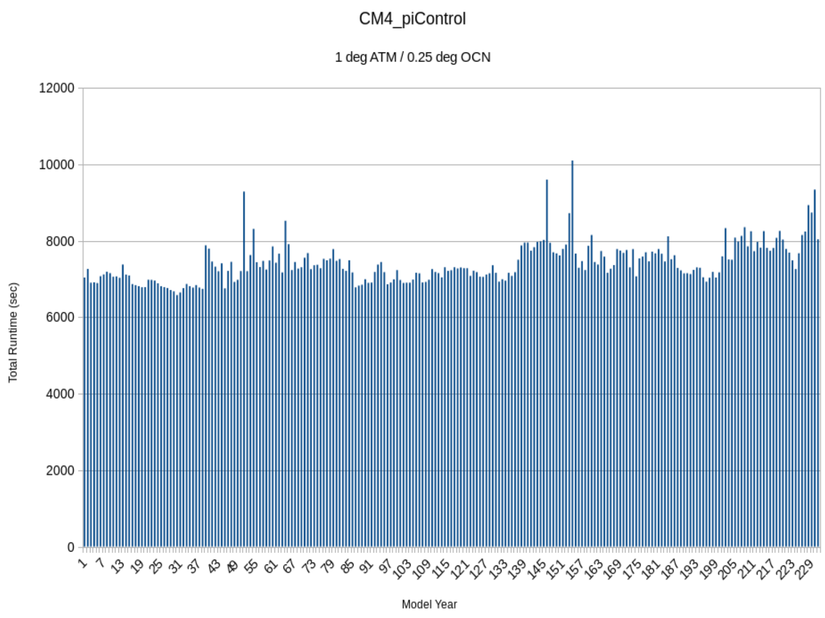
[1] Benjamin Mayer (ORNL)
Joseph Kennedy (ORNL)
Katherine Evans (ORNL)
Jeff Durachta (GFDL)
Published as ORNL/TM-2018/540

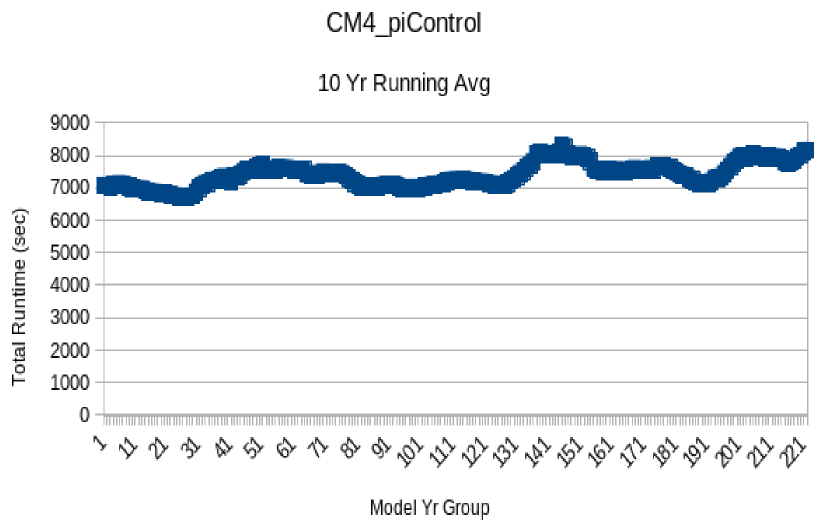
Progress in Earth System Science

- Many aspects require a constant flux of graphs, tables and charts
- Research requires many job runs related by time sequence and/or parameter variations
- Simulation environments have been undergoing enormous (catastrophic?) growth in complexity
- Complex system issues can rob workflows of throughput performance
- Finding root causes for throughput loss are typically time consuming and difficult

CM4_piControl Total Runtime per Sim Yr

A picture is worth….





Workflow Database

Build on the work of others

- GFDL Scheduler Data Visualizer
  - Extract base data set from scheduler
  - Ensure that **all** jobs are represented
- Employ a layered DB model
  - Jobs willing to provide more identification about themselves can record a vast array of related data
    - Curator: an “Experiment”
    - CPMIP: resources used by a class of Experiments
    - JLS[1]: throughput performance
      - Experiment end-to-end
      - Across entire system platform

[1]Job Log Scraping

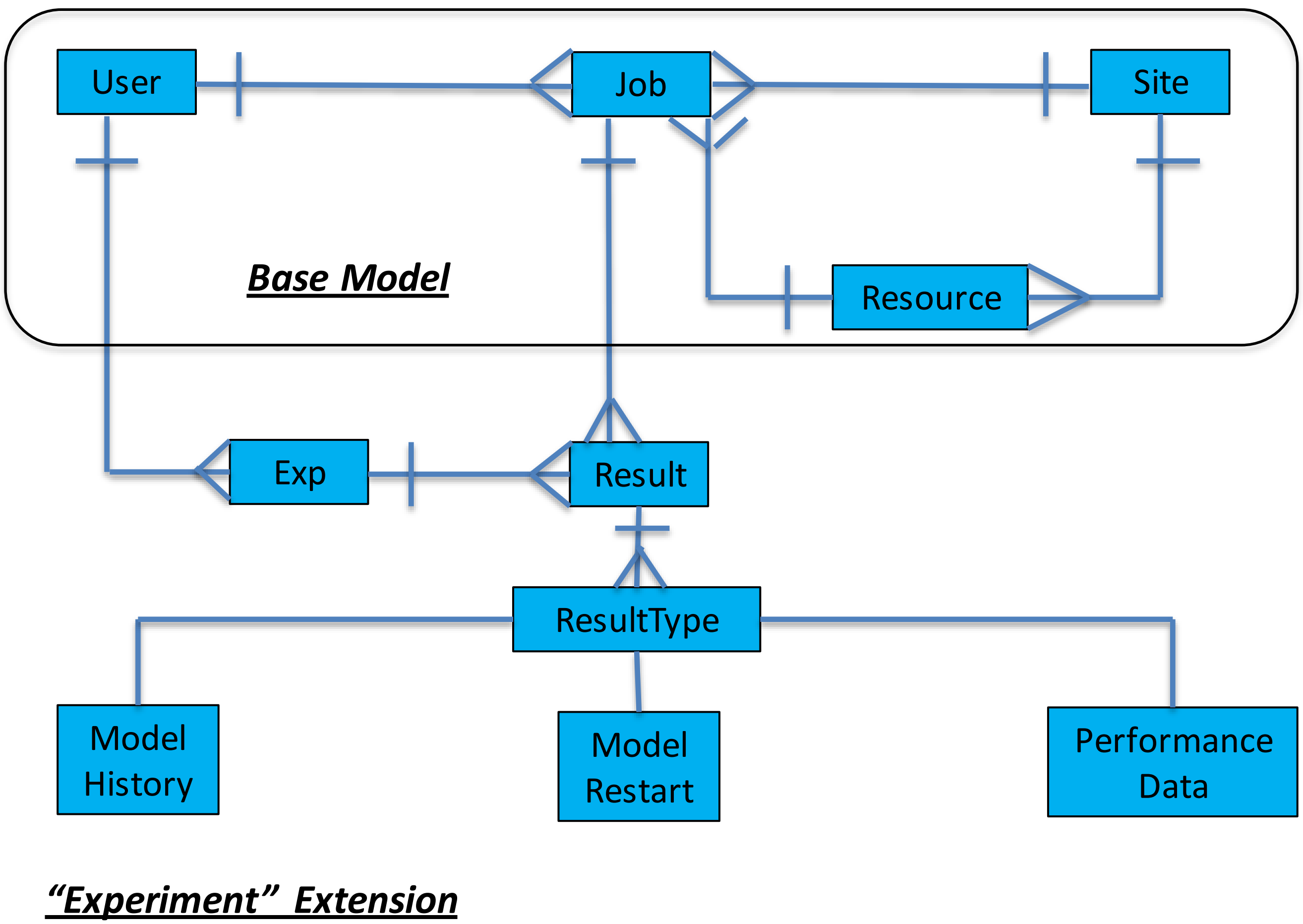
Potential Advanced Capabilities

- Enable new levels system debugging
  - Enhance workflow toolset logging capabilities
  - Cross correlate with other system logs
  - Answer: Who? Why? What?
- Enable predictive analysis
  - Trend analysis to influence preventive maintenance
  - Data driven decisions on system upgrades

[1]Job Log Scraping

Some Possible Entities & Relationships

No Attempt at Normalization



Conclusions

- We must transform “islands of data capture” into a comprehensive workflow data gathering and analysis infrastructure
- Design requirements
  - Light weight; non-intrusive; comprehensive
  - Modular, encapsulated; extensible
  - Deployed in stages; build complexity from simplicity
- Goal: Understand and optimize scientific data production throughput
  - See through the ever increasing volume and complexity

Workflow Data is Everywhere

Enable exascale science;

Not run just exascale models

Acknowledgements

The GFDL Workflow Team:

V Balaji[1], Chris Blanton[2], J Durachta[3], Colleen McHugh[2], Sergey Nikonov[1], Aparna Radhakrishnan[2], Seth Underwood[3], Chan Wilson[2], Hans Vahlenkamp[4]

Technical Systems Workflow Support:

Dan Gall[2] (Scheduler Data Visualizer)

General Workflow Data Capture

Philip Mucci[5]

[1] *Princeton University*
[2] *Engility Corporation*
[3] *US Federal*
[4] *University Corporation for Atmospheric Research (UCAR)*
[5] *Minimal Metrics, LLC*