Measuring and forecasting ESMs

Chandin Wilson, Chris Blanton, Curtis Bechtel,
Dan Gall, Engility
V. Balaji, Princeton
Jeff Durachta, Seth Underwood, NOAA GFDL

Overview

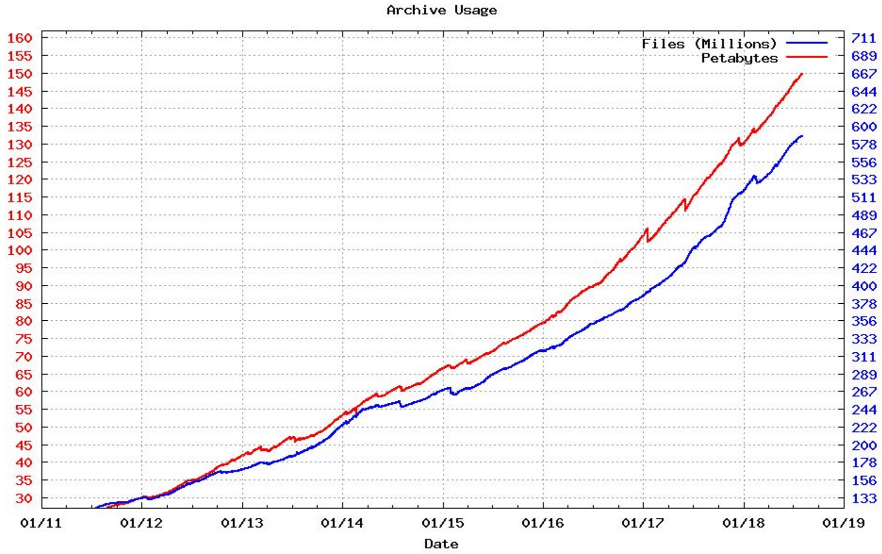
- Requirements gathering challenges: ESM specific problems
- How can we know?
- What do we do?

What is needed?

- More understanding of ESM use of system resources
- Diagnosing individual jobs and job streams
- Identify pervasive system performance issues
- Provide the best answers to future resource needs

What does this mean?

- How to provide the best, balanced answer to the questions of:
  - More memory / cores / cpus
  - I/O characteristics
  - File system utilization



ENGILITY
Engineered to Make a Difference

No single correct answer

- Typical HPC benchmarks do not reflect ESMs
- ESMs are composed of recursively complex components
- Performance can vary dramatically according to resolution, complexity, etc

ESMs are their own enemy

- An ESM competes with itself and others in the HPCS
- Different ESM components use different, resource-use-conflicting tactics
    - What is best for MPI is not best for OpenMP
- Scheduling resources is a vague approximation

Engineered to Make a Difference

Component layout of three ESMs

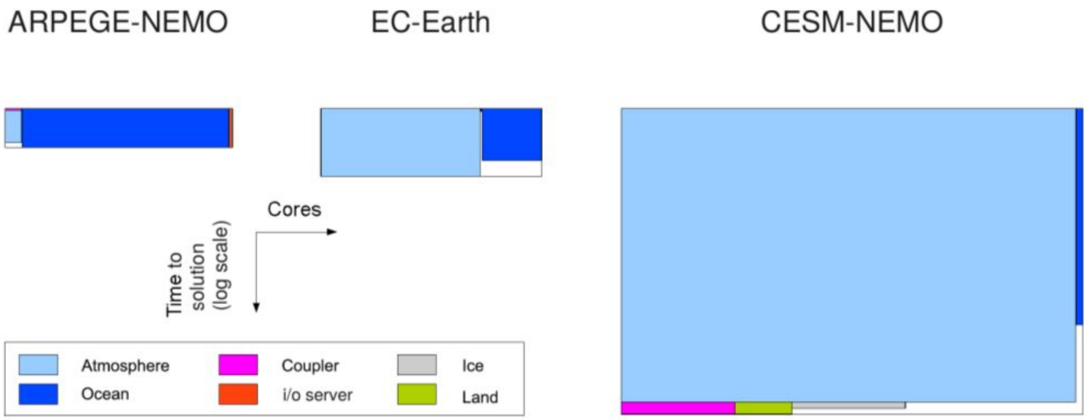


Figure 2. Component layout of three ESMs, in processor-time space (time increasing downward). Each box represents a component which is integrated either concurrently (coarse-grained concurrency; see text), in which case it is shown alongside the other components running at the same time, or sequentially, in which case it is shown below the previous component. The bounding rectangle shows the total cost of the coupled system, including waiting times due to load imbalance. Adapted from Fladrich and Maisonnave (2014).

CPMIP: measurements of real computational performance of Earth system models in CMIP6. *Geosci. Model Dev., 10, 19-34, 2017.* By V. Balaji *et al.*

ESM computational performance

- Efficiency curves: models can run for speed or capacity
- Compute cost: degrees of freedom = resolution + complexity
- Coarse-grained concurrency: components have their own grids, time steps, parallelization methods, computation profile

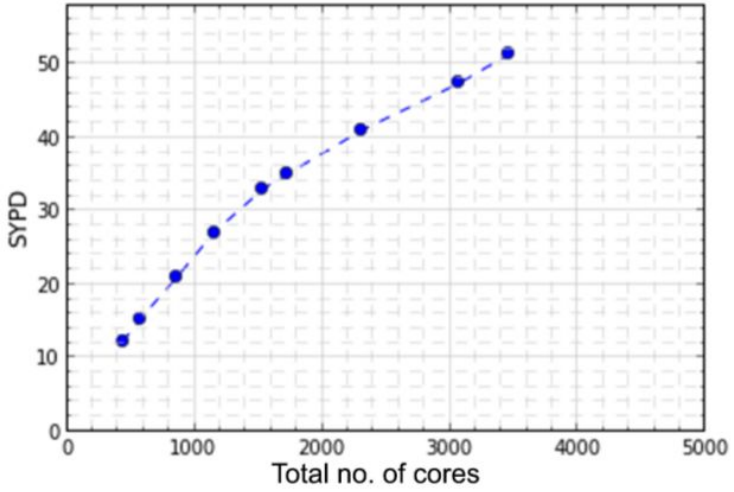


Figure 3. Scaling behaviour of a GFDL model. It illustrates that the model could be run at 50 SYPD in capability, or speed mode, but in practice is most often run at the shoulder of the curve, at around 35 SYPD, which gives the best throughput.

Find The Balance

- **Understanding the HPCS profile and ESM component mix is key to balancing:**
    - Overall system architecture
    - Scheduler allocations
    - Model resource requests

ENGILITY
Engineered to Make a Difference

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE

How to begin: Knowledge Is Power

- Coarse metrics such as user/cpu/memory (`time COMMAND`)
  - Timing a shell wrapper that calls a java applet which calls a python script which does netcdf operations
- O/S and I/O metrics across HPC
  - Cluster metrics seldom cross-referenced to jobs or experiments
- Instrument the ESM

Log everything

- Establish and share the log and metric repositories
  - Centralized tool/application-level syslog
  - HPCS O/S and I/O metrics
  - Scheduler logs

Engineered to Make a Difference

Know your HPCS

- Constantly gather HPC health
  - Historical and real time
  - O/S viewpoint across all nodes
    - Orphaned processes holding memory or cpu hostage
  - Interconnect fabrics
    - Are those fibers really clean?
  - I/O metrics, especially cluster filesystems

Know your scheduler

- Gather and understand job scheduler metrics
  - Individual jobs
  - Collections of jobs by experiment
  - GFDL Interactive Scheduler Visualizer (ISV)

Know your ESM

- Parse model output (Job Log Scraping)
  - very workflow centric method to extract useful metrics from stdout logs
- Tool / application metrics
- Instrument the ESM: CPMIP

ENGILITY
Engineered to Make a Difference

Know it all: Knowledge is Power

- Gather all the pieces together into a “workflow database”
- Across the entire lifecycle of experiment:
  - HPCS, scheduler, workflow metrics
  - CPMIP metrics, for compliant ESMs
  - JLS for throughput performance

ENGILITY
Engineered to Make a Difference

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE

How to get there: A Workflow database

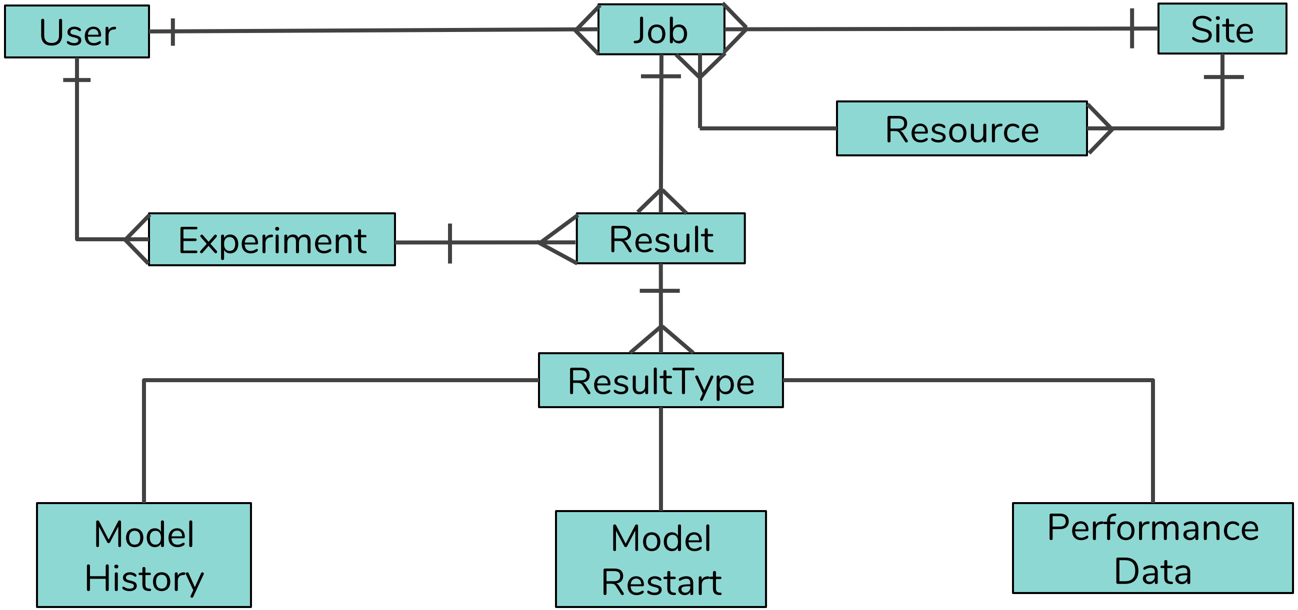
- Employ a layered DB model
  - Jobs willing to provide more identification about themselves can record a vast array of related data
- CPMIP for resources used by the ESM
- JLS for throughput performance
- Gather from the experiment, start to finish
- Across entire system platform
- Enhance workflow toolset with logging capability (a la gcp)
- Enable system debugging and predictive analysis

What do we gain?

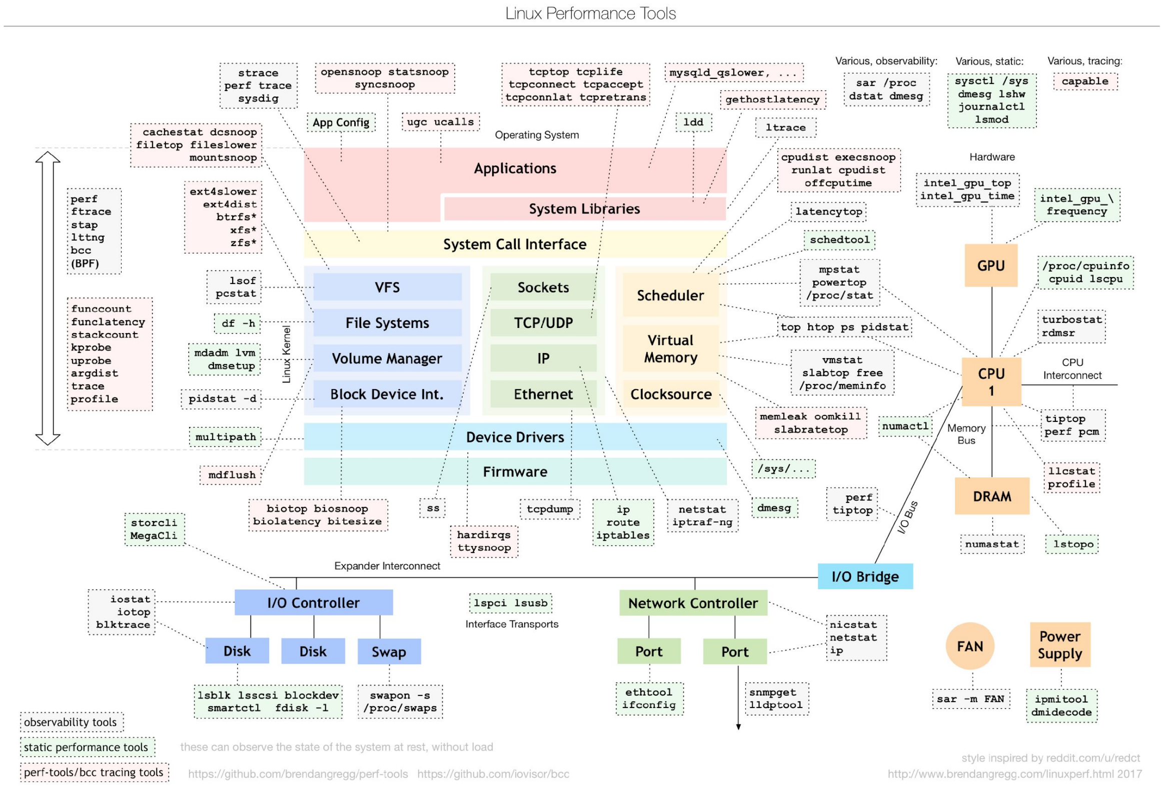
- Ability to debug specific job issues easier
    - Individual job details
    - Individual job across system context
- System health
    - Metrics on specific indicators
        - Data transfer effective bandwidth
        - Queue wait times
        - Job failure rates
- Visibility to HPCS and ESM performance over time
    - Over preceding 3,6,12,24 hours, weeks
    - Trends over months
- System resource utilization analysis
    - By user and/or group
    - By job stream (experiment)
    - Trends and projections

ENGILITY
Engineered to Make a Difference

Possible entities and relationships



Focus on: HPC Metrics



Focus on: HPC Metrics with PerfMiner

- Get relevant data at thread/process boundaries
- Connect the data with relevant context and record scope
    + Thread/process/job/queue/host/system
    + CPU, Threading, I/O, MPI, Memory, OS, NUMA
    + System configuration
    + Sysctl, /proc, file systems
    + Environment variables
    + Modules
    + Job script, parameters, status, output
    + Process/thread tree
    + Application calipers and output integration
- Store in NoSQL database, analytics backend
- Visualize via the browser, unique URL to every view

ENGILITY
Engineered to Make a Difference

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE

Focus on: Application / tool centralized logging

- Message brokers are lovely, but syslog is a fine lowest-common-denominator
  - “Application syslog server” listening on a non-standard port
  - Easy to use from Python, Perl, Java, Go, Bash
  - Leverage logging frameworks: log4j
  - Use key=value;key=value messages for ease of Splunk or ELK parsing
  - Generate a unique ID (`uuid, uuidgen`) key for each tool run
  - If all else fails:

    ```
    bash -c “echo key=value;key=value > /dev/udp/loghost/5514”
    ```

ENGILITY
Engineered to Make a Difference

NOAA
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

Focus on: Log examples

```
ts=2018-09-13T08:48:08.262;guid=bd1bc034-5471-4fa5-a047-aa5adf300ed9;p=INFO;
where=GCP::Driver::new.303;event=gov.noaa.rdhpcs.gcp.start;MOAB_JOBID_internal=gfdl.18522579;
PBS_JOBID_internal=19160605.moab01.princeton.rdhpcs.noaa.gov;
cwd=/vftmp/Yujin.Zeng/pbs19160605/fre/warsaw/ESM2Mb/ESM2Mb_pi-control_C1_dem3/gfdl.ncrc3-intel16-prod-openmp/ESM2Mb_pi-contr
ol_C1_dem3_20080101/work;
gcp_call=/usr/local/gcp/2.3.11/gcp -v
/vftmp/Yujin.Zeng/pbs19160605/tempCache/ocean_topaz_wc_btm/ts/monthly/4yr/ocean_topaz_wc_btm.200501-200812.ffedet_btm.nc .;
gcp_version=2.3.11;level=info;node=pp034.princeton.rdhpcs.noaa.gov;pid=15910;
prog=/usr/local/gcp/2.3.11/gcp;system_load1=0.26;user=Yujin.Zeng;
ts=2018-09-13T08:48:09.359;guid=bd1bc034-5471-4fa5-a047-aa5adf300ed9;p=INFO;where=GCP::Common::verbose.259;level=info;messag
e=Transfer took 1 seconds; 13.76MB/sec ;

ts=2018-09-13T08:48:09.361;guid=bd1bc034-5471-4fa5-a047-aa5adf300ed9;p=DEBUG;
where=GCP::Driver::log_end.1421;event=gov.noaa.rdhpcs.gcp.end;error=none;file_count=1;gcp_call=...;
level=info;node=pp034.princeton.rdhpcs.noaa.gov;pid=15910;prog=...;status=0;
transfer_size=14425864;transfer_time=1;transport_count=1;user=Yujin.Zeng;
```

ENGILITY
Engineered to Make a Difference

NOAA
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

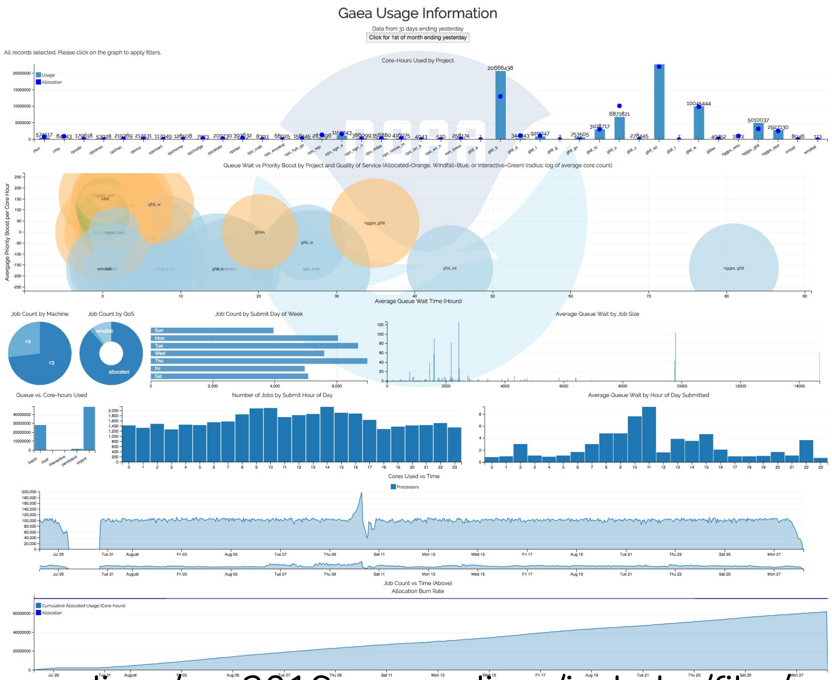
Focus on: Log examples

```
ts=2018-09-12T23:34:16.537;guid=7b8ed3e1-725e-465f-a31c-e36cc33b14d0;p=INFO;where=GCP::Driver::new.303;event=gov.noaa.rdhpcs
.gcp.start;MOAB_JOBID_internal=;PBS_JOBID_internal=5567900.moab03.ncrc.gov;cwd=/lustre/f1/Lori.Sentman/verona/ESM2G_pi-contr
ol_C2.1000mocean/ncrc3.intel15-prod-openmp/stdout/run;gcp_call=/ncrc/usw/gcp/local/opt/gcp/2.3.11/gcp --create-dirs
--verbose /lustre/f1/Lori.Sentman/verona/ESM2G_pi-control_C2.1000mocean/ncrc3.intel15-prod-openmp/history/02110101.nc.tar
gfdl:/archive/Lori.Sentman/verona/ESM2G_pi-control_C2.1000mocean/gfdl.ncrc3-intel15-prod-openmp/history/;gcp_version=2.3.11;
level=info;node=rdtn06.ncrc.gov;pid=25445;prog=/ncrc/usw/gcp/local/opt/gcp/2.3.11/gcp;system_load1=1.62;user=Lori.Sentman;

ts=2018-09-12T23:35:10.166;guid=7b8ed3e1-725e-465f-a31c-e36cc33b14d0;p=INFO;where=GCP::Common::verbose.259;level=info;messag
e=Transfer took 54 seconds; 110.21MB/sec ;

ts=2018-09-12T23:35:11.141;guid=7b8ed3e1-725e-465f-a31c-e36cc33b14d0;p=INFO;where=GCP::Driver::log_end.1423;event=gov.noaa.r
dhpcs.gcp.end;dtn_destination=dtn-003.princeton.rdhpcs.noaa.gov;error=none;file_count=1;gcp_call=...;gcp_version=2.3.11;leve
l=info;node=rdtn06.ncrc.gov;pid=25445;prog=/ncrc/usw/gcp/local/opt/gcp/2.3.11/gcp;status=0;transfer_size=6240600064;transfer
_time=54;transport_count=1;user=Lori.Sentman;
```

Focus on: Workflow task metrics

```
Sums
Count What                Duration         User      Sys      Memory
-------------------------------------------------------------------------------------------
7     get                 00:09:20.3596    32.30     1.94     598M
7     getHistoryFile      00:09:14.4776    35.74     9.65     191M
114   gcp                 00:06:53.2195   108.29    26.21     3.6G
35    stage               00:03:56.4770   161.29    14.22     3.0G
838   timavg.csh          00:02:19.1930    93.00     8.62     7.9G
79    split               00:02:16.8187    57.22    20.15     2.0G
9     timeaverage         00:02:15.8146    41.33     2.09     774M
1326  ncks                00:01:31.9639    28.29    15.29     8.5G
6     zinterp             00:01:27.7668    30.12     1.22     514M
3     timeseries          00:01:25.3784    27.44     0.98     259M
575   zinterpShard        00:01:24.5581    48.09    11.96      19G
13    put                 00:01:14.4513    52.80     2.51     1.1G
919   ncrcat              00:01:05.2191    22.77    11.49     5.0G
924   splitMonth          00:00:52.4225    20.00     6.49     6.1G
7     splitVar            00:00:20.8035     7.77     5.97     138M
76    ncdump              00:00:05.9765     1.13     0.00     289M
24    list_ncvars.csh     00:00:02.6120     0.64     0.20     107M
1     /home/Erik.Mason/Scripts/utils/split_ncvars/split_ncvars.py 00:00:01.6632    0.37    0.27      20M
1     split_ncvars.py     00:00:01.6632     0.37     0.27      20M
5     ncatted             00:00:00.2179     0.07     0.00      24M
```

Focus on: GFDL ISV



https://cug.org/proceedings/cug2016_proceedings/includes/files/pap149s2-file1.pdf

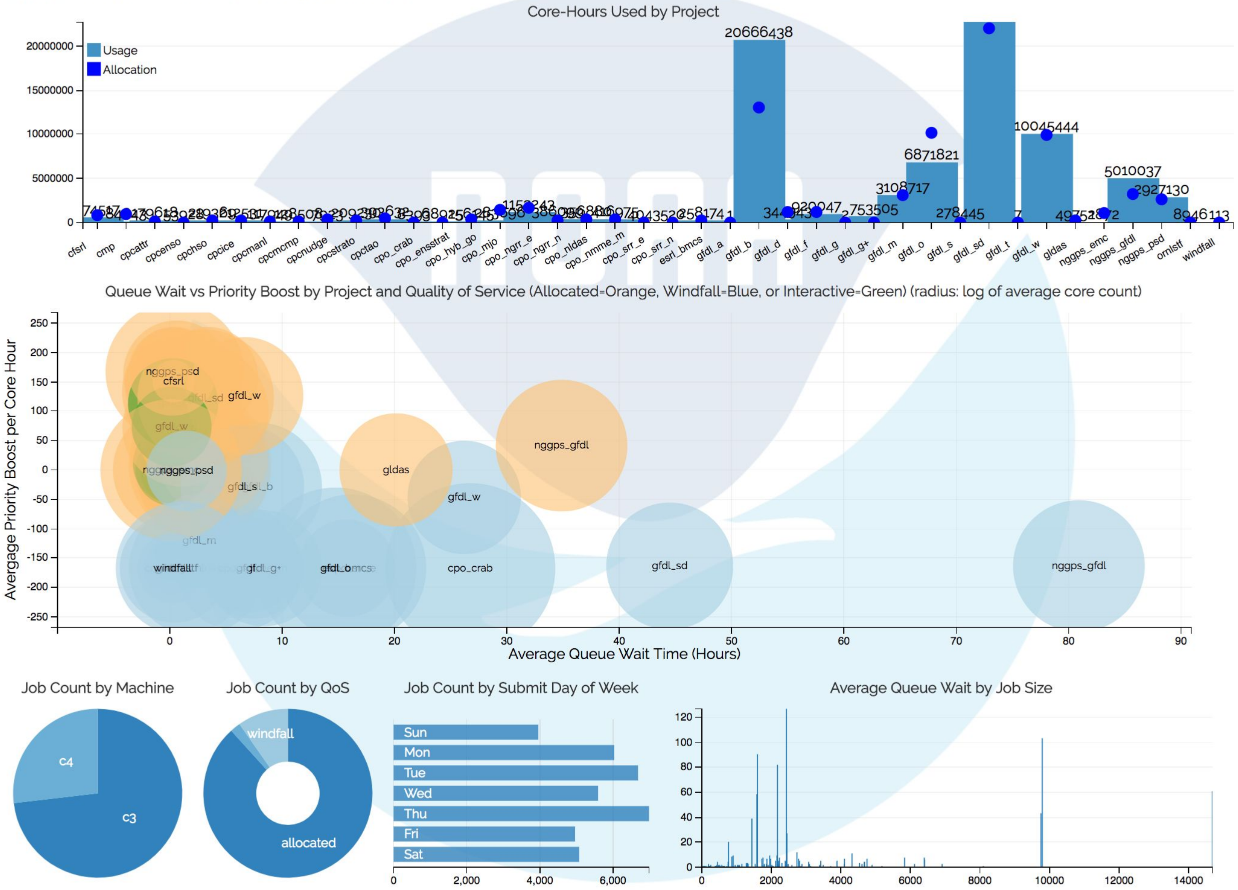
Engineered to Make a Difference

Gaea Usage Information

Data from 31 days ending yesterday

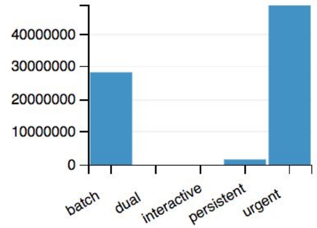
Click for 1st of month ending yesterday

All records selected. Please click on the graph to apply filters.

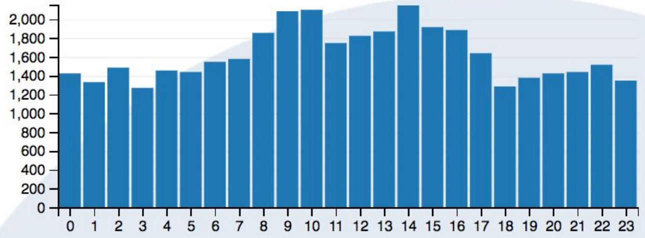


Engineered to Make a Difference

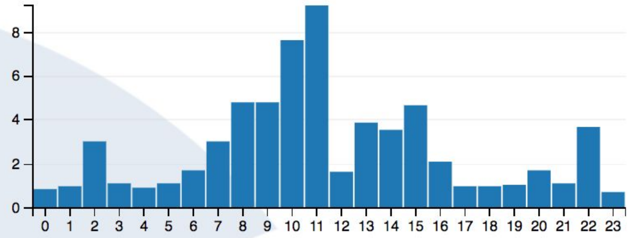
Queue vs. Core-hours Used



Number of Jobs by Submit Hour of Day



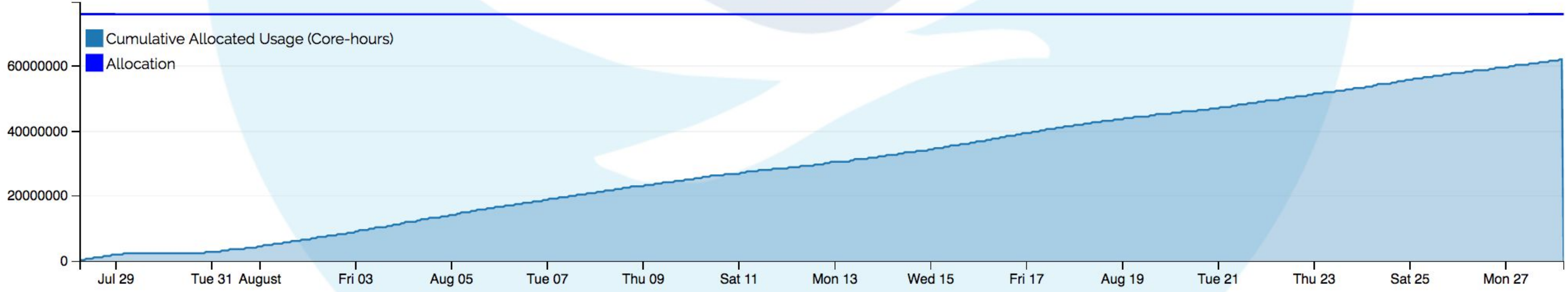
Average Queue Wait by Hour of Day Submitted



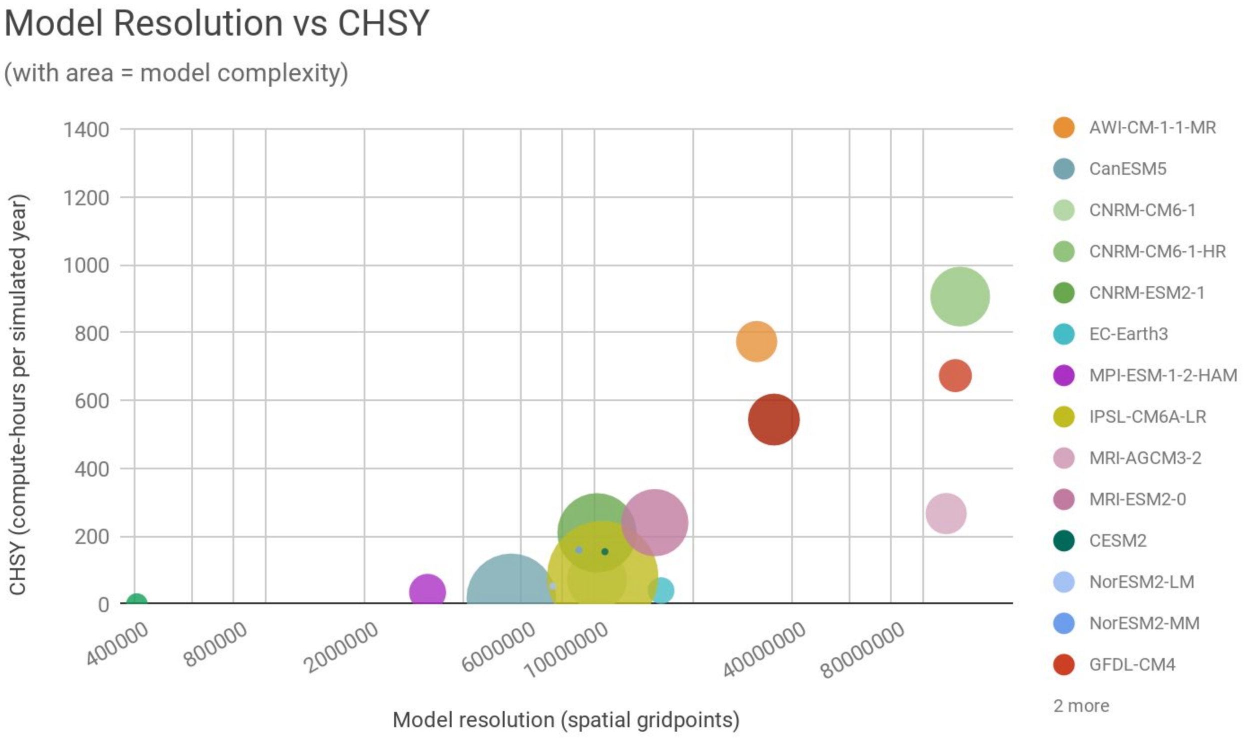
Cores Used vs Time

Job Count vs Time (Above)

Allocation Burn Rate



Focus on: Measure the ESM: CPMIP



ENGILITY
Engineered to Make a Difference

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE

CPMIP: Computational Performance MIP

- Available from current ESMs
- Represents <u>actual performance</u> as run in a science setting
- Lifecycle of ESM run, covering both data and computational load
- Computational cost of ESM
  - Simulated Years Per Day: SYPD
  - Core Hours Per Simulated Year: CHSY
  - $/€ cost per ESM

CMIP6 and the CPMIP Metrics

- CMIP6: The Coupled Model Intercomparison Project Phase 6
- CPMIP: measurements of real computational performance of Earth system models in CMIP6. *Geosci. Model Dev., 10, 19-34,* 2017. By V. Balaji and 5 GFDL and 9 other co-authors
- Modeling groups will be asked to submit CPMIP metrics to the CMIP6 archive along with the model output and other metadata
- Each CMIP6 simulation will have a metadata “landing page”, and its URL will be within the model output NetCDF attributes.

Focus on: Job Log Scraping

- Performance Analysis of Large Scale HPC Workflows for Earth System Models
- ORNL/TM-2017/540
- https://info.ornl.gov/sites/publications/Files/Pub104382.pdf
- Proof of concept
    - Examined 800GB of job logs
    - Goal: Build per sim diag component model of thruput
    - Begin to understand and address scaling issues
    - Develop some initial workload models

ENGILITY
Engineered to Make a Difference

NOAA
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

Harvesting metrics from job output

Category	Metric	Harvester source
model	resolution (number of grid points) and complexity (number of prognostic variables)	restart files
platform	core count, clock speed, clock cycle concurrency	static per compute platform
compute cost	simulated years per day (SYPD), actual SYPD (ASYPD), core hores per SY (CHSY), Joules per SY, number of cores	model log files
coupling cost	total runtime and number of cores, runtime and number of cores for each component	model log files
other	data intensity (amount of data per compute-hour)	history files

ENGILITY
Engineered to Make a Difference

NOAA
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

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 from simplicity to complexity
- Goal: Understand and optimize scientific data production throughput
  - See through the ever increasing volume and complexity
  - Enable peta-scale science, not just peta-scale models

ENGILITY
Engineered to Make a Difference

NOAA
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

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*

ENGILITY

Engineered to Make a Difference

NOAA

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

U.S. DEPARTMENT OF COMMERCE