HPC Workflow Performance

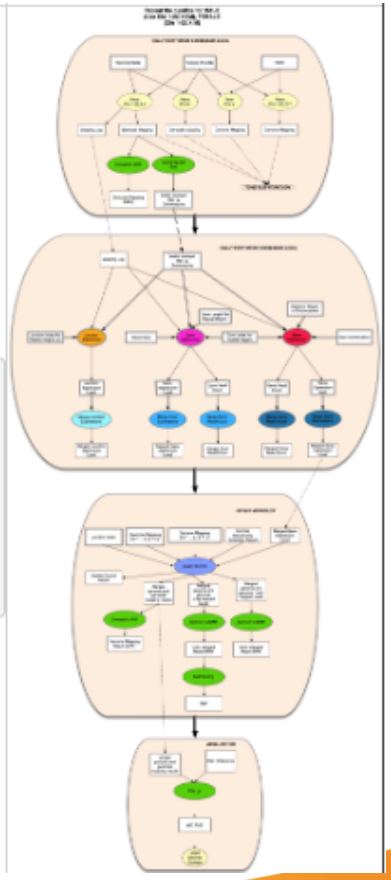
Karen L. Karavanic
New Mexico Consortium & Portland State University
David Montoya (LANL)

August 2, 2016

UNCLASSIFIED - LA-UR-16-23542

What is an HPC Workflow ?

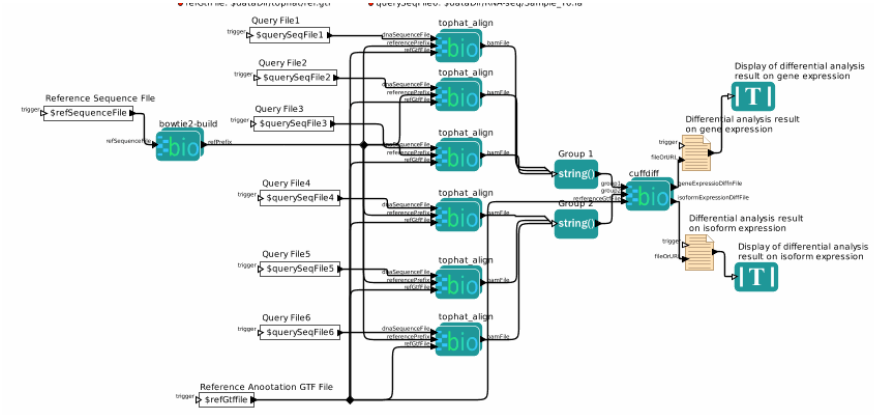
- *Application View*
  - Runtime system for single platform to schedule and run a large number of lightweight tasks
  - Node = task ; edge = dependency/data
  - Ex: Pegasus*: Mapper, Execution Engine, Task Manager, Monitoring



**E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny, and K. Wenger, "Pegasus: a Workflow Management System for Science Automation," Future Generation Computer Systems, vol. 46, pp. 17-35, 2015.*

What is an HPC Workflow ?

- *Experiment View*
  - Grid- or web-based workflow management system schedules across platforms and locations for one science experiment
  - Focus: selecting existing components, reformatting the data between steps, mapping of components to resources (Condor class-ad), Provenance (for scientific needs)



Ex: bioKepler*: Cloud or Grid Platforms

**J. Wang, Crawl, D., and Altintas, I., "A Framework for Distributed Data-Parallel Execution in the Kepler Scientific Workflow System", in 1st International Workshop on Advances in the Kepler Scientific Workflow System and Its Applications at ICCS 2012 Conference, 2012.*

What is an HPC Workflow ?

Holistic View

- One science effort across a period of time/campaign, or for 1 specific goal – may include multiple platforms or labs
- Track resource utilization, performance, and progress, data movement
- Includes System Services – power, resource balance, scheduling, monitoring, data movement, etc.
- Includes Data Center – power, cooling, physical placement of data and jobs
- Informed by & Interfaces with the Application and Experiment Views
- Includes hardware, system software layers, application

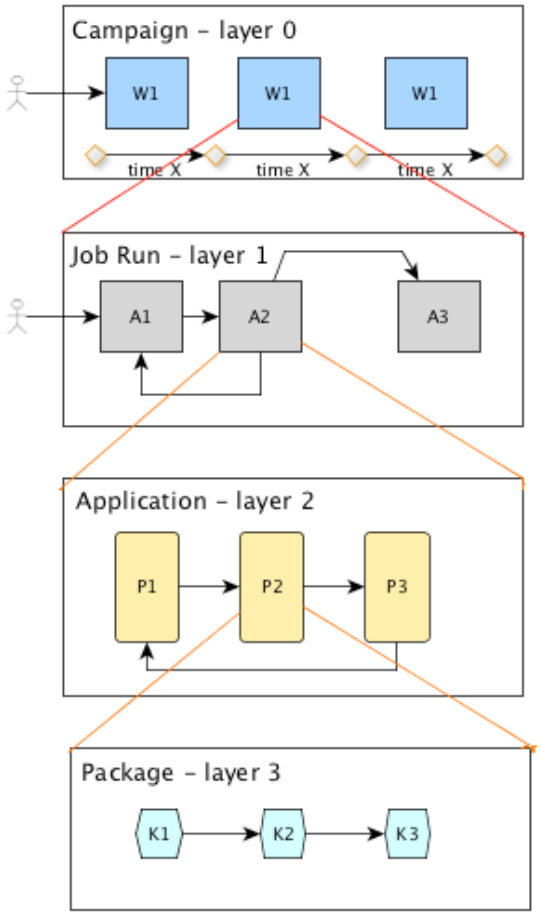
Foundational Work: All Layers of Workflow and their Relationships

Layer 0 – **Campaign**
- Process through time of repeated Job Runs
- Changes to approach, physics and data needs as a campaign or project is completed - Working through phases

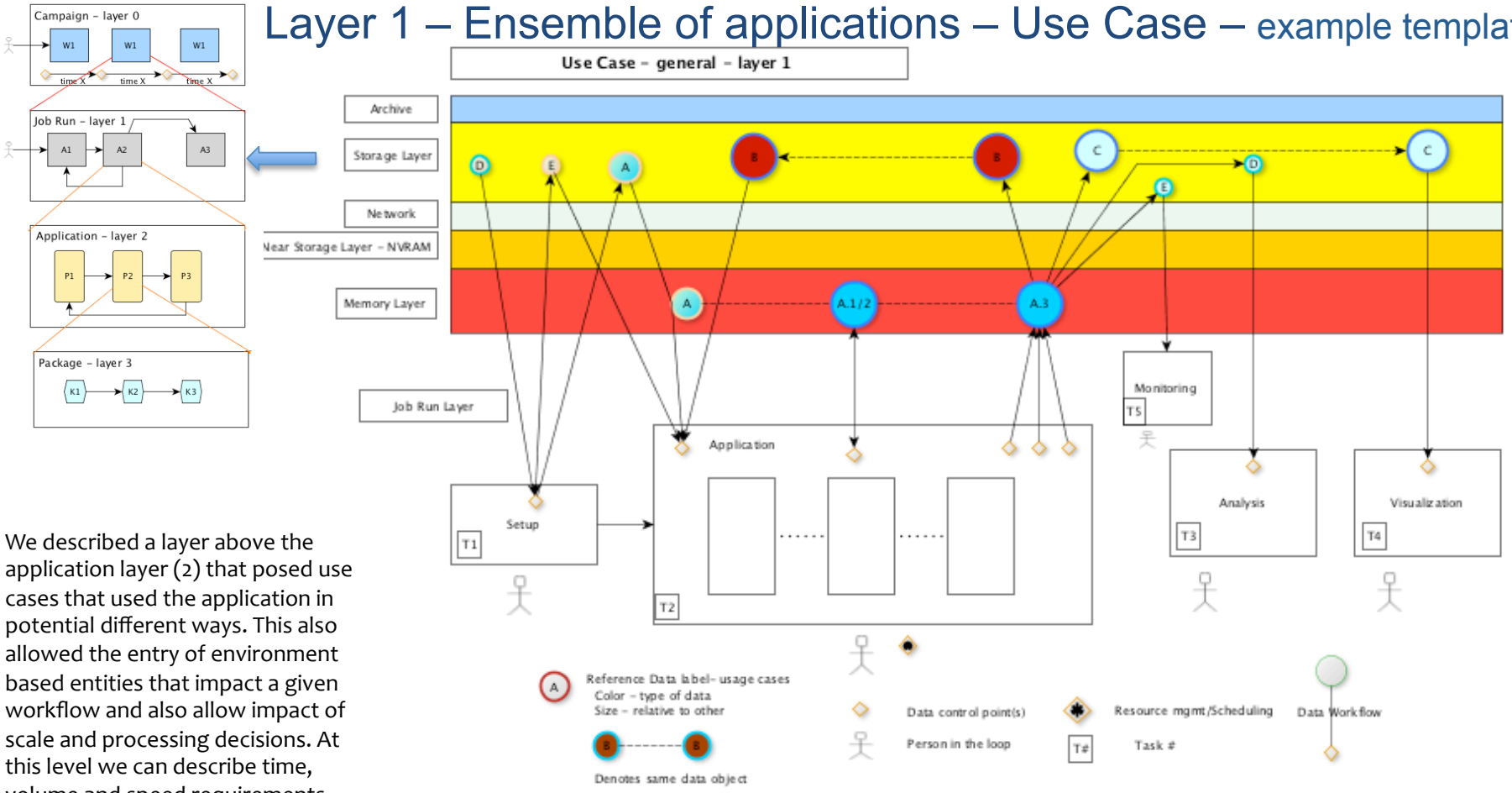
Layer 1 – **Job Run**
- Application to application that constitute a suite job run series
- May include closely coupled applications and decoupled ones that provide an end-to-end repeatable process with differing input parameters
- User and system interaction, to find an answer to a specific science question.

Layer 2 – **Application**
- One or more packages with differing computational and data requirements Interacts across memory hierarchy to archival targets
- The subcomponents of an application {P1..Pn} are meant to model various aspects of the physics

Layer 3 – **Package**
- The processing of kernels within a phase and associated interaction with various levels of memory, cache levels and the overall underlying platform
- The domain of the computer scientist



Layer 1 – Ensemble of applications – Use Case – example template



We described a layer above the application layer (2) that posed use cases that used the application in potential different ways. This also allowed the entry of environment based entities that impact a given workflow and also allow impact of scale and processing decisions. At this level we can describe time, volume and speed requirements.

Our Goal

Measurement infrastructure in support of Holistic HPC Workflow Performance Analysis and Validation

What is Holistic HPC Workflow Performance?

- HPC Workflow Performance encompasses the monitoring and analysis of performance problems that span across traditionally separated aspects of an HPC effort
- Ex: Power as a first-class performance issue
  - Requires integration of room data (location of racks and nodes, rack- node- or component-level power measurements) for mapping of jobs to power consumption)
- Ex: Application-level diagnosis of interference
  - Requires selected performance data from network, file system, power management system, resource manager, etc.
- Ex: Future Planning & Design
  - Describe a “typical” workflow

Holistic HPC Workflow Performance

- **New capabilities**
  - Distinguish “interference” from application-based root cause of performance issue
  - Store appropriate application and system metrics to help evaluate high end platforms and guide future design
  - Procurement
  - Research spanning applications and {power, hardware, system software}

Holistic HPC Workflow Performance

- **WHO cares? (everyone!)**
  - Humans: Developers, Users(Domain Scientists), Sys Admins, Procurement Team, Researchers
  - SW: Scheduler, Power Mgmt System, Security Mgmt System, OS, Runtime System, Checkpoint, Application
- **WHEN do we care? (all stages and timescales!)**
  - Runtime – adapt core placement, tune application, detect security/resilience events
  - Post-mortem – code/job submission request/platform changes needed?

Holistic HPC Workflow Performance: WHAT do we care about?

- ***User***
  - Is there something unusual/ unexpected happening as I run this code?
  - If so, who should I tell (developer or sys admin)?
  - How much resources do I need to run this code?
  - How should I move over the input data set?
  - Where can I analyze the output data?
- ***Developer***
  - Will my code perform well on this [new] platform?
  - Will it meet performance constraints?
    - Exec time, power, memory utilization, number of nodes, data throughput
  - Does this new input data set affect performance?
  - [Where] is there a performance bottleneck?
  - Should I use the accelerators?
  - How can I build in flexibility?

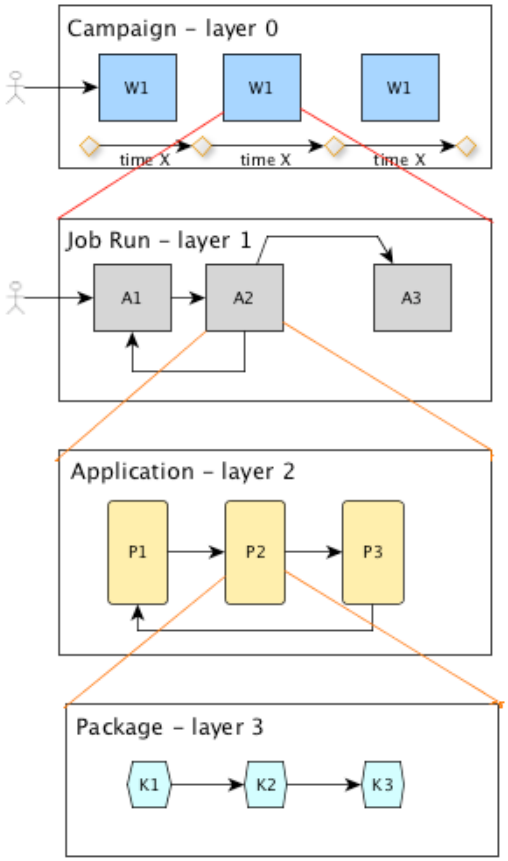
Holistic HPC Workflow Performance: WHAT do we care about?

- ***User***
  - Is there something unusual/ unexpected happening as I run this code?
  - If so, who should I tell (developer or sys admin)?
  - How much resources do I need to run this code?
  - How should I move over the input data set?
  - Where can I analyze the output data?
- ***Sys Admin***
  - Is there something unusual/ unexpected happening as this code runs?
  - If so, what is the root cause?
  - Is this job disrupting other jobs?
  - Do we need to increase capacity?
  - Are jobs completing in reasonable time frame?
  - Did the latest software upgrade cause problems or improvements?

Holistic HPC Workflow Performance

- **WHAT do we care about?: Future Planning and Design**
  - How can I characterize our codes/workload to potential vendors of future platforms?
  - Can benchmarks accurately reflect our workload?
  - Is our workload experiencing bottlenecks on our current platforms?
  - Is the current system meeting performance targets?

What do we care about? HPC Workflows



Collection approaches	For jobs
Pull data from data bases summarized for historic runs	Requirements across time. Scale, checkpoint, data read/written, Data needs over time, overall power, other.
What is collected from each run – job level information. App and system – integrated and tracked. Feeds up.	Requirements for job run. Data movement, checkpoint and local needs, data analysis process, data management. Multiple job tracking, resource integration into system.
During run of app, mainly from within app- data, phases – integrated with system data for environmental perspective. Feeds up.	Memory use, BB utilization, differences between packages in app, time step transition, analysis/preparation of data for analysis, IO
During run of app, mainly from within app – more intrusive collection. Performance, algorithm, architecture, compiler impact etc. Feeds up.	Detailed measurements traditionally done through instrumentation and traditional tools such as Tau, HPC Toolkit, Open| SpeedShop, Cray Apprentice, etc. Focus on - MPI, threads, vectorization, power, etc.

HPC Workflow Performance: Key *Measurement* Challenges

- Total number of measurement points is HUGE
  - Can't buffer all Perf Data in memory
  - Perf Data movement/storage has bandwidth limits, perturbs application
- How to measure overheads of schedulers, runtime system, I/O system, and measurements themselves?
- How to reduce redundant measurements, storage of redundant performance data?
- How to anticipate/workaround/eliminate interference between different measurement tools?

Measurement Challenges Example: Monitoring on Trinity

- Syslog, SEDSI, LDMS
- > 15 TB raw data per day *

- Drains: monitoring network, rabbitMQ, Zenoss, SPLUNK, OVIS

→ must do some intelligent online filtering

** Sanchez et al, Design and Implementation of a Scalable HPC Monitoring System, HPCMaspa 2016.*

HPC Workflow Performance Measurement Application / System Tool Integration

- Goals
  - Can we "mix" app-level data with "typically" collected system data ?
  - What new capabilities will we gain?
  - How could we make it easier?
- Challenges
  - What's in a Name? JOBID, jobid, JobId, ID, …
  - Linking measurements from different layers

  - How to link runtime, I/O utilization, power, cooling – for one job?

HPC Workflow Performance: Key *Diagnosis* Challenges

- Linking measurements from app, system, hw, room
  - How to link I/O utilization, power, cooling – for one job?
- Defining “unusual” or “unexpected” behavior
  - Aggregating data for scalability is wrong
  - Different input data, sw stack “upgrades” change performance
- TimeScales - function time vs power consumption
- Heuristics needed – can’t measure everything

Moving Forward: Research Initiatives

Application/System Monitoring Interface

- Protection Boundaries – accessing data from different system tools requires specialized interfaces
- Ex: accessing Lustre file system monitoring data
  - Tool: all jobs together
  - Need: access to data ONLY FOR MY JOB

Moving Forward: Research Initiatives

New Metric: Workflow Critical Path

- App-level profilers focus performance tuning to specific function or loop or line of code
- What part of the *Entire Workflow* to focus on?
- Portland State Drought Prediction Project

Embedded Performance Probes

- Check runtime performance against Expected Performance
- Specific pre-defined measurement locations in code
- Application – specific: increases accuracy of "expected values"
- Initial work: postmortem Goal: runtime

Summary

- Holistic HPC Workflows characterize One science effort across a period of time/campaign, for 1 specific goal, and may include multiple platforms or labs
- HPC Workflow Performance encompasses the monitoring and analysis of important performance problems that span across traditionally separated aspects of an HPC effort
- Success requires new monitoring infrastructure, runtime interfaces, and a tool for storage/analysis of data from different layers and tools that allows both pre-defined and ad hoc queries

Thank You

- "Where in the World is Karen?" now back in Portland
- Contact: Karen L. Karavanic karavan@pdx.edu

- Acknowledgments:

Unless otherwise indicated, this work was conducted at the Ultrascale Systems Research Center (USRC) supported by Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 with the U.S. Department of Energy. The U.S. Government has rights to use, reproduce, and distribute this information. This work supported in part by Portland State University and by the New Mexico Consortium.

It took a whole village to do the work mentioned and described here.