Continuous Metric Collection / Assessment
- An Ecosystem Approach -

Scalable Tools Workshop 2017

August 6 - August 10, 2017

David Montoya

August 7, 2017

Los Alamos
NATIONAL LABORATORY
EST.1943

NNSA
National Nuclear Security Administration

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

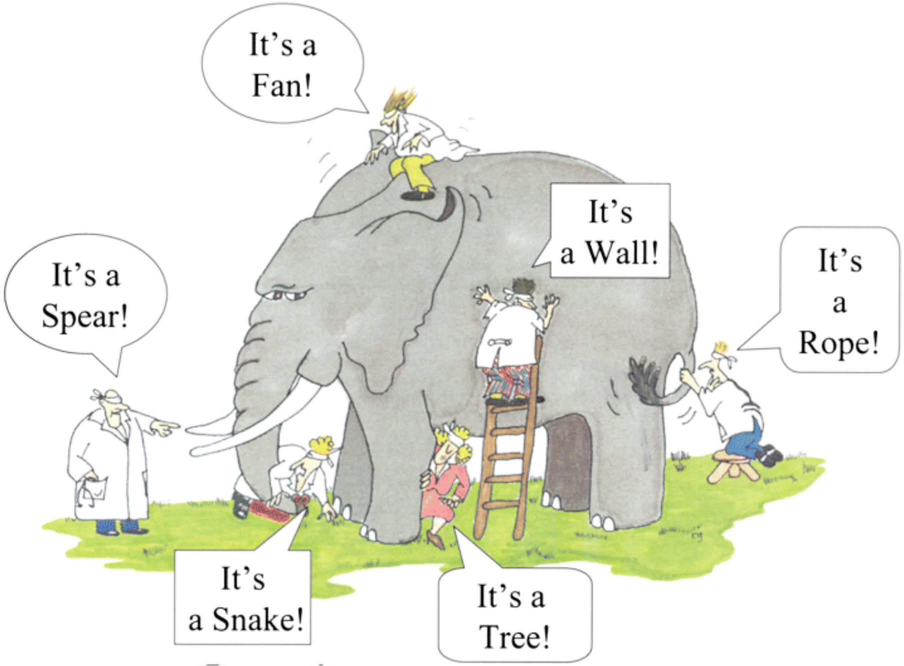
LA-UR-17-26791

The convergence of Performance Analysis tools with Monitoring and Analytic tools

Evolution

- Traditional performance analysis has **targeted application internals** and its immediate resources.
- **Complexities of increasing scale and unique architectures** are driving a more holistic perspective toward performance.
- Design of the computational environment, its constraints, and **ability to balance resource use** is a key aspect in performance. There are varying perspectives of performance depending on differing stakeholders.
- Let's review many of the drivers, approaches, and challenges of an integrated performance environment.

Context for Discussion



- **2014 – 2015, LANL effort to characterize IO patterns for 2020 Crossroads RFP**
  – Data to drive storage system design
  – Workflow Taxonomy developed
  – Lack of data being collected
- **2015 – 2016, ASC Fast Forward/ Design Forward workflow efforts**
  – Vendors desiring workflow data
  – Broader characterization of performance
  – Push toward monitoring infrastructure
- **2016 – 2017, The drive for Holistic collection / Analytics**
  – Blend of taxonomy, Apps / Systems data, collection mechanisms, analytics
  – A broader sense of "Shared Fate"

2014 – 2015, Workflow Taxonomy

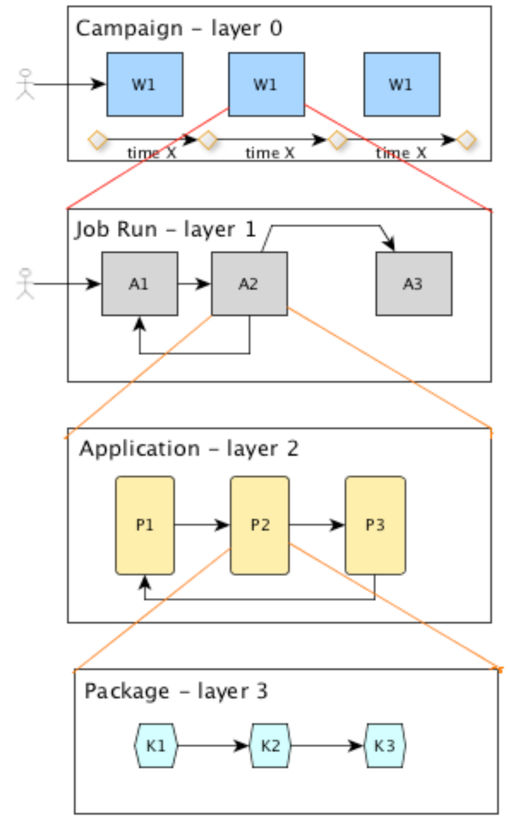
The work up to this point is what has come together as the workflow characterization project has evolved. Developed a taxonomy, talked to teams, used for RFP whitepaper

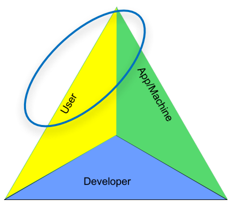
Layer 0 – **Campaign layer**. Process through time of repeated Process layer jobs with changes to approach, physics and data needs as a campaign or project is completed. Working through phases.

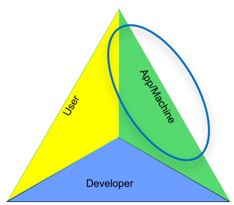
Layer 1 – **Job Run layer**. Application to application that constitute a suite job run series, which may include closely coupled applications and decoupled ones that provide an end-to-end repeatable process with differing input parameters. This is where there is user and system interaction, constructed to find an answer to a specific science question. Layer 0 and 1 are from the perspective of a end user.

Layer 2 – **Application layer**. Within an application that may include 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 1 and 2 are the part of the workflow that incorporates the viewpoint of the scientist.

Layer 3 – **Package layer**. This describes the processing of kernels within a phase and associated interaction with various levels of memory, cache levels and the overall underlying platform. This layer is the domain of the computer scientist and is where the software and hardware first interact.

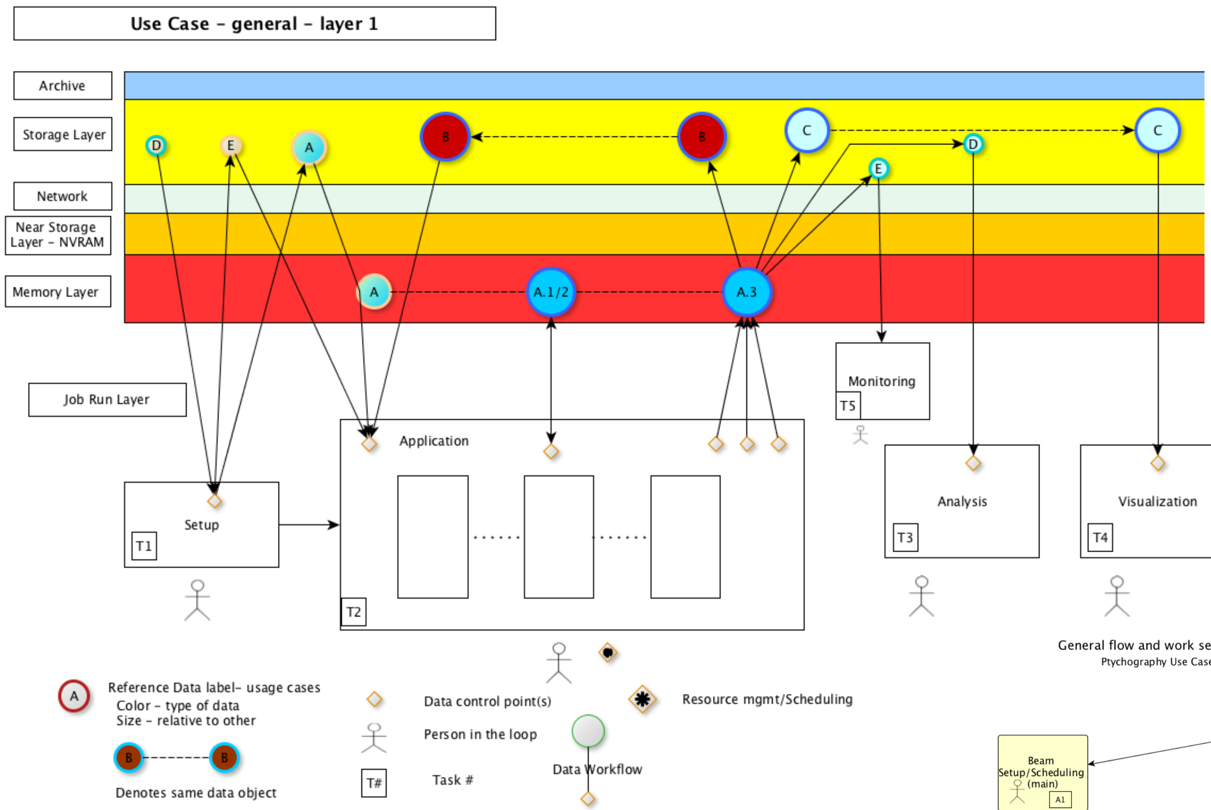


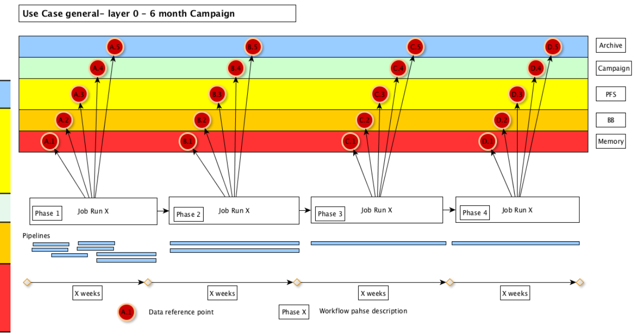


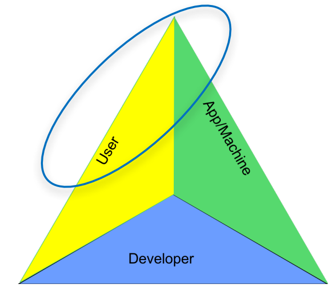


A language to describe different WFs

2014 – 2015, Workflow Taxonomy

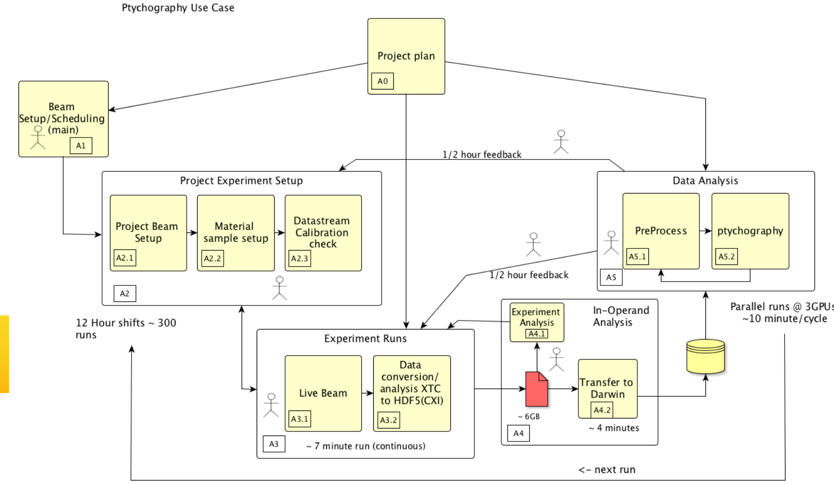






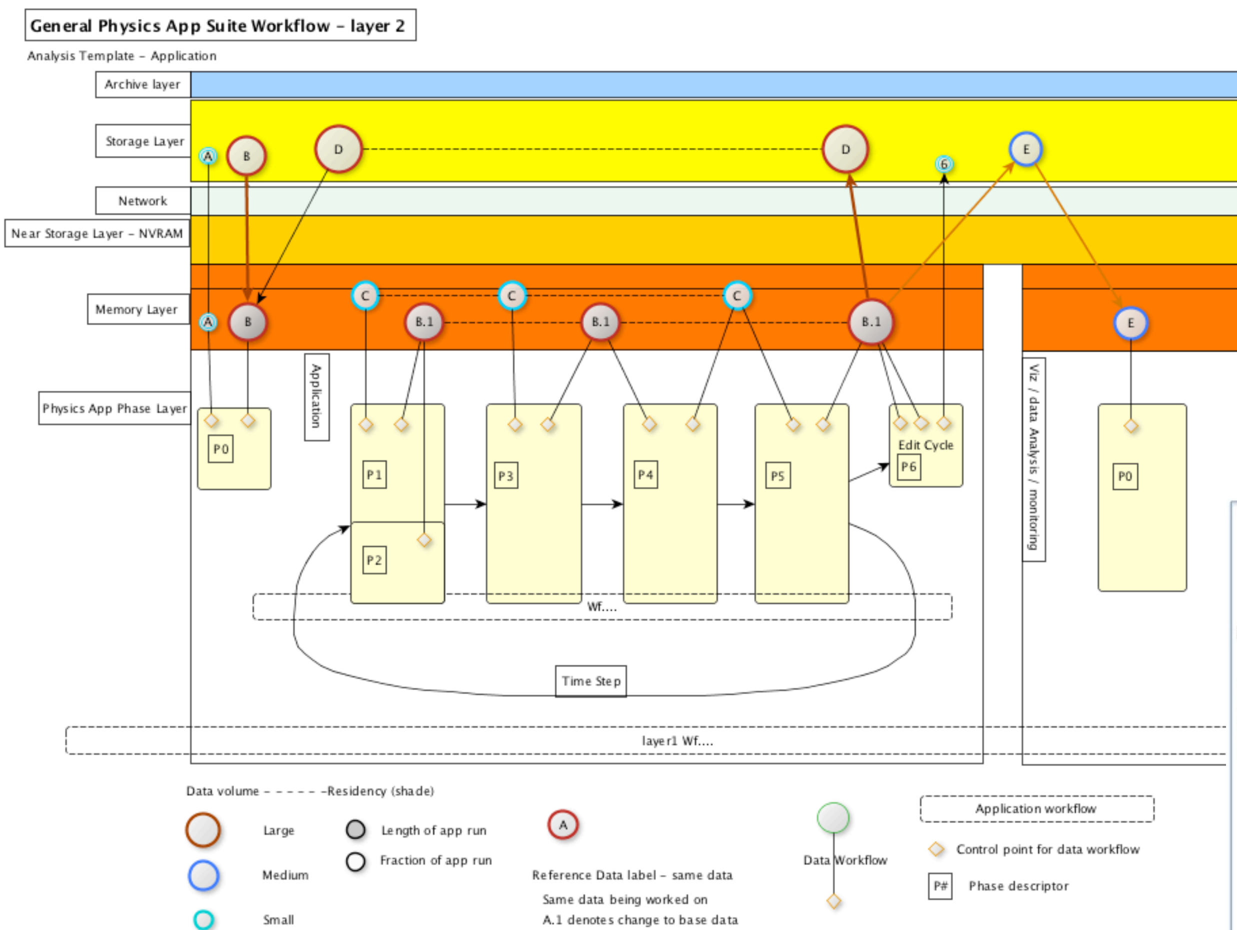
General flow and work segments

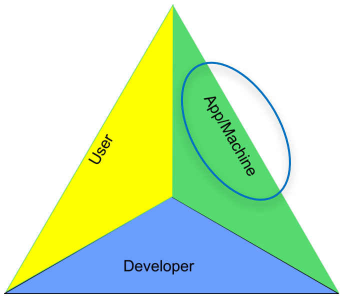
Ptychography Use Case

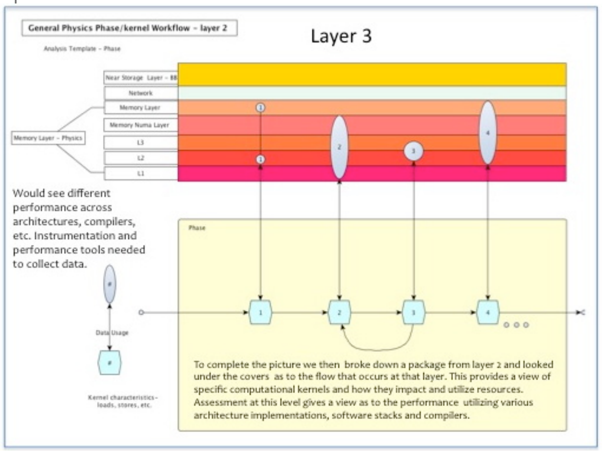


Layers 0 and 1 – Focuses on the user

2014 – 2015, Workflow Taxonomy







Layers 2 and 3 - Understanding what's under the covers

2014 – 2015, IO Analysis for RFP

APEX WF Whitepaper

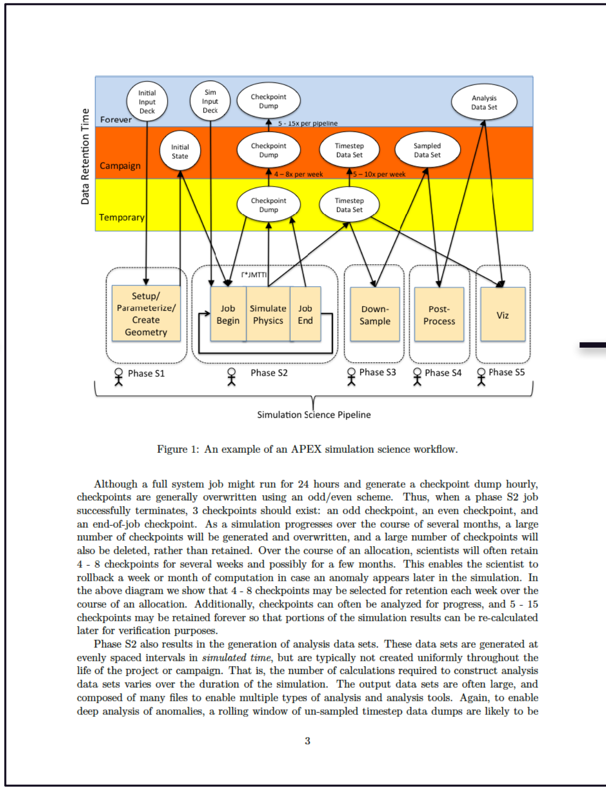


Figure 1: An example of an APEX simulation science workflow.

Although a full system job might run for 24 hours and generate a checkpoint dump hourly, checkpoints are generally overwritten using an odd/even scheme. Thus, when a phase S2 job successfully terminates, 3 checkpoints should exist: an odd checkpoint, an even checkpoint, and an end-of-job checkpoint. As a simulation progresses over the course of several months, a large number of checkpoints will be generated and overwritten, and a large number of checkpoints will also be deleted, rather than retained. Over the course of an allocation, scientists will often retain 4 - 8 checkpoints for several weeks and possibly for a few months. This enables the scientist to rollback a week or month of computation in case an anomaly appears later in the simulation. In the above diagram we show that 4 - 8 checkpoints may be selected for retention each week over the course of an allocation. Additionally, checkpoints can often be analyzed for progress, and 5 - 15 checkpoints may be retained forever so that portions of the simulation results can be re-calculated later for verification purposes.

Phase S2 also results in the generation of analysis data sets. These data sets are generated at evenly spaced intervals in *simulated time*, but are typically not created uniformly throughout the life of the project or campaign. That is, the number of calculations required to construct analysis data sets varies over the duration of the simulation. The output data sets are often large, and composed of many files to enable multiple types of analysis and analysis tools. Again, to enable deep analysis of anomalies, a rolling window of un-sampled timestep data dumps are likely to be

3

	Tri---Labs workload							
	LANL				SNL		LLNL	
Workflow	EAP	LAP	Silverton	VPIC	Dakota A	Dakota S	pF3D	R15
Workflow type	Sim	Sim	Sim	Sim	Sim/UQ	UQ	Sim	UQ
Workload percentage	60	5	15	10	10	10	10	20
Representative workload percentage	20	2	5	3	3	3	3	6
Wall time (hours)	262.4	64.0	128.0	157.2	100.0	100.0	2304.0	76.8
Hero Run Cielo Cores	65536	32768	131072	70000	131072	65536		
Routine Number of Cielo cores	16384	4096	32768	30000	8192	4096	2048	4096
Number of workflow pipelines per allocation	30	10	6	4	1[illegible] x 100	30 x 300	2	100
Anticipated increase in problem size by 2020	10 to 12x	8 to 12x	8 to 16x	8 to 16x	4 to 8x	1.25 to 1.5x		1x
Anticipated increase in workflow pipelines per allocation by 2020	1x	1x	1x	1x	2 to 8x	2 to 4x		10x
Storage APIs	POSIX	POSIX	POSIX	POSIX	HDF5 or NetCDF	HDF5 or NetCDF	POSIX	POSIX
Routine number of analysis datasets	100	100	225	150				
Routine number of analysis files								
Checkpoint style	N to 1	N to 1	N to 1	N to N	N to N	N to N	N to N	N to N
Files accessed/created per pipeline								
Data description (95% of storage volume)								
Data retained per Pipeline (percentage of memory)	268.00	510.00	463.00	360.25	5.87	32.54		
Temporary	30.00	75.00	285.00	222.75	0.02	30.00		
Analysis			5.00	200.00				
Checkpoint	30.00	75.00	210.00	18.75	0.02	30.00		
Input			70.00	5.00				
Out---of---core								
Campaign	170.00	170.00	100.00	115.00	2.00			
Analysis	80.00	70.00	30.00	60.00	2.00			
Checkpoint	90.00	100.00	70.00	50.00				
Input				5.00				
Forever	68.00	265.00	78.00	22.50	3.85	2.54		
Analysis	25.00	250.00	8.00	10.00	0.85	2.04		
Checkpoint	40.00	10.00	70.00	12.50				
Input	3.00	5.00			3.00*	0.50*		

Data Perspective

This provided the basis for discussions with vendors and is opening conversations with users and development teams

Revising as we ask additional questions and further validate

Data metrics and usage behavior

2014 – 2015, What did we learn

- Differences in application WFs and data usage patterns are invaluable to industry architects (vendors), HPC infrastructure environment development and provides insight to application projects

- Data describing WF was hard to validate (mainly interviews and some observations). Little data being collected that could continuously feed development of a behavior model.

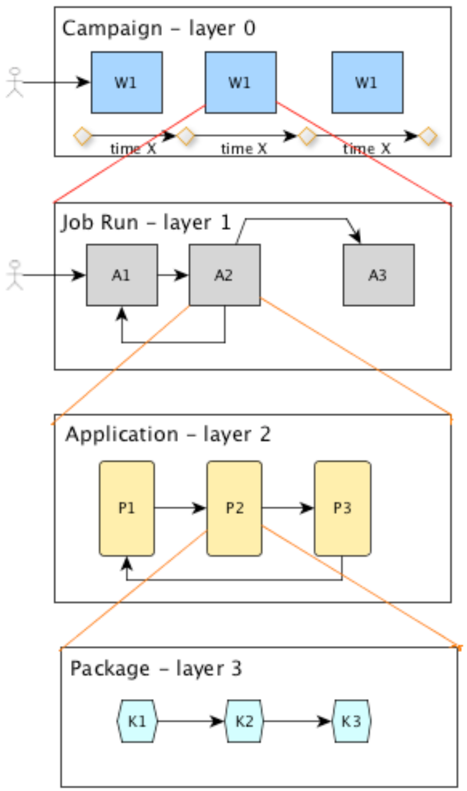
- Determined that we were driving blind and needed to assess data collection approaches to support better characterization. At all layers of the WF taxonomy.

- Still need a way to extrapolate application utilization/performance data on future architecture lanes and have the ability to assess their impact

Need data to build a crystal ball

2014 – 2015, Collection points for data - layers

What are important metrics for each layer?



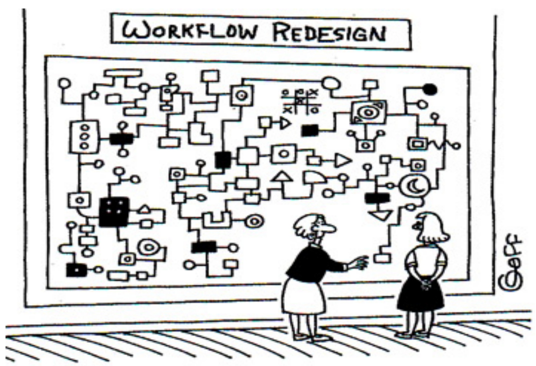
Collection approaches	For jobs
- Pull data from data bases summarized for historic runs	• Requirements across time. Scale, checkpoint, data read/written, Data needs over time, overall power, other.
- What is collected from each run – job level information. App and system – integrated and tracked. Feeds up.	• Requirements for job run. Data movement, checkpoint and local needs, data analysis process, data management. Multiple job tracking, resource integration into system. Tools such as Darshan, and other Perf tools
- 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, traces
- 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.

Data has value beyond its initial use case

2015 – 2016, ASC FF/ DF Workflow efforts

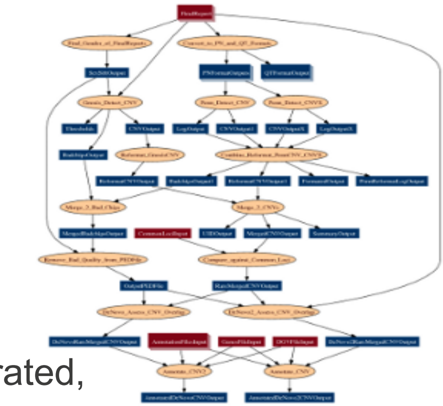
Using workflows to charaterize

- **Environments and Architectures getting more complex**
- **Needing to understand App behavior**
- **Demands on individuals increasing**
- **Ability to document process and reproduce results getting harder**
- **Need to work across teams increasing and broad collaboration becoming the norm**



"And this is where our ED workflow redesign team went insane."

System Drivers



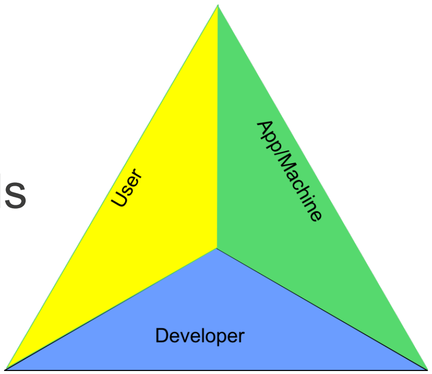
- **Future systems** becoming more integrated, both self and environment aware
  – Asynchronous task and data processing
  – Heterogeneous architectures
  – Deeper memory through to storage
- **Integrated runtimes / emerging programming models**
- **Data analysis and reduction**
- **Power management**

Understanding the need for a workflow methodology

2015 – 2016, FF/DF Workflow efforts

System Architecture efforts strive to understand application behavior/needs in the wild…

- Need an understanding of applications and usage in specific workflows
  - Not easy, applications are just one of the tools they use for specific problem analysis. Application has core behavior, but the study will drive variations.
- Need to communicate to vendors on application and WF needs
  - Initial approaches have been done by vendors. Needs to be a collaboration driven by supporting data.
- Architecture ecosystem and integration of services getting more complex with increased scale
  - Interrelationships have a growing impact. Services become their own sub-systems.



Pain Points!!

2016 – 2017, The drive for Holistic Collection / Analytics

Where are we now?

Continued Evolution – the little fish have grown legs and are stepping into new envrionments…

Observations- (continuation of information K Karavanic presented, 2016)

- We see convergence of traditional performance tools and on-going monitoring.
- An understanding that integration of data from many sources is key to understand behavior and performance issues.
- App and system teams, tool providers – moving out of the stovepipes that they could control and building relationships (they want data).
- An assessment of collection system, analytic engines, and needs of different stakeholders.

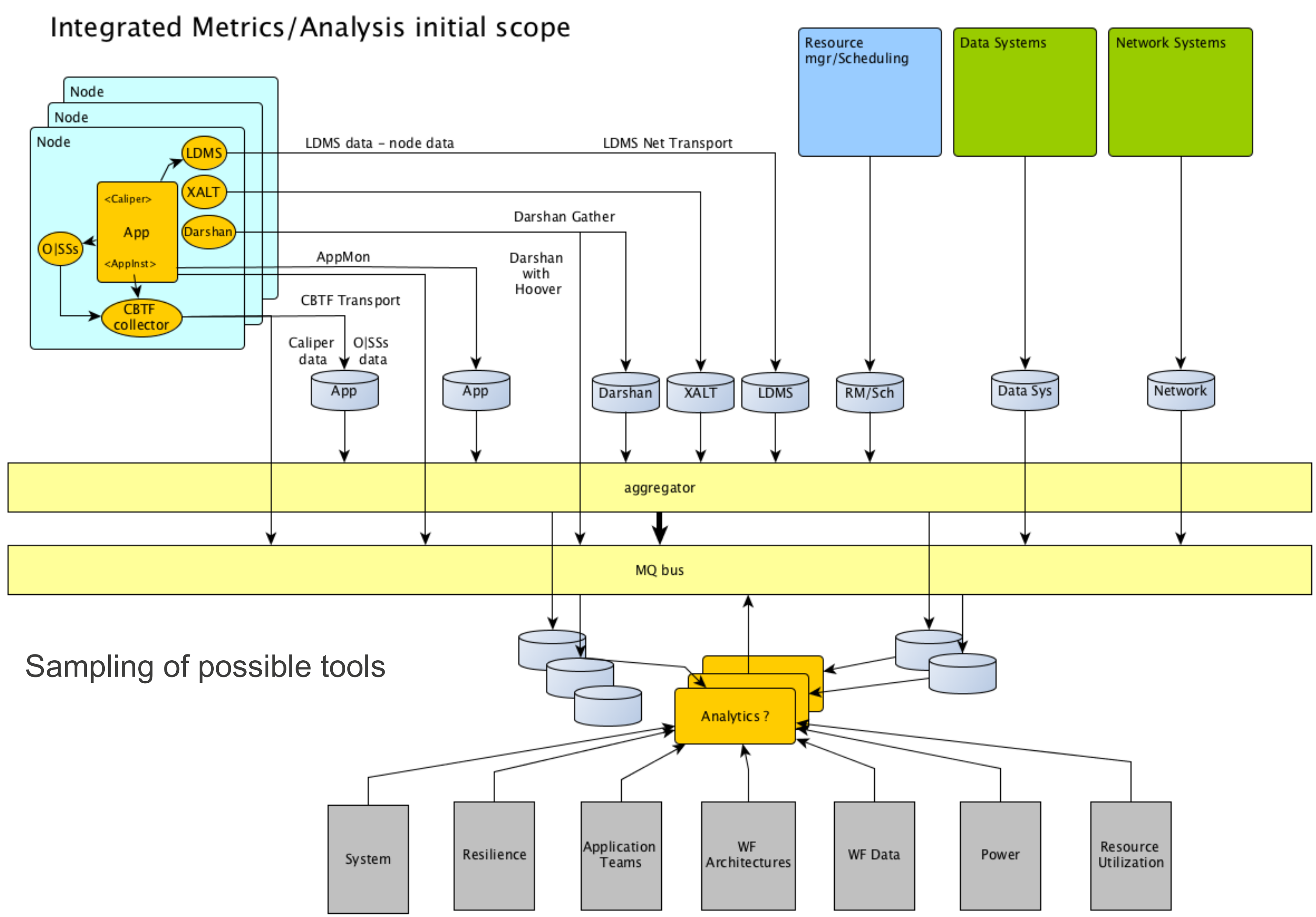
Shared Fate and a bit of Trust

2016 – 2017, LANL Status

Currently in the Pipeline

- Initial focus on Application data collection and a bit of Storage.
- Identified initial collectors to focus on from key tool providers.
- Identified initial tools to work toward integration (many others have features that may be integrated later).
- Have a robust RabbitMQ infrastructure for transport and distribution of some of the collected data with Fluentd as generic data integrator.
- Have initial settled on analytics infrastructure that includes Elasticsearch, OpenTSDB, Grafana, with some prototyping of various machine learning approaches.
- Much room for optimization and growth.

2016 – 2017, LANL (and others) Collection and Analytic Infrastructure development



Conclusion

- **Collection systems will start to be baked into HPC infrastructures and environments – needed for information and for the feedback needed for the system, software environment, and applications**
- **There is a need for much more convergence across tools and infrastructure – positives and negatives**
  – Some tool frameworks will try to do it all.
  – Keeping tool identity allows for the specific capability that the tool framework provided for targeted domain area.
  – There are some natural convergence points regarding transports, pub/sub infrastructures, etc.
  – A more cohesive ecosystem approach to drive vendor adoption.
  – We've had this discussion before ??
  – Much opportunity in collector space, may allow for some shared capability.
  – The analytic infrastructure will be the new wild west for a few years.

- **This should be fun!! Let's continue to leverage the community..**

Thanks for Listening..

Discussion and Questions

The End..