Scalable Composition and Analysis Techniques for Massive Scientific Workflows.

18th IEEE International Conference on eScience

Dong H. Ahn*, Xiaohua Zhang, Jeffrey Mast, Stephen Herbein*, Francesco Di Natale*, Dan Kirshner, Sam Ade Jacobs, Ian Karlin*, Daniel J. Milroy, Bronis de Supinski, Brian Van Essen, Jonathan Allen, and Felice C. Lightstone

Oct 14, 2022

* These authors are now with NVIDIA Corporation

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

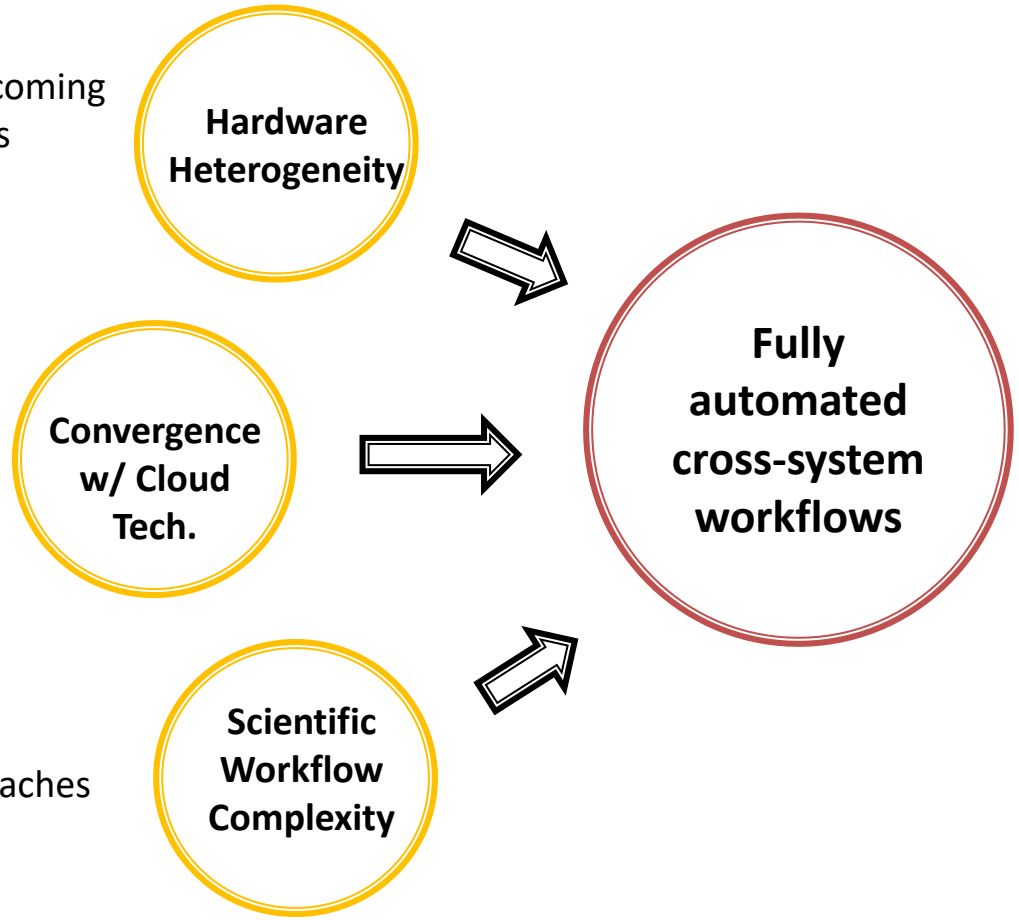
Lawrence Livermore National Laboratory

Current trends are creating an significant need for fully automated scientific workflows that must run across multiple systems.

- GPUs are mainstream, DPUs and AI accelerators are coming
- Leading AI accelerators already fielded at HPC centers
- Diversity will only increase

- Driven by economy and need for higher diversity
- Must leverage cloud for our long-term viability

- More apps are combined
- Data-science facilitate workflow approaches



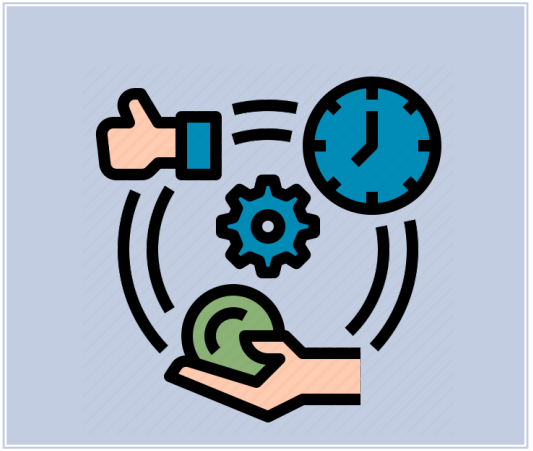
Our work identifies three key challenges and solutions needed to compose, analyze and optimize a cross-system workflow.

Conceptualization

- *Knowing the key performance space of a cross-system workflow that is composed of many application components.*

Performance Analysis

- *Gathering performance data from large numbers of components without losing modularity and uniformity*

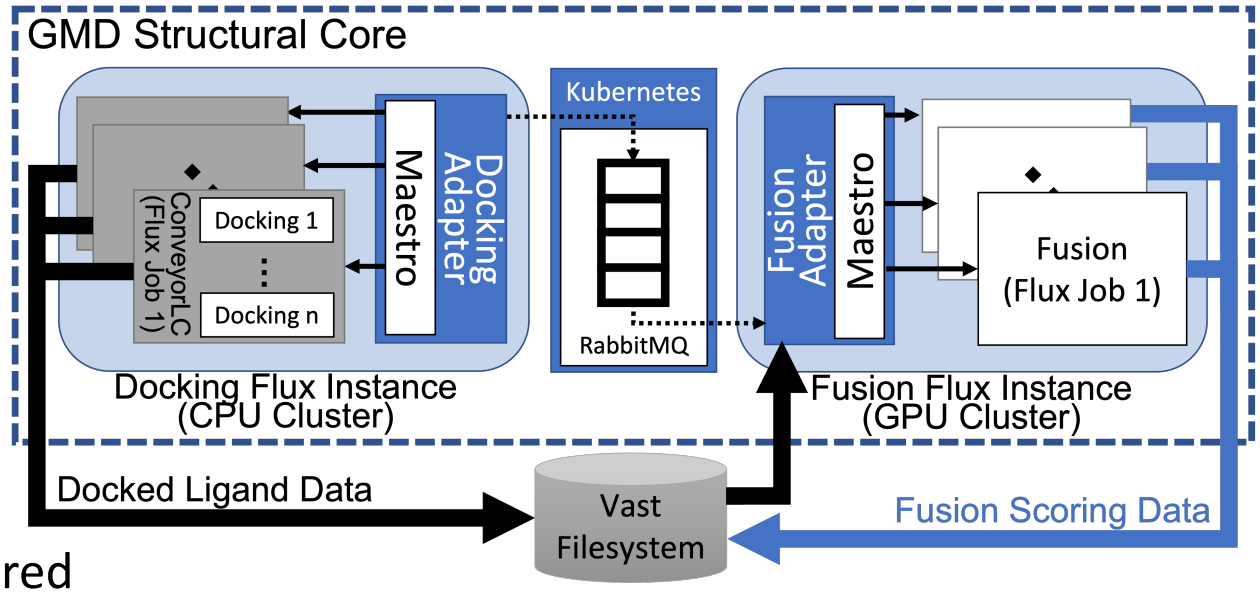


Performance Optimization

- *Easily re-configuring the workflow to tune the turn-around time performance of an end-to-end workflow*

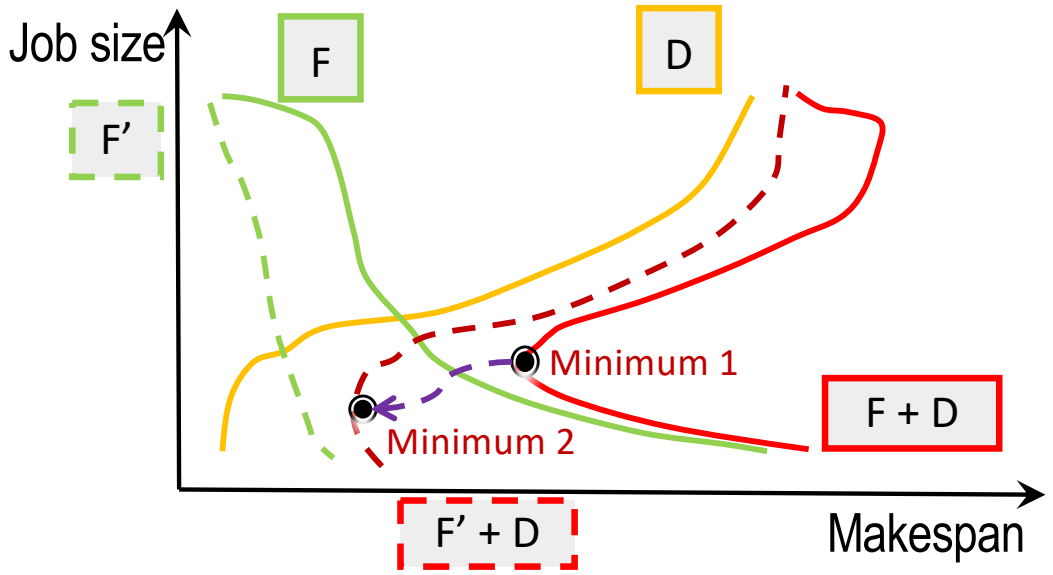
Our work targets a small molecule drug design and development workflow with concrete cross-system computational requirements.

- Need for screening billions of molecules
  - Combine traditional HPC simulation with ML
- Map to large supercomputers and Kubernetes
  - Of 3 completely different computing hardware
- Combine many application components
  - General-purpose workflow management components
  - Domain-specific application components
- Software engineering is a contribution but …
- Mostly talk about performance challenges encountered
  - Measuring, analyzing and optimizing
  - The performance of the complete workflow



AHA Moles workflow architecture

Challenge 1: Conceptualization of the performance tradeoff space of large composite workflows is non-existent.

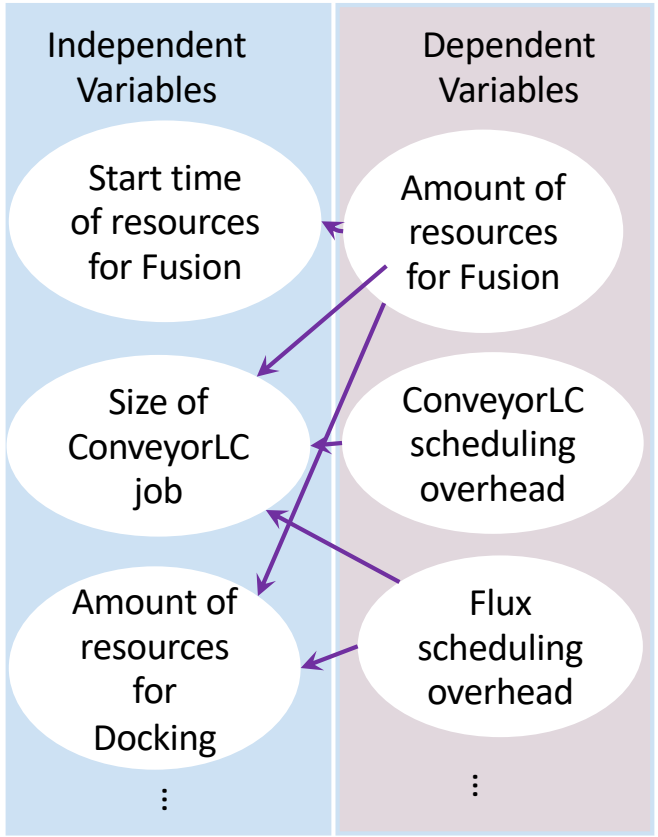


F: Default Flux scheduler
F': Optimized Flux scheduler
D: Docking task scheduler within ConveyorLC simulation code

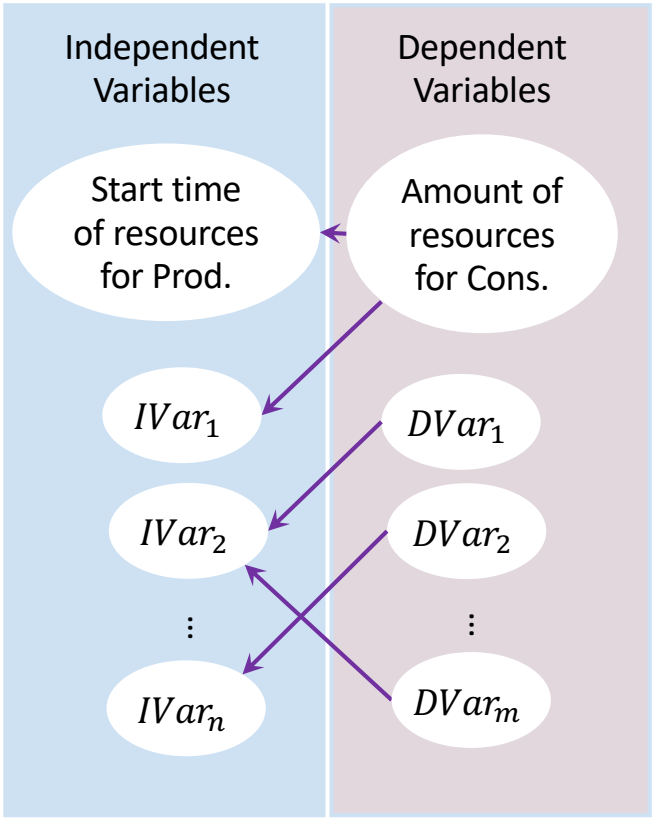
- Complex interplays of many different application
  - Generate, schedule and execute tasks in concert
  - A task must be adapted between large numbers of components
  - Single component optimization often does not mean …
- An example found in our drug screening workflow
  - F ◦ D: A large job size is better for Flux but can expose scalability issues within ConveyorLC Docking
  - F' ◦ D: Flux optimization non-intuitively shifts optimality
- Optimality can be reasoned about
  - Only when we consider the composite performance functions
  - Across the interdependent application components

Our proposed performance variables-based approach allows for easy reasoning of the key performance trade-off space.

- Independent performance variables
  - Amounts of resources allocated to docking
  - Size of ConveyorLC docking jobs
  - Start time of resources allocated to Fusion
- Dependent performance variables
  - Performance of Flux job scheduling
  - Performance of ConveyorLC scheduling
- A point optimization must be reasoned
  - With such performance dependencies



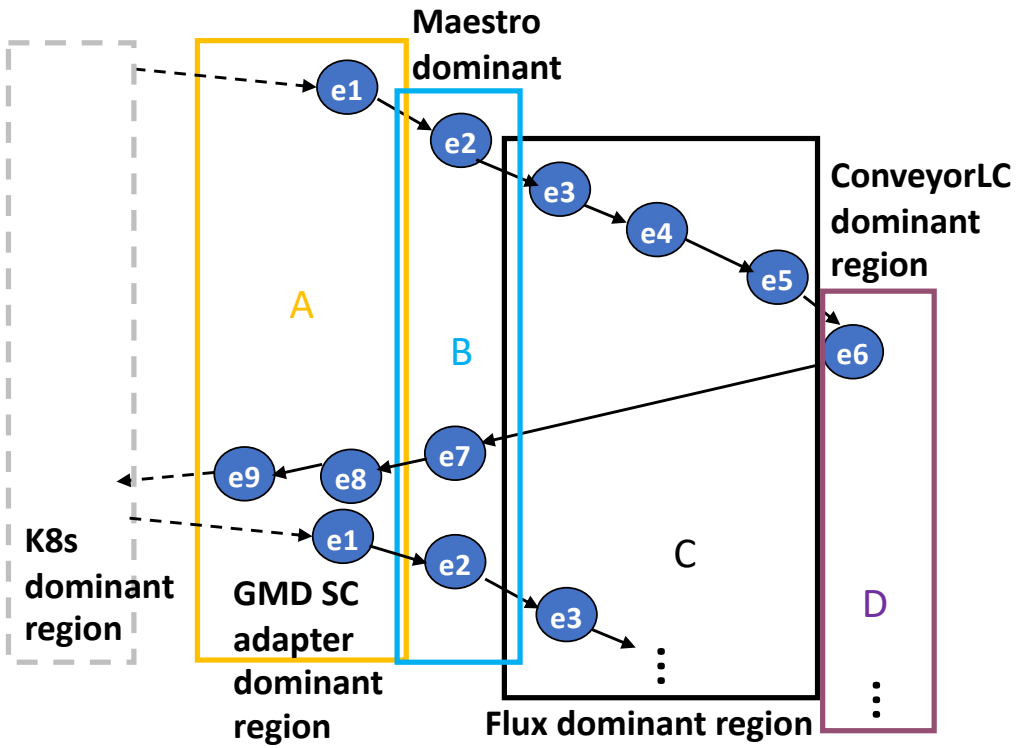
Performance variables and dependencies of drug-screening composite workflow



Generalization

Challenge 2: Analyzing the overall turn-around time performance of a workflow in a manageable fashion.

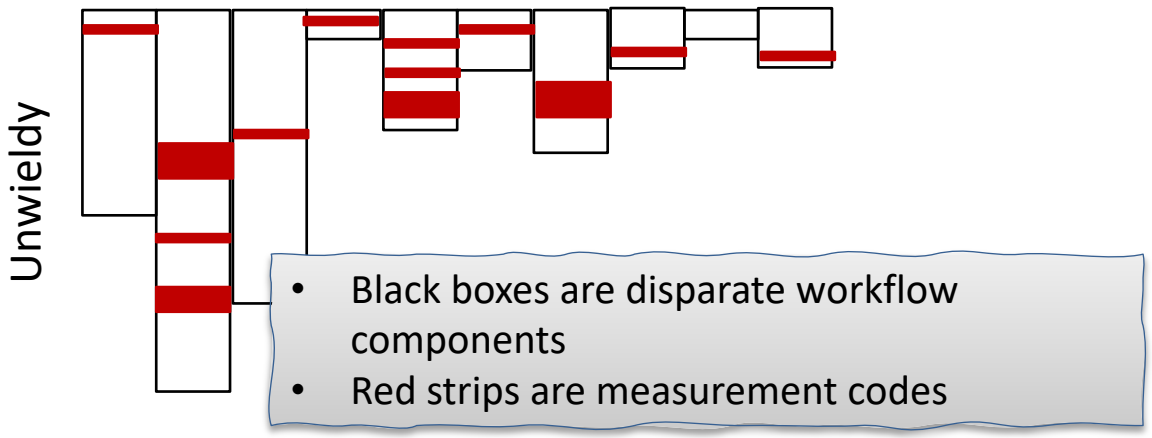
- Many HPC profilers, tracers and techniques exist, but …
  - Target a single application performance analysis
- Proposed principles for composite workflow analysis
  - Each component scheduler exposes its performance characteristics
  - An overall workflow-level performance model guides the composition
  - Full observability on the performance of the complete workflow
- Critical path analysis (CPA) is used as our modeling basis
  - To embody the first two principles
  - Determine the cause of a workflow's makespan
  - By finding the event path in the task execution history of the workflow that has the longest duration.
- Developed PerfFlowAspect for the 3rd principle

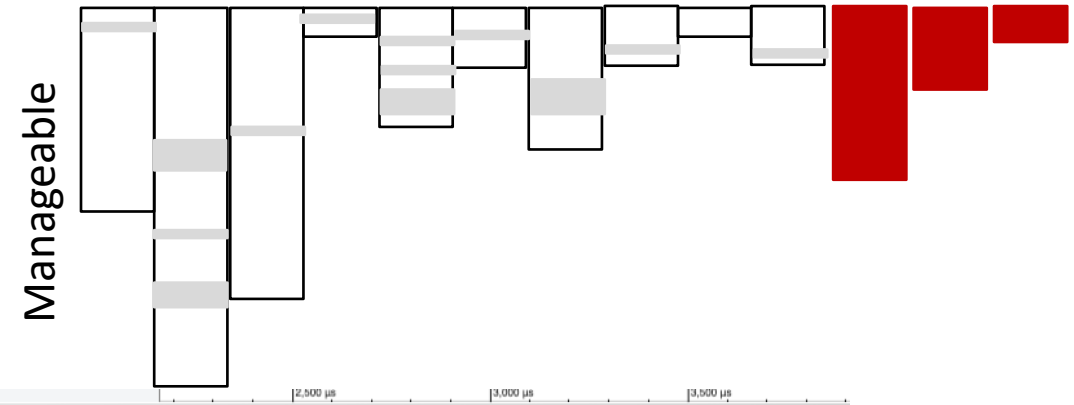


Critical path of Docking tasks for AHA MoleS

We use Aspect Oriented Programming (AOP) to observe the performance of the workflow with modularity and uniformity.

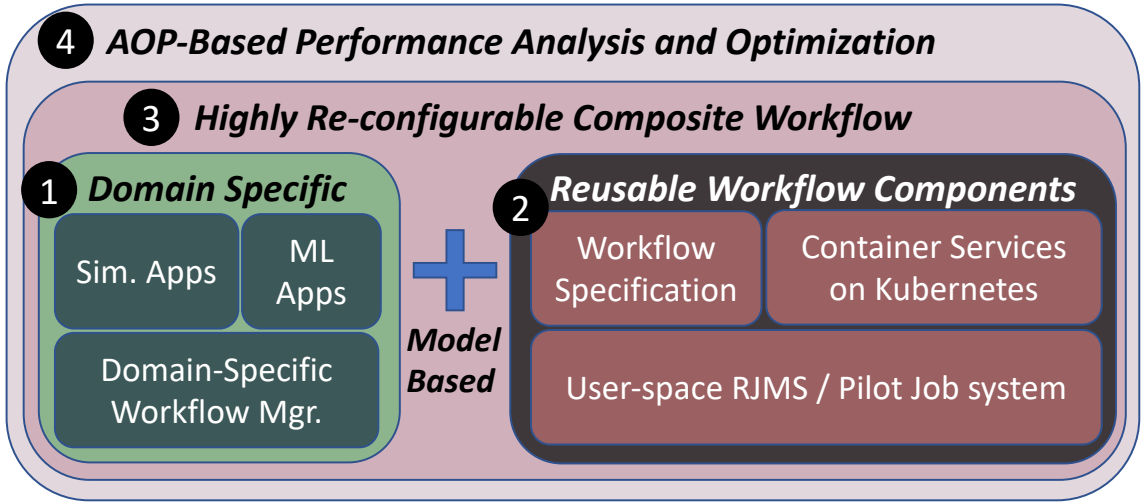
- AOP paradigm allows for
  - Minimum modifications to the disparate application components
  - Casting the cross-cutting performance-analysis concerns
- PerfFlowAspect implements this concept
  - Decorator-based annotation for Python
    - **@critical_path (pointcut=<type>)**
  - __attribute__ based compiler instrumentation (LLVM-based)
    - **__attribute__ ((annotate("@critical_path (pointcut=<type>)")))**
  - Provide one type of "advice"
    - Advice emits tracing data on every annotated function invocation
    - In Chrome Tracing Format (CTF)
- Component application owners are responsible for
  - Annotating their components independently
  - On those functions on the *plausibly* critical path
- Resulting composite workflow produces
  - Performance traces and profiles
  - Without sacrificing the requisite modularity and uniformity





Challenge 3: Easily re-configuring the workflow for iterative exploration of the end-to-end workflow turn-around time performance.

- Addressed directly at the workflow software architecture level
- Performance modeling and variables guide blending
  - Via well-defined API
- Form a highly re-configurable base platform
  - W/ respect to critical performance variable according to the CPA m
  - E.g., Size for ConveyorLC docking jobs
- Iterate over the key performance configuration space
  - Captured in our performance variables
  - Analyzing the AOP traces/profiles for each iteration
  - Using the performance traces/profiles from PerfFlowAspect



Proposed target composite science workflow architecture

Cross-system workflow evaluation environment:

- Build AHA Moles workflow software as a multidisciplinary team until …
  - We can perform controlled benchmark runs at large scale
  - Simultaneously using two supercomputers and on-premises RedHat OpenShift Kubernetes cluster at LLNL
- Ruby: a CPU supercomputer with a total of 1,512 compute nodes.
  - Each node contains two Intel Xeon CLX-8276L CPUs with a total of 56 cores.
- Lassen: a CPU-GPU heterogeneous supercomputer with a total of 795 nodes.
  - Each of the nodes has two IBM Power9 CPUs with a total of 44 cores
  - Four Nvidia Tesla V100 (Volta) GPUs.
- The RedHat OpenShift Kubernetes cluster consisting of
  - Three Kubernetes servers and five worker nodes
  - Together with a NetApp AFF A400 storage system providing object and persistent container storage

Performance of eight different configurations of AHA Moles.

Flux Version	Docking Nodes/job	Fusion Total nodes	Makespan (s)	Per Mol. (ms)
Default	145	220	15208	7.67
Default	5	220	7841	3.95
Default	145	140	15257	7.69
Default	5	140	9717	4.90
Opt.	145	220	19049	9.60
Opt.	5	220	8041	4.05
Opt.	145	140	19236	9.70
Opt.	5	140	9666	4.87

Optimized Flux led to generally worse performance!

2.46x speed-up

- Flux Default (flux-core v0.26, fluxion v0.15)
- Flux Optimized (flux-core v0.27, fluxion v0.16)

X

- Fusion Over-provisioned (220 Lassen nodes)
- Fusion Under-provisioned (140 Lassen nodes)

X

- Large Docking Job (145 nodes)
- Small Docking Job (5 nodes)

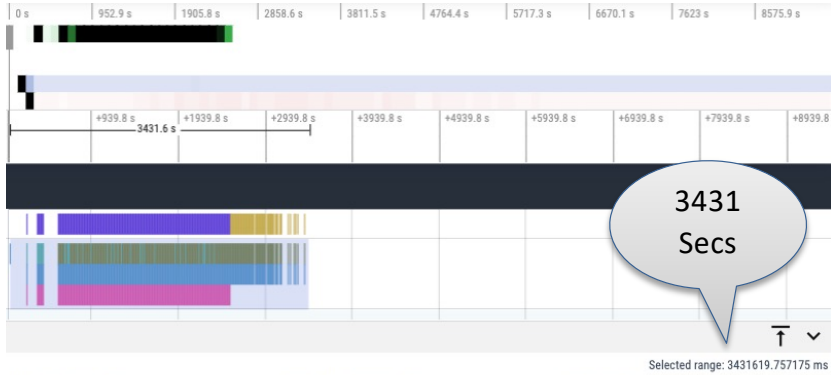
Results synthesis and analysis

- Choice of independent performance variables is critical
- The fact that the default Flux version generally outperformed the optimized version is unexpected
  - Signal the non-linear nature of the composite performance functions
- A suboptimal choice of another independent performance variable can explain a significant under-utilization of Lassen resources
  - Start time of resources for Fusion
  - Lead to a poor speedup with Fusion resource scaling.
    - In the best case, a 1.57x Lassen resource increase provides only a 1.24x speedup.
    - In the worse case, the same 1.57x Lassen resource increase is unable to offer any speedup whatsoever.

Effectiveness of performance analysis and optimization:

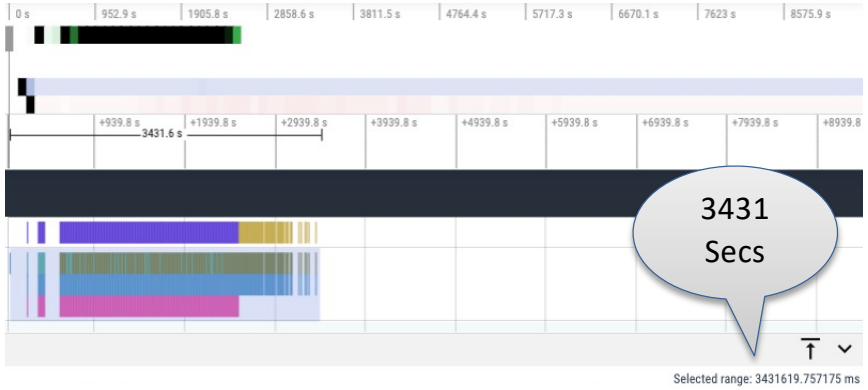
- Can our performance analysis techniques help researchers better understand the impacts of a key performance variable?

(a) Large ConveyorLC Docking job, large Fusion and default Flux

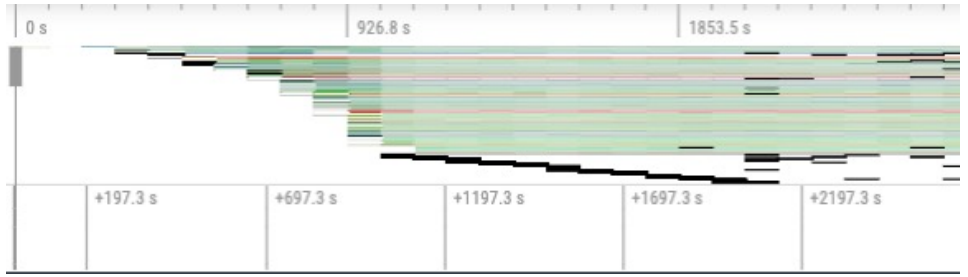


(a) Small ConveyorLC Docking job, large Fusion and default Flux

- Can our analysis inform researchers of a previously unknown area for performance optimization?



(a) Small ConveyorLC Docking job, large Fusion and default Flux



(b) Fusion traces for the best configuration

Our solutions can enhance the capacity of a multi-disciplinary team to create, analyze and optimize a high-performance composite workflow.

- A growing consensus that three major characteristics define the next-generation of HPC centers
  - Extreme hardware heterogeneity at all levels
  - Closer convergence of HPC with cloud computing software
  - Complex scientific workflows that must automatically run on next-generation resources across the entire center
- Our work provides a window into the challenges that these cross-system workflows will face
  - Based on multi-disciplinary effort to run a drug screen workflow across 3 completely different types of center resources
- Creating a cross-system workflow using portable software components is an important step, but
  - Gaining a deep understanding of the performance space of the composite workflow is equally challenging
  - Introduce the concept of performance variables to capture the interplay of different software components
  - Instrumenting and analyzing workflows are an unmet challenge which we overcome with PerfFlowAspect
  - Difficulties of composite science workflow performance tuning is solved by our direct iteration-based exploration support
- Our experiments show that our solutions significantly address the corresponding challenges
  - A better choice for an independent performance variable alone can provide AHA MoleS up to a 2.45x improvement
  - Identification of dependent variables significantly helps narrow down the search space of end-to-end workflow tuning.

Title for full-frame image

Subtitles can be used on longer titles, 24pt “Regular” (no bold)