Parallel Performance Existentialism

Allen D. Malony

University of Oregon

Université de Versailles-Saint-Quentin-en-Yvelines

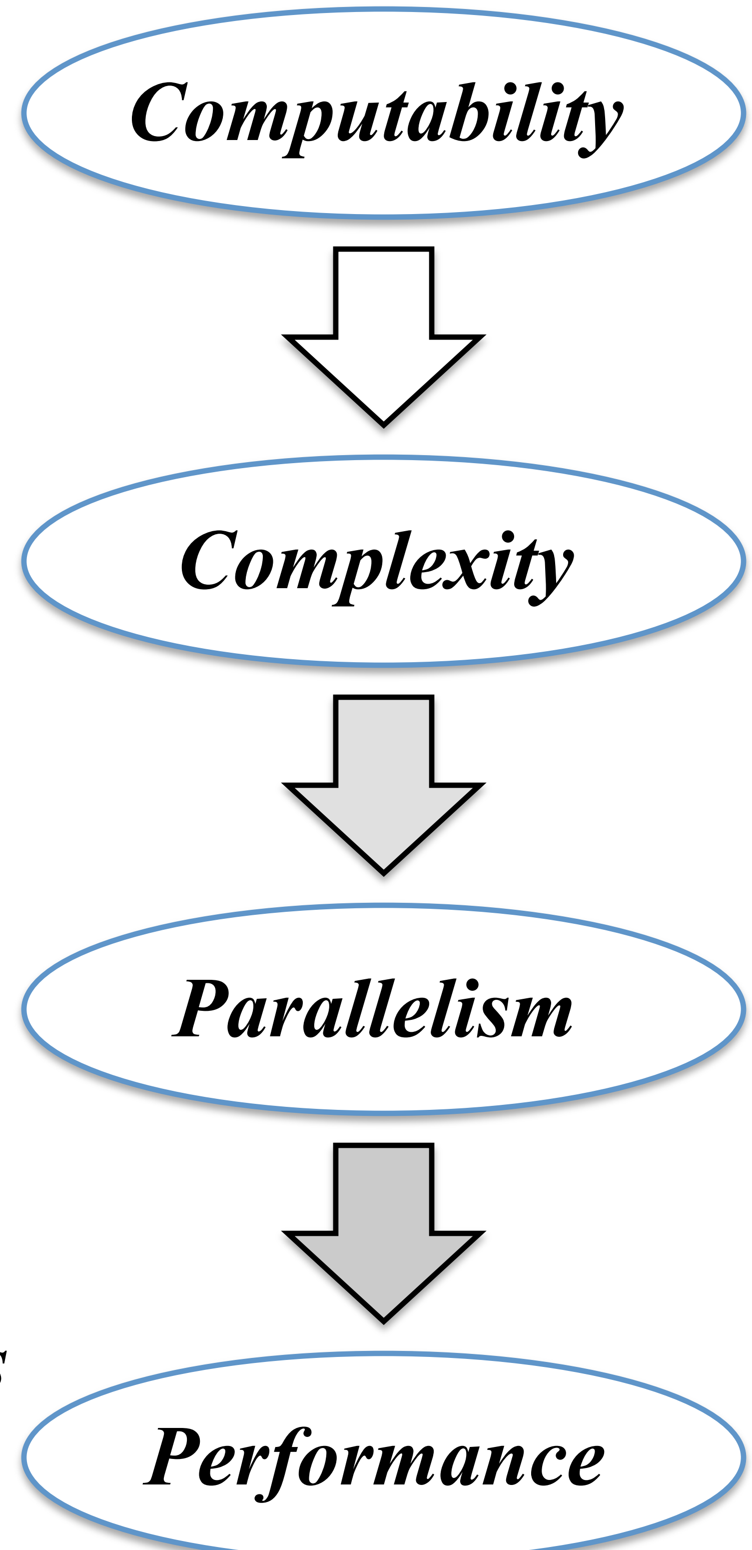
VI-HPS 10th Anniversay Workshop

June 23, 2017

FULBRIGHT

COMMISSION FRANCO-AMÉRICAINE

Computability to Performance

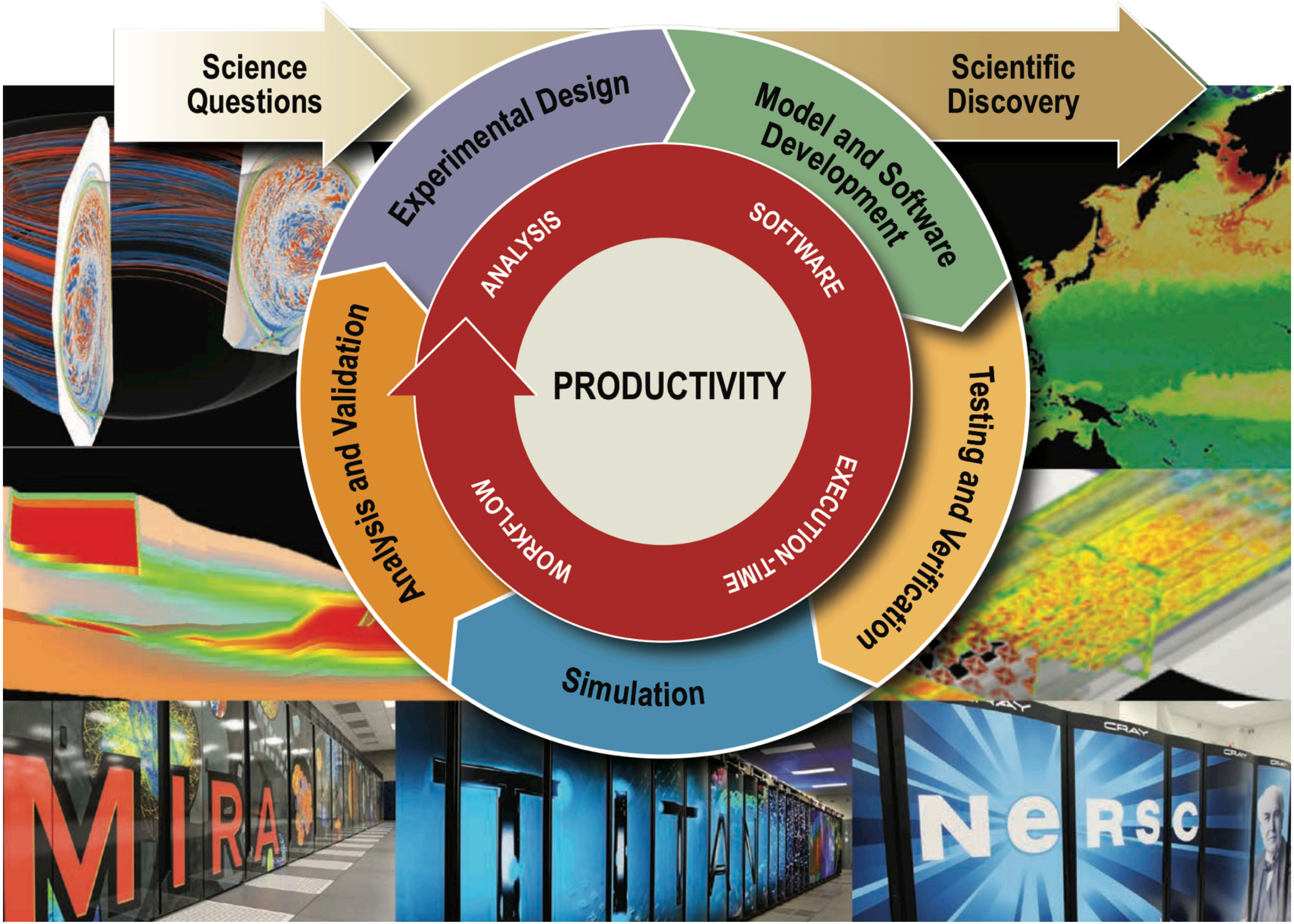


Question		*Formalism*
What can be computed? • *computational power*	**Computability**	*Computation model* • *Church-Turing thesis*
How hard is a problem? • *complexity classes* • *how problems scale*	**Complexity**	*Complexity theory* • *P, NP* • *NP-hard, NP-complete* • *steps (time), space*
Is there concurrency? • *dependencies* • *parallel behavior*	**Parallelism**	*Parallel algorithm* • *scaling models* • *isoefficiency*
How well are requirements of the computation are met by computing resources?	**Performance**	*Parallel programs run on parallel machines* • *theory / simulation* • *empirical evaluation*

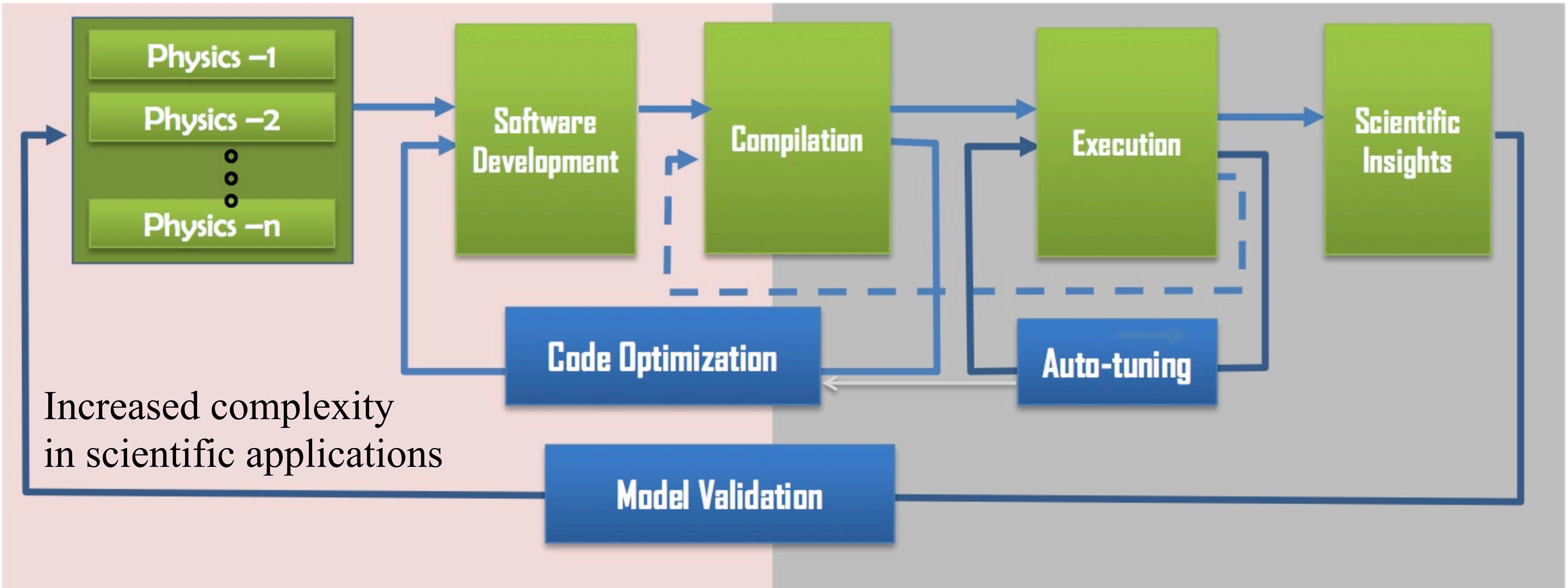
L'existence Précède L'essence

- Performance is the *raison d'être* of parallelism
  - Reduce time to solution
  - Computer larger problems
  - Handle greater complexity
- *Productivity* is the *raison d'être* of (parallel) performance
  - Advance outcomes of *value*
- A (*high-performance) parallel computer* uses advanced technology with high computational *potential*
- Computational potential is delivered to high value outcomes by *realizing* high performance for applications
- Performance is relative to its environment (in context)
  - Machine, application, operating system, …
  - Performance portability and performant applications

(High-Performance) Scientific Productivity



End-to-End Productivity



Application Software Productivity

- Metrics/attributes
- Ease of use
- Heterogeneous portability
- Reusability
- Interoperability/maintainability
- Hardware capabilities discovery
- Software performance engineering

Execution-time Productivity

- Metrics/attributes/execution models
- Observability /controllability/monitoring
- Instrumentation/measurement/ traceability
- Productivity modeling/simulation:
  prediction, diagnosis
- Performance engineering
- **Dynamic performance adaptation**

Parallel Performance, Methodology, and Tools

- What is the nature of parallel performance?
- There are fundamental theoretical issues
  - Performance observation and analysis uncertainty
- Achieving performance is an (empirical) engineering process
  - *Observation*: measure and characterize behavior
  - *Diagnosis*: identify and understand problems
  - *Tuning*: modify to run optimally on high-end machines
- Want the process to be effective and productive
  - What is the nature of the performance problem solving?
  - What is the performance technology to be applied?
- Compelling reasons to build and integrate performance tools
- Parallel systems evolution will drive changes in the technology and process and how they are applied in practice

Performance Technology Eras

- Performance methodology and tools have evolved to serve the dominant architectures and programming models
- ***Observability era*** (1991 – 1998)
  - Instrumentation, measurement, analysis
- ***Diagnosis era*** (1998 – 2007)
  - Identifying performance inefficiencies
- ***Complexity era*** (2008 – 2012)
  - Scale, memory, multicore, accelertor
- ***Productivity era*** (2013 – future)
  - Extreme scale, variance, performance portability, dynamic adaptability, …

TAU Performance System®

Performance Observability (1991-1998)

- Performance evaluation problems define the requirements for performance measurement and analysis methods
- *Performance observability* is the ability to “accurately” capture, analyze, and present understand (collectively *observe*) information about parallel software and system
- Tools for performance observability must balance the *need* for performance data against the *cost* of obtaining it (environment complexity, performance intrusion)
  - Too little performance data makes analysis difficult
  - Too much data perturbs the measured system
- Important to understand performance observability complexity and develop technology to address it

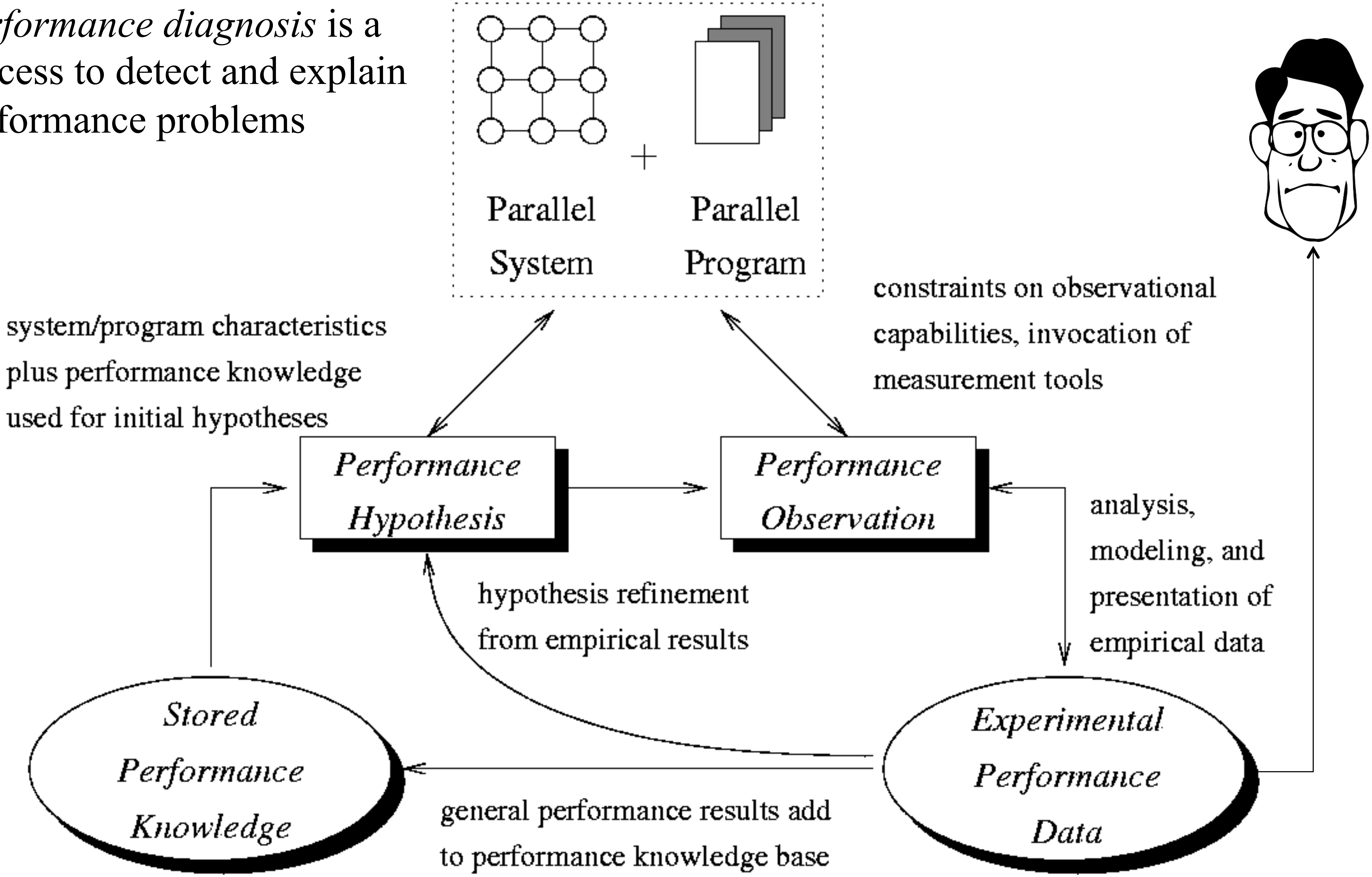
A. Malony, “Performance Observability,” Ph.D. Thesis, University of Illinois, Urbana-Champaign, 1991.

Performance Uncertainty

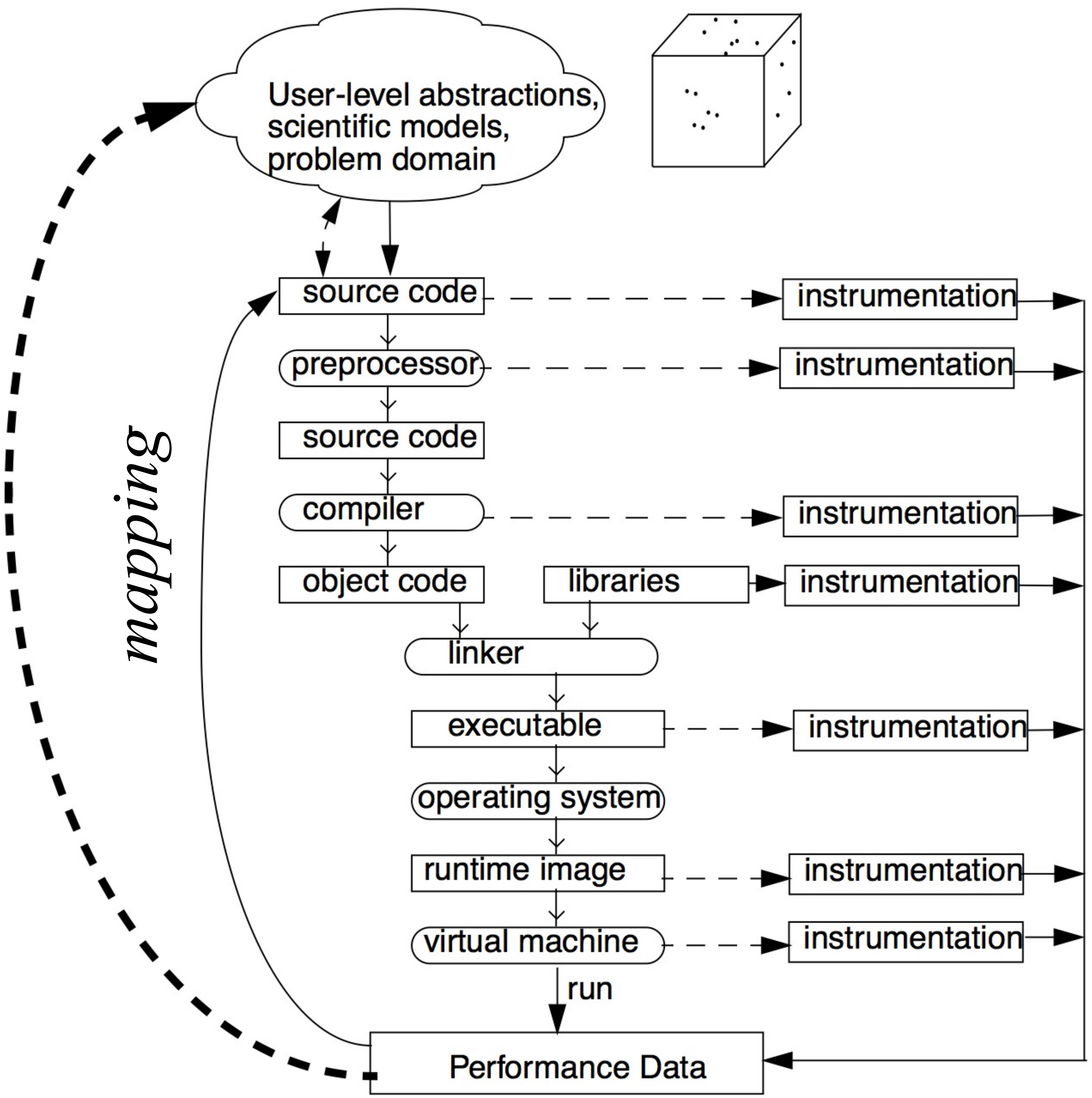
- How do we understand (*true*) parallel performance?
- Performance "science" theory and methodology
- Want to apply to real HPC-class machines
- Performance (observation and analysis) uncertainty
  - Performance analysis requires performance observation
  - Any performance observation will be *intrusive*
  - Any performance intrusion may *perturb* the system state
- Uncertainty applies to all experimental methods
  - "Truth" lies just beyond the reach of observation
- Performance technology must embrace uncertainty
  - Develop performance observation systems that can deliver robust performance data efficiently with low overhead
  - Rationalize about performance measurement effects
    - perturbation analysis, …

Performance Diagnosis (1998-2007)

Performance diagnosis is a process to detect and explain performance problems



Semantic Gap in Performance Mapping



- Semantic entities, attributes, associations (SEAA)
  - *Entities*: represent semantics at any level
  - *Attribute*: encode entity semantics
  - *Association*: link entities across levels to map performance data
- SEAA ability to map low-level data to high levels of abstraction reduces the semantic gap for user

S. Shende, "The Role of Instrumentation and Mapping in Performance Measurement," Ph.D. Thesis, University of Oregon, 2001.

Performance Diagnosis Projects

- ***APART – Automatic Performance Analysis - Real Tools***
  - Problem specification and identification
- ***Poirot*** – theory of performance diagnosis processes
  - Compare and analyze performance diagnosis systems
  - *Heuristic classification*
  - *Heuristic search*
  - Lack of explanation power
- ***Hercule*** – knowledge-based (model-based) diagnosis
  - Capture knowledge about performance problems
  - Capture knowledge about how to detect and explain them
  - Knowledge comes from *parallel computational models*
    - associate computational models with performance models

*L. Li, "*Model-based Automatic Performance Diagnosis of Parallel Computations,*" Ph.D. Thesis, University of Oregon, 2007.*

How to Explain and Understand Performance

- Should not just redescribe the performance results
- Should explain performance phenomena
  - What are the causes for performance observed?
  - What are the factors and how do they interrelate?
  - Performance analytics, forensics, and decision support
- Need to add knowledge to do more intelligent things
  - Automated analysis needs good informed feedback
    - iterative tuning, performance regression testing
  - Performance model generation requires interpretation
- We need better methods and tools for
  - Integrating meta-information
  - Knowledge-based performance problem solving

K. Huck, "Knowledge Support for Parallel Performance Data Mining," Ph.D. Thesis, University of Oregon, 2008.

Performance Complexity (2008-2012)

- Performance tools have evolved incrementally to serve the dominant architectures and programming models
  - Reasonably stable, static parallel execution models
  - Allowed application-level observation focus
- Observation requirements for 1st-person measurement:
  - Performance measurement can be made locally (per thread)
  - Performance data collected at the end of the execution
  - Post-mortem analysis and presentation of performance results
  - Offline performance engineering
- Architecture factors increase performance complexity
  - Greater core counts and hierarchical memory system
  - Heterogeneous computing
  - Significantly larger scale
- Focus on performance technology integration

Evolution

- Increased performance complexity and scale forces the engineering process to be more intelligent and automated
  - Automate performance data analysis / mining / learning
  - Automated performance problem identification
- Even with intelligent and application autotuning, the decisions of what to analyze are difficult
  - Performance engineering tools and practice must incorporate a performance knowledge discovery process
- Extreme scale performance is an optimized orchestration
  - Application, processor, memory, network, I/O
- Application-level only performance view is myopic
- Reductionist approaches will be unsuccessful
- Need for whole performance evaluation

Productivity Era (2012 – ???)

- Challenges of performance growth and power will cause exascale systems to depart from conventional MPP designs
  - Greater core counts and hardware thread concurrency
  - Heterogeneous hardware and deeper memory hierarchy
  - Hardware-assisted global addressing space support
  - Power limits and reliability concerns built in
- Emerging exascale programming models emphasize message-driven computation and finer-grained parallelism semantics
  - More asynchronous and lower-level thread management
  - More exposure of concurrency through task-level parallelism
  - Global address space models versus conventional message passing
  - Heterogeneity in parallel execution and locality optimization
- *Productivity and performance are coupled at exascale*
  - Applications are more complex and must be mapped to systems
  - Growing crisis for performant and maintainable scientific software

Uniformity Assumptions are No Longer Valid

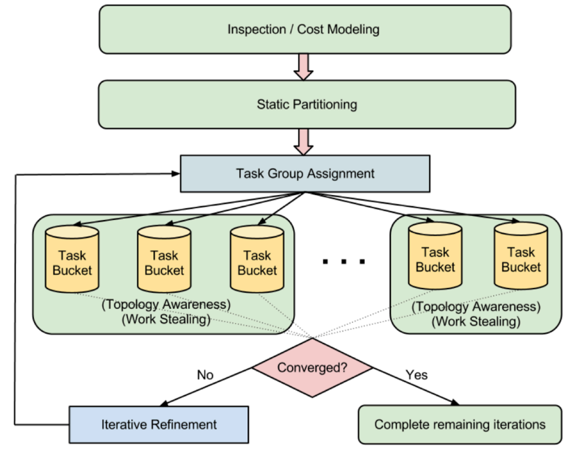
- Exascale design directions raise issues of uniformity
  - Components and behaviors are not the same or regular
  - Heterogeneous compute engines behave differently
  - Fine-grained power management affects homogeneity
  - Process technology results in non-uniform execution behavior
  - Fault resilience introduces inhomogeneity
- Bulk synchronous model is increasingly impractical
  - Removing sources of performance variation (jitter) is unrealistic
  - Huge costs in power/complexity/performance to extend the life
- Embrace performance heterogeneity!!!
  - *Variation, variation, variation*
  - Can not assume a stable "state" of the system *a priori*
  - Post-mortem performance analysis fails for lack of repeatability

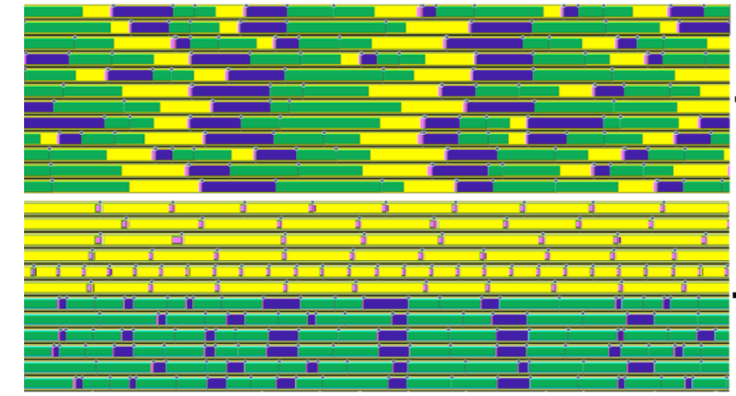
A New Performance "Observability"

- Key exascale parallel "performance" abstraction
  - Inherent state of exascale execution is dynamic
  - Embodies non-stationarity of "performance"
  - Constantly shaped by the adaptation of resources to meet computational needs and optimize objectives
- Fundamentally different performance "observability"
  - "1st person" + "3rd person" performance introspection
  - Designed to support introspective adaptation
  - In-situ analysis of performance state, objectives, and progress
  - Aware of multiple performant and productivity objectives
  - Policy-driven dynamic feedback and adaptation
  - Reflects computation to execution model mapping
  - Integration in exascale productivity environment

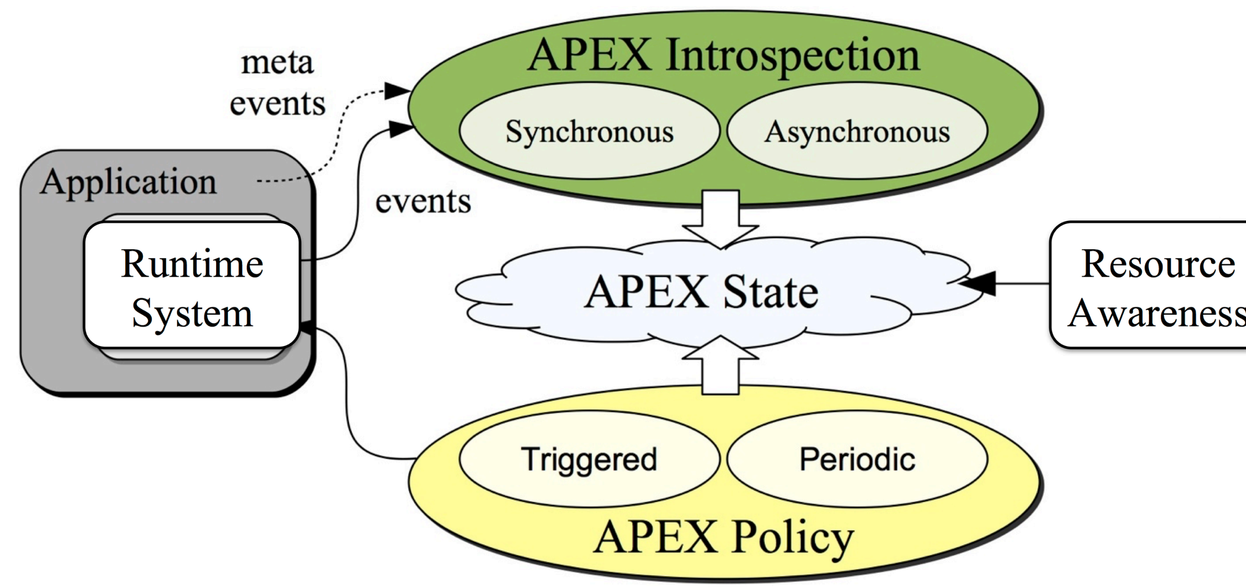
Ph.D. Thesis

- D. Ozog, *High Performance Computational Chemistry*, Ph.D. thesis, December 2016.

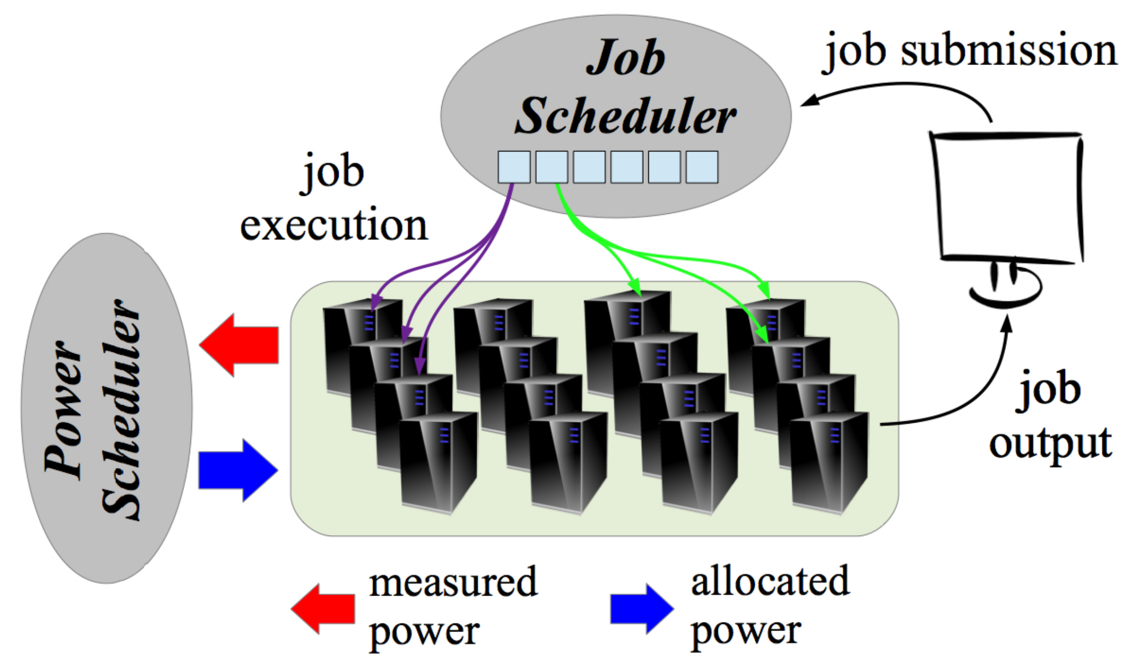




- N. Chaimov, *Insightful Performance Analysis of Many-task Runtimes through Tool-Runtime Integration*, Ph.D. thesis, June 2017.



- D. Ellsworth, *System-wide Power Management Targeting Early Hardware Overprovisioned High Performance Computers*, Ph.D. thesis, June 2017.



TAU History

1992-1995: DARPA pC++ (Gannon, Malony, Mohr). TAU (Tools Are Us) is born. *[parallel profiling, tracing, performance extrapolation]*

1995-1998: Shende Ph.D. (performance mapping, instrumentation). TAU v1.0. *[multiple languages, source analysis, automatic instrumentation]*

Observability

1998-2001: Significant effort in Fortran analysis and instrumentation, work with Mohr on OpenMP, Kojak tracing integration, focus on automated performance analysis. *[performance diagnosis, source analysis, instrumentation]*

2002-2005: Focus on profiling analysis, measurement scalability, and perturbation compensation. *[analysis, scalability, perturbation analysis, applications]*

2005-2007: More emphasis on tool integration, usability, and data presentation. TAU v2.0 released. *[performance visualization, binary instrumentation, integration, performance diagnosis and modeling]*

Diagnosis

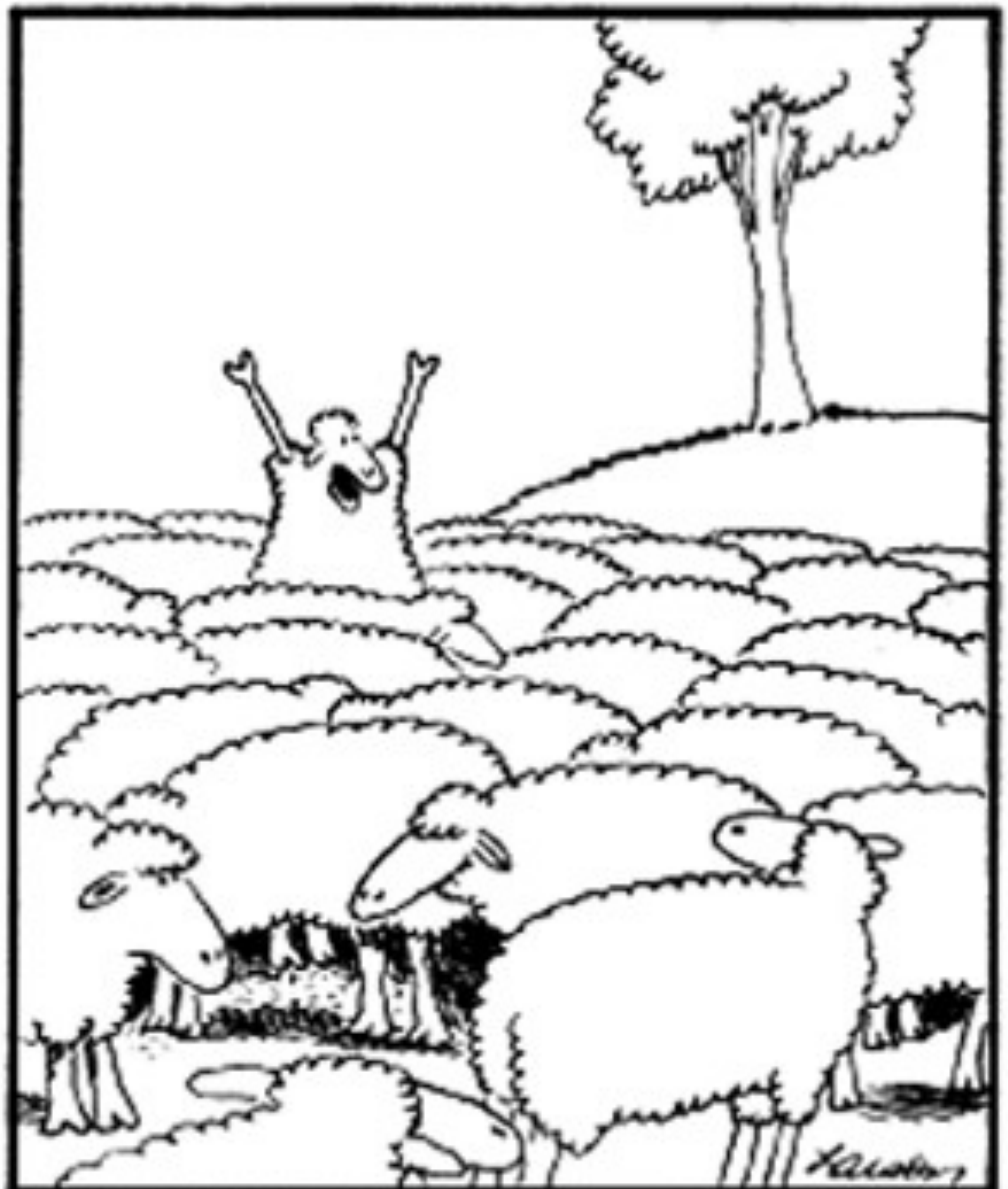
2008-2011: Add performance database support, data mining, and rule-based analysis. Develop measurement/analysis for heterogeneous systems. Core measurement infrastructure integration (Score-P). *[database, data mining, expert system, heterogeneous measurement, infrastructure integration]*

Complexity

2012-present: Focus on exascale systems. Improve scalability, heterogeneous support, runtime system integration, dynamic adaptation. Apply to petascale / exascale applications. *[scale, autotuning, introspection, autonomic]*

Exascale

Parallel Performance is more than the NPB!



"Wait! Wait! Listen to me! ... We don't have to be just sheep!"

Parallel Performance Research Future

"When you come to a fork in the road, take it. – Yogi Berra


I compute, therefore I am!