AIMES

Abstractions and Integrated Middleware for Extreme-Scale Science

Shantenu Jha, Matteo Turilli - Rutgers University

Jon Weissman, Francis Liu - University of Minnesota

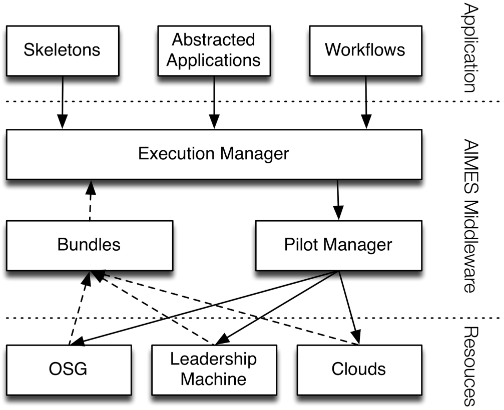
Daniel S. Katz, Zhao Zhang, Michael Wilde - University of Chicago

funded by DOE ASCR under grants:

DE-FG02-12ER26115, DE-SC0008617, DE-SC0008651

Objectives

- Enable extreme-scale distributed computing via dynamic federation of heterogeneous infrastructure.
- Support reasoning about applications and infrastructure:
  - Develop new abstractions.
  - Characterizing performance.
- Develop extensible prototype of middleware enabling experimental exploration of extreme-scale science.



AIMES: Enabling both big and long-tail distributed science at extreme scale through federation of resources.

Progress and Accomplishments

- Model of execution strategy for extreme-scale workloads on distributed resources.
- Experiment-based predictive models for resource availability.
- Implemented skeletons as a model of distributed (and many-trask) applications.
- Implemented federation layer by means of resource overlays.
- Implemented and characterized resource bundles.

Impact

- Identifying design principles of extreme scale distributed resource management.
- Laying the foundations of a conceptual framework for resource federation.
- Understanding role of integrated middleware and how it can be designed for distributed resource management while hiding complexity of distributed computing infrastructure

Project Vision and Overview

- Distributed computing fundamental to extreme-scale science:
    - Enabling both **big** and **long-tail** distributed science at extreme scale through federation of heterogenous resources.
    - Providing capabilities to support applications, consumers, and tools.
- Reasoning about executing distributed workloads:
    - Exploring the principles of distributed execution and spatio-temporal federation of heterogeneous resources.
    - Modeling resource bundles, execution strategies, and application skeletons.
- Improving the ability to utilize diverse and distributed resources:
    - Prototyping the AIMES software stack while examining the importance, challenges, and limitations of integrated middleware.
    - Supporting scientific communities by enabling patterns such as: large ensembles, adaptive applications, and distributed scatter-gather.

Application Skeletons

- Theoretical Contribution: hide application complexity while capturing essential characteristics.
  - **Application Skeleton** is a simple yet powerful tool to build synthetic applications that represent real applications, with similar performance.
- Design and Implementation:
  - Applications are represented by a compact set of parameters:
    - for Bag of Tasks, (iterative) map-reduce, and (iterative) multistage workflow applications.
  - Application Skeleton tool parses these, builds:
    - executables and input data sets.
    - control logic: shell script, Swift script, or Pegasus DAX.

Application Skeletons

- Experiments and results:
  - Skeletons used to successfully model 3 complex multi-stage applications, with similar performance: <3% error per stage and overall
  - Used in UC work to test and show system improvements, e.g. distributed data caching, task scheduling, I/O tuning
  - Used in AIMES to test middleware developments
  - Open source: https://github.com/applicationskeleton/Skeleton

TABLE II. TIME-TO-SOLUTION COMPARISON OF SKELETON MONTAGE AND REAL MONTAGE (SECONDS)

	mProject	mImgtbl	mOverlaps	mDiffFit	mConcatFit	mBgModel	mBackground	mAdd	Total
Montage	282.3	139.7	10.2	426.7	60.1	288.0	107.9	788.8	2103.7
Skeleton	281.8	136.8	10.0	412.5	59.2	288.1	106.2	781.8	2076.4
Error	-0.2%	-2.1%	-0.2%	-3.3%	-1.5%	0.03%	-1.6%	-0.9%	-1.3%

TABLE IV. TIME-TO-SOLUTION COMPARISON OF SKELETON BLAST AND REAL BLAST (SECONDS)

	split	formatdb	blastp	merge	Total
BLAST	74.4	82.1	1996.3	35.9	2188.7
Skeleton	72.9	81.6	2028.9	36.3	2219.7
Error	-1.9%	-0.6%	1.6%	1.1%	1.4%

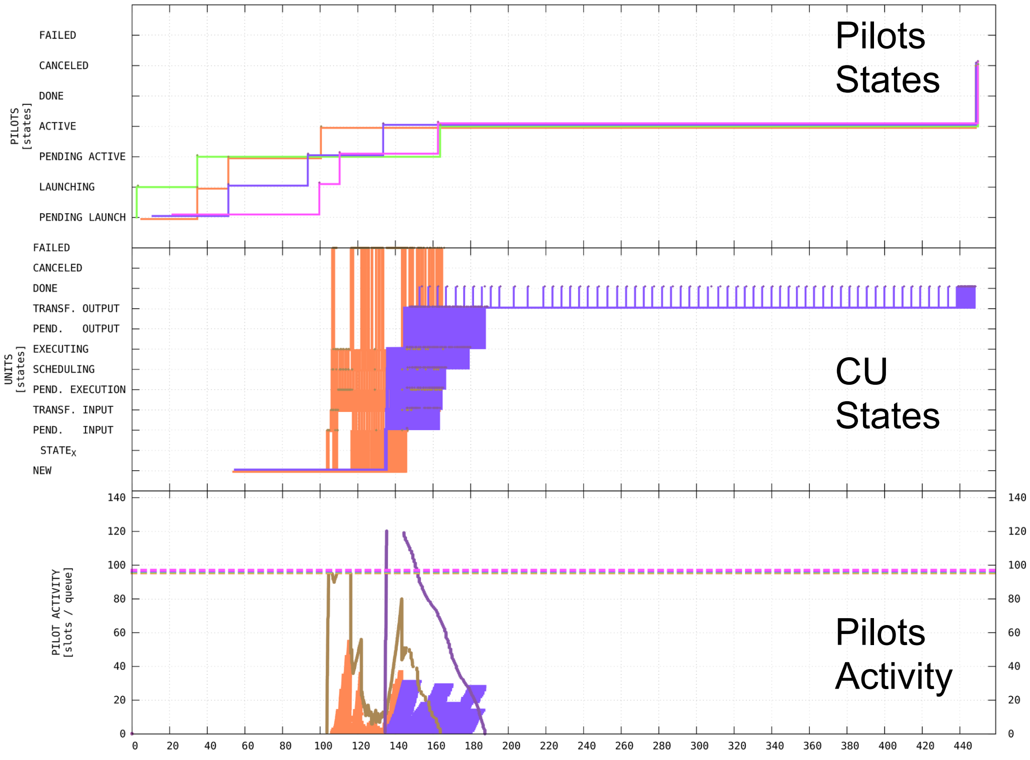
TABLE VI. TIME-TO-SOLUTION COMPARISON OF SKELETON CYBERSHAKE AND REAL CYBERSHAKE (SECONDS)

	Extract	Seis	PeakGM	Total
CyberShake	571.5	2386.5	81.5	3039.4
Skeleton	586.3	2443.3	83.3	3112.9
Error	2.6%	2.4%	2.3%	2.4%

Workload and Resource Management

- Theoretical Contribution: characterize workload description and execution requirements on federated resources.
  - Qualitative: Modeling the concept of ‘execution strategy’ for extreme-scale scientific workloads.
  - Quantitative: Defining key choices that need to be made when executing a given workload; understanding the performance trade-offs of choices.
  - Normative: Providing a consistent representation of execution strategies for a heterogenous set of federated resources.
- Design and Implementation:
  - Pilot-based overlay of heterogeneous resource federation.
  - Transparent workload placement and scheduling algorithms across multiple pilots.

Workload and Resource Management



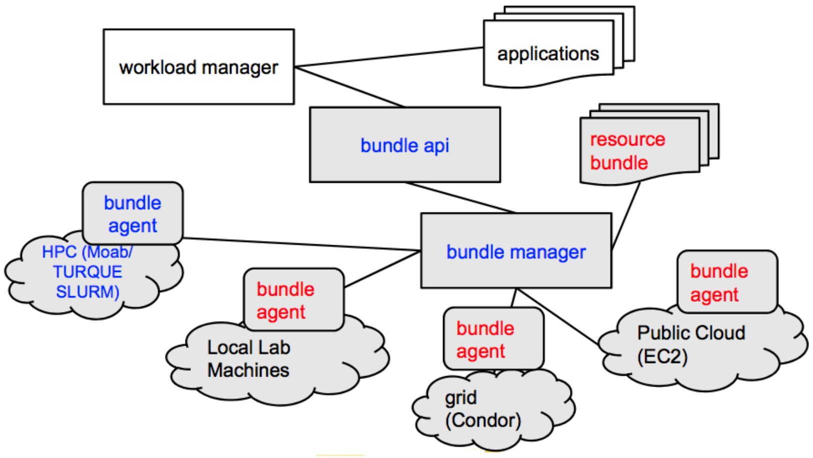
384 CUs executed on 4 96-core pilots on Trestles, Stampede, Gordon, and Blacklight.

- **Overlay** based federation of four heterogeneous resources.
- **Late-binding**: compute units (CU) are bound to a resource dynamically based upon resource availability.
- **Backfilling scheduling algorithm**: given multiple pilots, each pilot is initially filled with CUs and then kept filled as slots free up.

Bundles

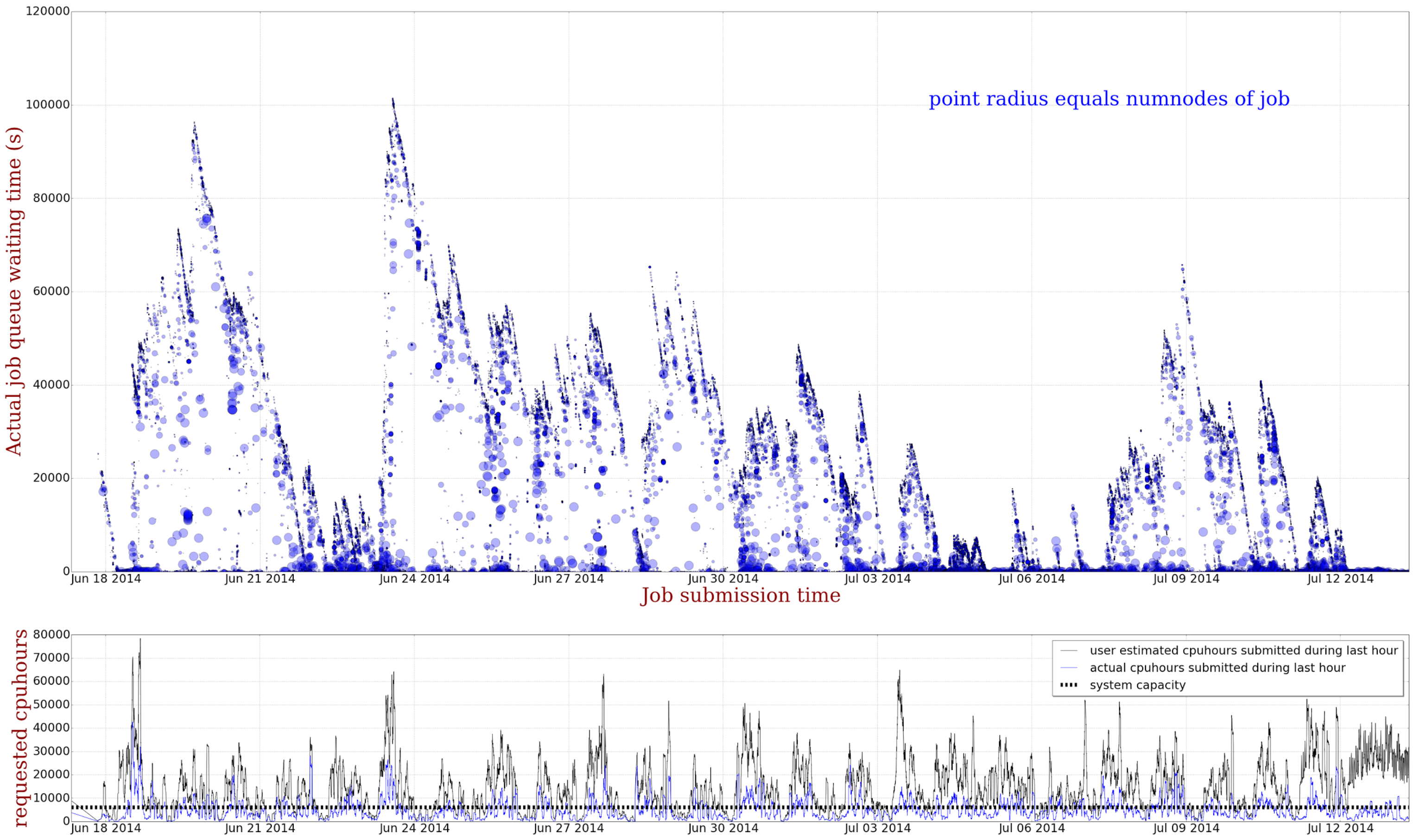
- Theoretical Contribution: define a unifying representation of heterogeneous resources.
  - **Bundles** is an abstraction that provides a characterization of the underlying resource pool.
  - Hides platform-specific details, providing a uniform query interface.
  - Enables automatic, on-demand selection of resources.

- Design and Implementation:



Blue components implemented, e.g., compute Bundle for batch-scheduled supercomputers.

Bundles



Bundle provides resource characterization (XSEDE - Stampede):

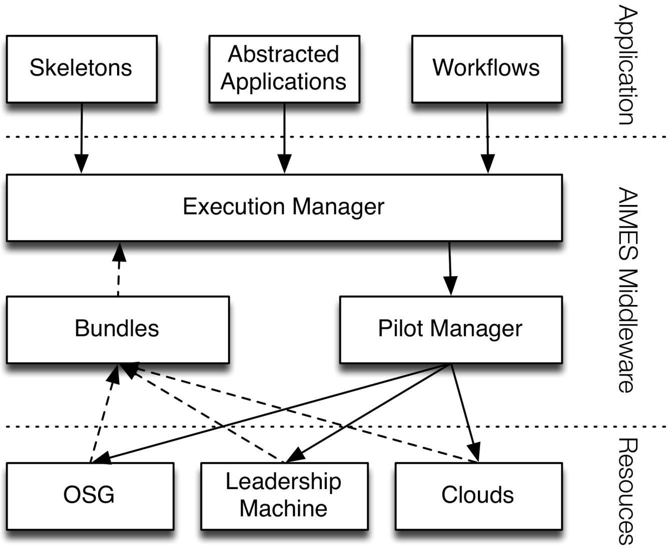
1) Reveals priority for large core-count jobs

2) Reveals skewed distribution of job's waiting time (either very long or very short).

3) Reveals weak correlation between load and wait time distribution

AIMES: Towards End-to-End Integration

- Designed for workload-resource integration:
  - Bundles provide real-time information about the state of diverse resources.
  - Skeletons provide a well-defined description of a given workload.
  - Execution Manager derives an execution strategy matching the workload requirements to the resource capabilities.
  - The execution strategy is enacted.



- Validation of architecture and approach:
  - Functional integration of components; PY1, demonstrated at SC'13.
  - Quantification of advantages; PY2, SC'14.
  - Provide conceptual understanding and reasoning; PY3, SC'15.

What questions does your research motivate you to ask now?

- How to improve the qualitative and quantitative aspects of distributed execution?
  - Qualitative: conceptual and infrastructural complexity.
  - Quantitative: adaptive planning to improve resource utilization; prediction to determine best set of resources to federate.

- What are the design principles and architectures for next generation of distributed computing infrastructure?
  - How to architect infrastructure for specific performance and requirements?
  - How to trade-off between usability and sophistication?

- How to effectively use what is available *versus* how to design what we need?
  - What are the guiding principles? metrics?