Enabling Science and Engineering Workflows on Dynamically Federated Cloud Infrastructure using CometCloud

Center for Autonomic Computing

Hyunjoo Kim, Moustafa Abdelbaky, Manish Parashar

CometCloud

Motivation

Dynamically federated, hybrid cloud infrastructures that integrate private clouds, enterprise datacenters and grids, and public clouds are becoming increasingly important computational platforms that can transform computational and data-enabled science and engineering. Such on-demand, elastic federated cloud infrastructures provide opportunities to:

1. Reduce overheads, improve productivity and qualities of service (such as throughput, time or budget constraints, etc.) for complex application workflows with heterogeneous resource requirements.
2. Enable new science-driven formulations and practices based on on-demand and elastic access to resources, services and data.

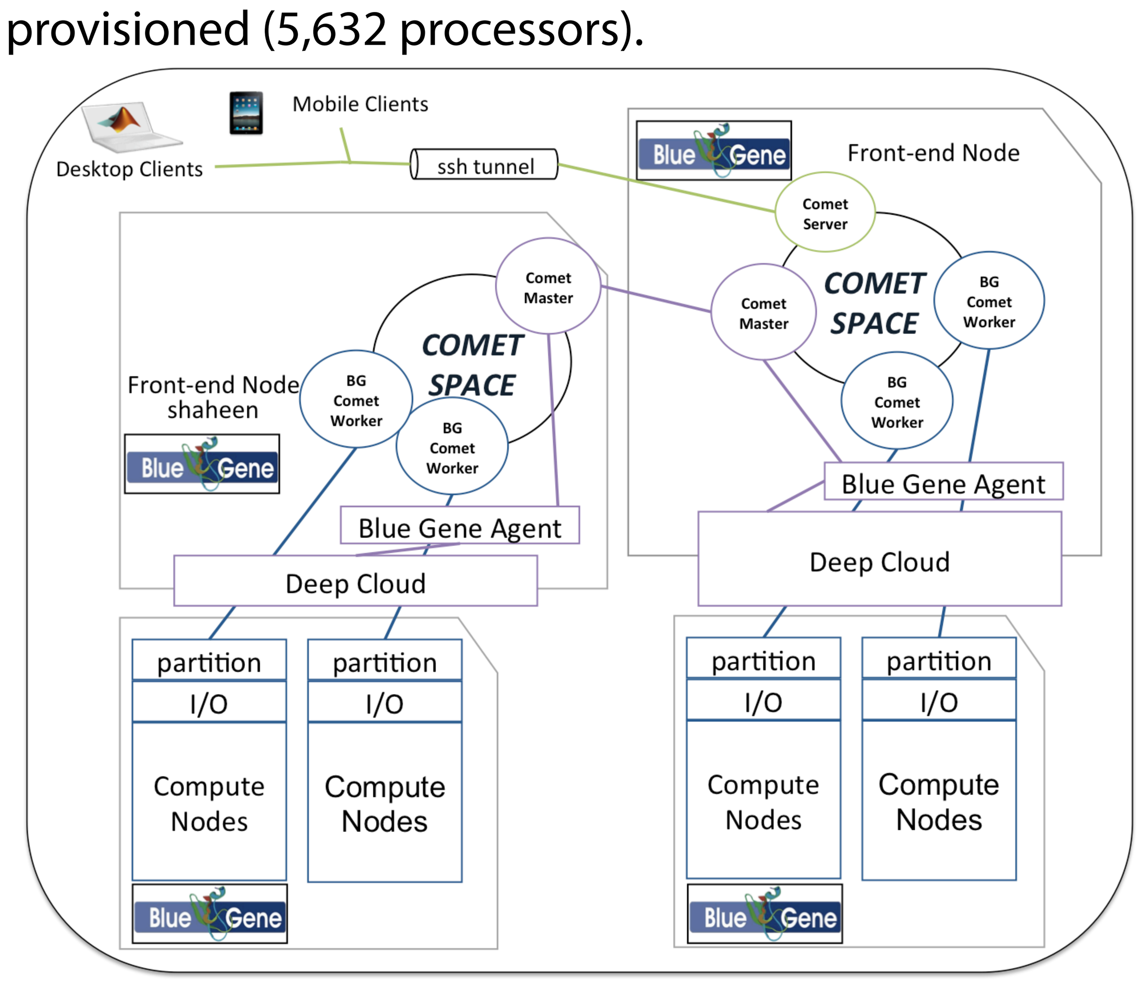
Objectives

The overarching objective of this projects is to enable new paradigms and practices in science and engineering enable by clouds. Specifically, this poster demonstrates how the CometCloud autonomic cloud engine can:

1. Dynamically federate (cloud, Grid, HPC) resources to synthesize an customized elastics virtual clouds.
   - Autonomic cloud-bursting and cloud-bridging
2. Provide programming abstractions for compute/data intensive application workflows
3. Address system and application heterogeneity and dynamics to ensure application objectives and constraints are satisfied.
4. Support autonomic provisioning and system/application adaptation based on policy (deadline, budget, privacy boundaries, etc.), constraints (failure, network, availability, etc.), user requirements, etc.

Use Case 1: HPC-as-a-Service

Objective: Demonstrate how the cloud abstraction can be used to support ensemble geo-system management applications on a geographically distributed federation of HPC resources.
Application: Oil-reservoir history matching application workflow using IPARS (Implicit Parallel Accurate Reservoir Simulator).
Federated infrastructure: IBM BlueGene/P systems at IBM T.J. Watson (US) and KAUST (Saudi Arabia).
Demonstration scenario:
•Initial workflow composed of 1 stage with 10 initial ensemble members provisioning 5 partitions (640 processors) at Watson.
•Request for improved throughput results in elastic scale-up and provisioning 10 partitions (1,280 processors).
•Request or increased level of accuracy results in increasing ensemble members to 150 and results in elastic scale-up and scale-out: all128 partitions available at Watson and 22 partitions at KAUST provisioned (5,632 processors).



Autonomic Application Management Using CometCloud

CometCloud is an autonomic cloud engine that enables:
•*Cloud-bursting*: dynamic scale-out to address dynamic workloads, spikes in demand, other extreme requirements.
•*Cloud-bridging*: dynamic on-demand federation of resource (grids/clouds/clusters/hpc).
•*Autonomic management:* Resource provisioning/application management based on user objectives and application/system state.

Application/Programming layer autonomics: Dynamic workflows; Policy based component/service compositions and adaptations
Service layer autonomics: Robust monitoring and proactive self-management; dynamic application/system/context-sensitive adaptations.
Infrastructure layer autonomics: On-demand scale-out; resilience to failure and data loss; handle dynamic joins/ departures; support "trust" boundaries.

Redbox denotes open source.

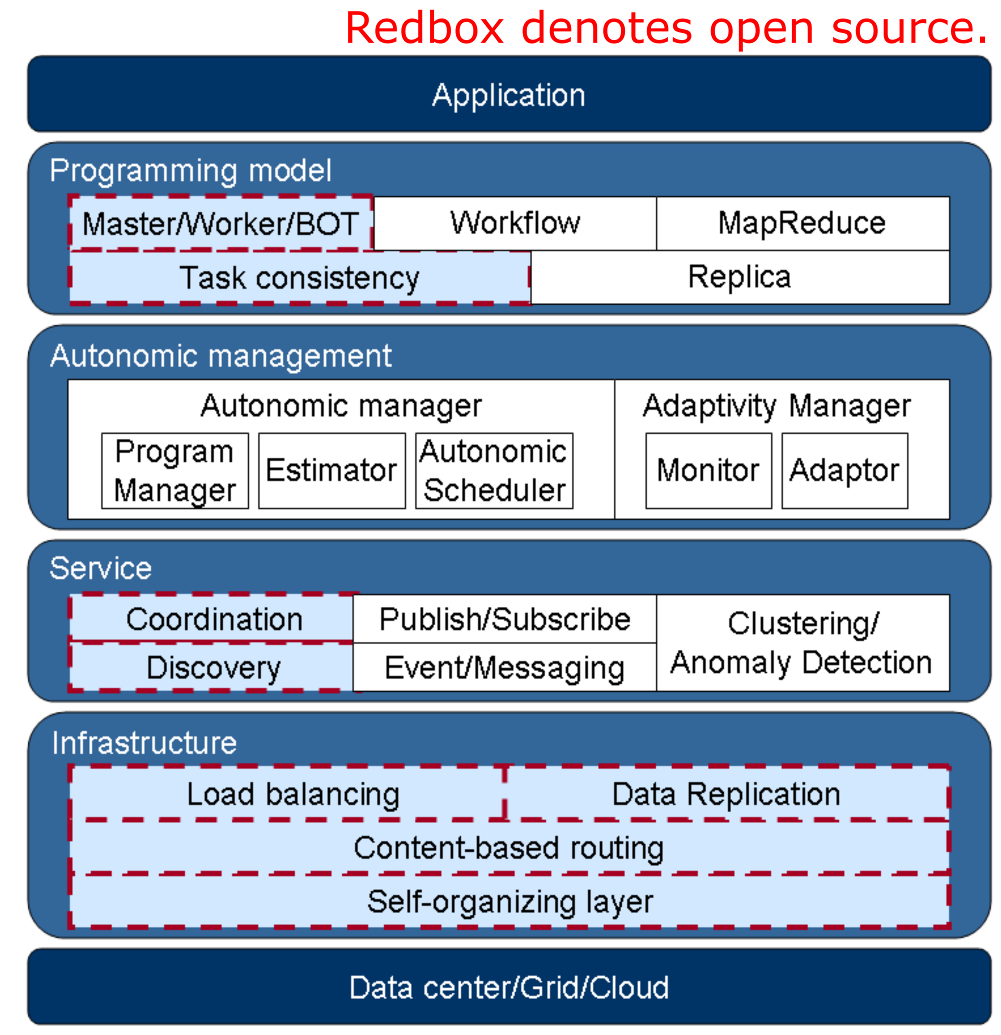


Fig 1. CometCloud architecture stack

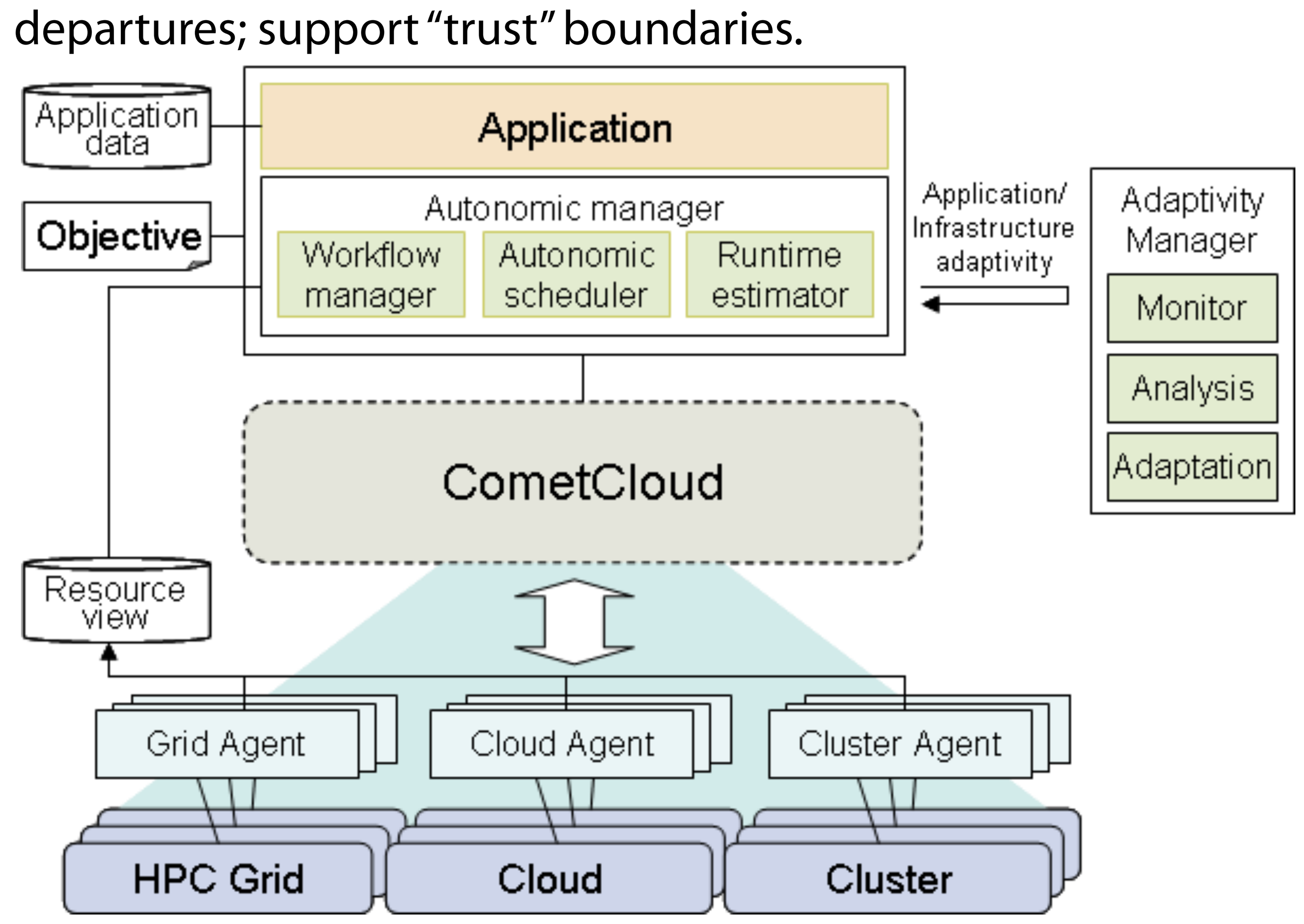


Fig 2. Autonomic application management on a federated cloud

Autonomic Management in CometCloud

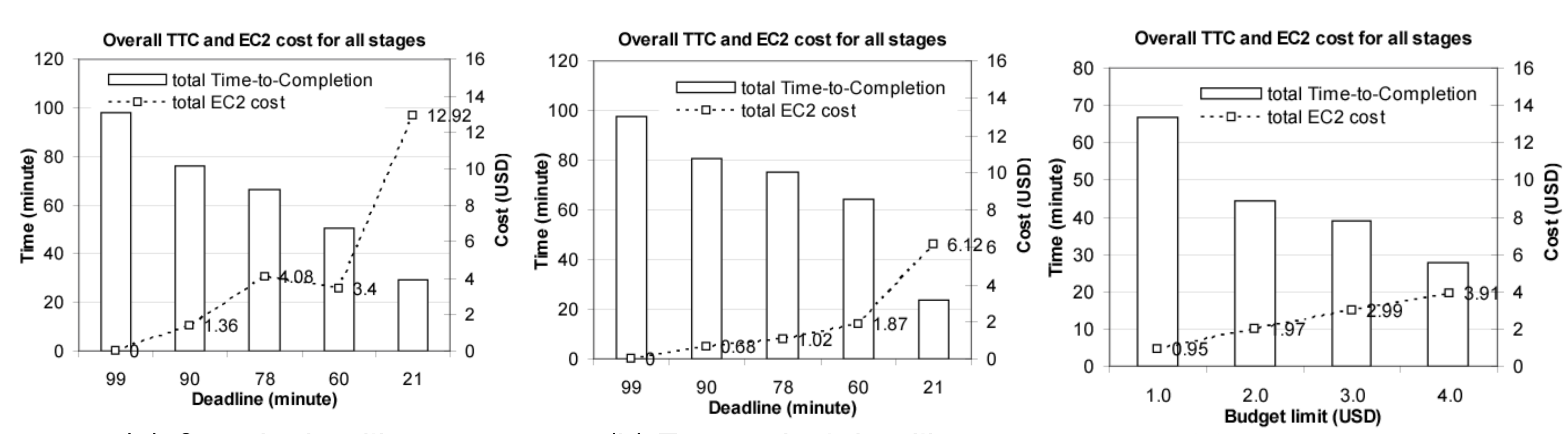
- **Autonomic managers**: Manage overall provisioning, scheduling and execution:
  - Workflow manager coordinates the execution of the application workflow
  - Runtime estimator translate hints about computational complexity provided by the application into runtime and/or cost estimates on specific resources.
  - Autonomic scheduler clusters these to identify potential scheduling blocks and computes anticipated runtimes for these blocks on available resource classes.

- **Adaptivity manager**: Monitors performance for each application task, compares its runtime with the estimated runtime, and checks if there is a possibility that a user constraint will be violated. If so, it adjusts resources and scheduling for the remaining tasks by interacting with the autonomic manager.
- **Grid/Cloud/Cluster/HPC agents**: Manage local cloud resources; access tasks from CometCloud and feed them to their local resources; gather results from local workers and send them to the workflow manager.

Use Case 2: Accelerating the runtime of HPC using cloud resources

Objective: Explore how cloud resources can accelerate HPC applications.
Application description: Ensemble kalman filter based BlackOil reservoir simulator workflow based on the Cactus scientific computing framework and the Portable Extensible Toolkit for Scientific computing (PETSc). Kalman filter runs at the end of every stage to decide the parameters for the next stage. Workflow composed of 3 stages, 128 ensemble members with varying size and computing requirements per stage.
Federated infrastructure: TeraGrid and 5 Amazon EC2 instance types (m1.small, m1.large, m1.xlarge, c1.medium, c1.xlarge)
Scheduling decisions (resource class selection) to achieve deadline constraint

- Greedy deadline: c1.xlarge (the fastest but expensive)
- Economical deadline: c1.medium (good runtime but cheap)
- Budget limit: c1.medium and m1.small



(a) Greedy deadline (b) Economical deadline (c) Budget limit

China – US Software Workshop, September 2011
The Center for Autonomic Computing is supported by the National Science Foundation under Grant No. 0758596.

RUTGERS THE STATE UNIVERSITY OF NEW JERSEY