Managed Virtualized Platforms: From Multicore Nodes to Distributed Cloud Infrastructures

Ada Gavrilovska
Karsten Schwan, Sanjay Kumar, Ripal Nathuji
Mukil Kesavan, Hrishikesh Amur, Adit Ranadive

Center for Experimental Research in Computer Systems
Georgia Institute of Technology

Compute Clouds Today

- Amazon EC2, Yahoo-Intel-HP, IBM Blue Could, VMware vCloud, Microsoft, …
- Emerging as a promising computing platform which addresses
  - technology heterogeneity and complexity, management, reliability, energy costs, developers of systems software and services, …
  - or simply result of "economics and current technology"
- Not just "Grid" revisited
  - ubiquitous presence of virtualization technology
  - manycore nature of hardware components
  - present-day concerns
  - nature of current and future workloads and usage models, programming paradigms…

Objectives

Effective, scalable management infrastructure:

- Develop solutions to *effectively manage the aggregate resources* on the individual *multicore nodes* and across the entire *distributed virtualized cloud infrastructure*

Specifically:

- Improve platform resource utilization
  - CPU cores, memory, IO bandwidth, …
- Create additional opportunities for consolidation and sustainable VM throughput
- Reduce resource requirements for existing VM loads
  - Energy efficiency
- Provide guest/client VMs with performance guarantees

Virtualized Platform Management

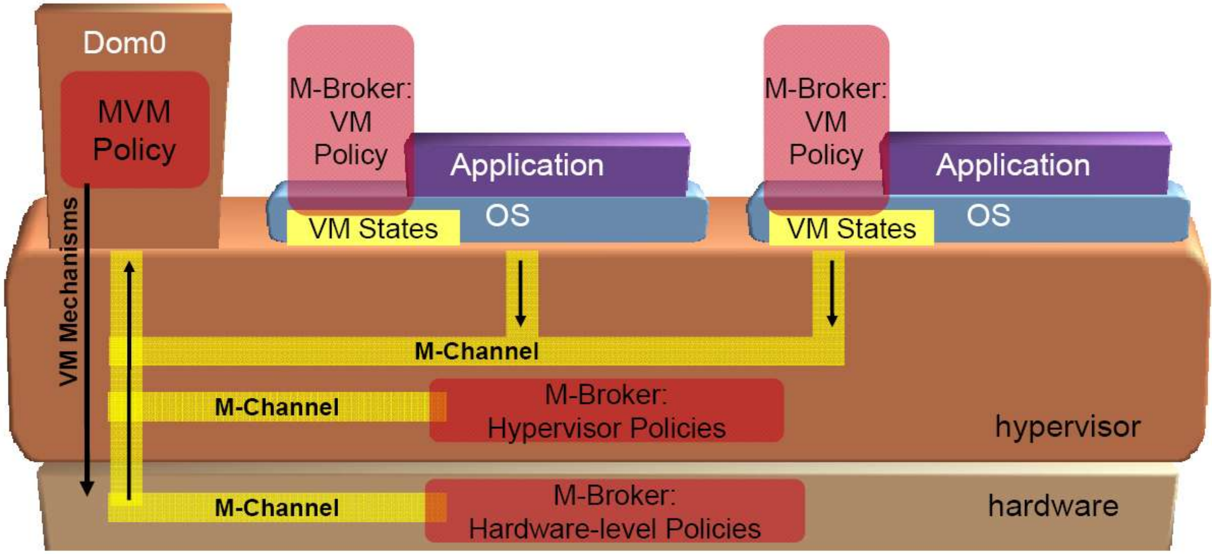
Challenges

- Quality of Service:
  - meet expected VM-level SLAs
    - SLA metric?
  - individual as well as sets of VMs
- Dynamism:
  - deal with bursty application/VM behavior
  - enable good resource utilization
    - static, worst-case allocation policies insufficient
- Coordination:
  - across nodes and sites
  - across multiple VMs' and their policies for management of virtual resources
    - e.g., VMs' OSs make conflicting decisions regarding platform power mgt
  - across different management layers
    - e.g., HP's iLO management hardware and VMM CPU scheduler
  - allocation decisions regarding one resource type require adjustments to other resources
    - e.g., IO buffer size and CPU scheduling

Management Architecture

- Management brokers
  - make and enforce ‘localized’ management decisions
    - within VMs
    - VMM-level – CPU scheduling, allocation of memory or device resources, ..
    - at hardware level

- Management channels
  - enable inter-broker coordination through well-defined interfaces
  - event and shared memory based
- Management VMs
  - platform wide policies and cross-platform coordination



Representing Platform Resources

- Platform Units
  - vector representing aggregate platform resources and properties
    - CPU, memory, IO, power budget, …
    - reliability, trust, architecture type …
- Class of Service
  - mapping of VM’s SLA to vector of resource requirements
  - actual resource allocation is continuously refined based on VM profile, specific input or runtime behavior
    - static CoS-level (Gold, Silver, Bronze) determines initial allocation and fluctuation limits
    - dynamically adjust runtime allocation within specific boundaries
- Compensation Credits
  - encourage VM’s participation in management processes

Resource Allocation Policies

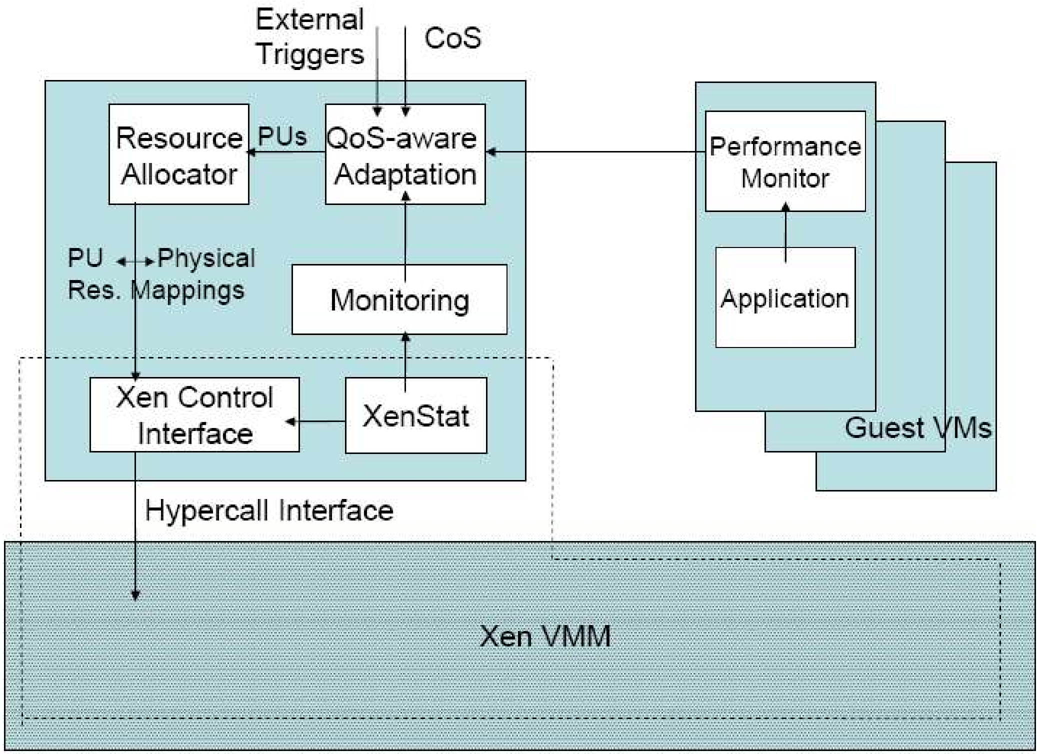
- Enforced within platform level management VM
- External rules
  - static CoS specifications
  - well-understood exceptions
- VM inputs
  - management agents in VMs’ OSs or applications
    - e.g., platform power states
    - e.g., application agents leveraging WSDM standards
- Observation-based
  - black-box runtime monitoring of per VM resource utilization
  - support for range of algorithms, machine learning or statistical techniques…
- Profile-based
  - rely on offline analysis of VM behaviors for classes of workloads, correlation techniques, etc…

Current Realization

- Xen 3.1 – Runs as a Dom0 application. Uses XenStat and Xen Control interface to monitor and actuate resource usage and allocation respectively.

- ESX 3.0.1 – Runs as a CoS application. Uses VSI for monitoring and actuating resources usage and allocation respectively.

- Several different resource allocation policies
  - e.g., VM input for SLAs and power state mgt; algorithm based on TCP AIMD for CPU & IO mgt, …



Experimental Evaluation

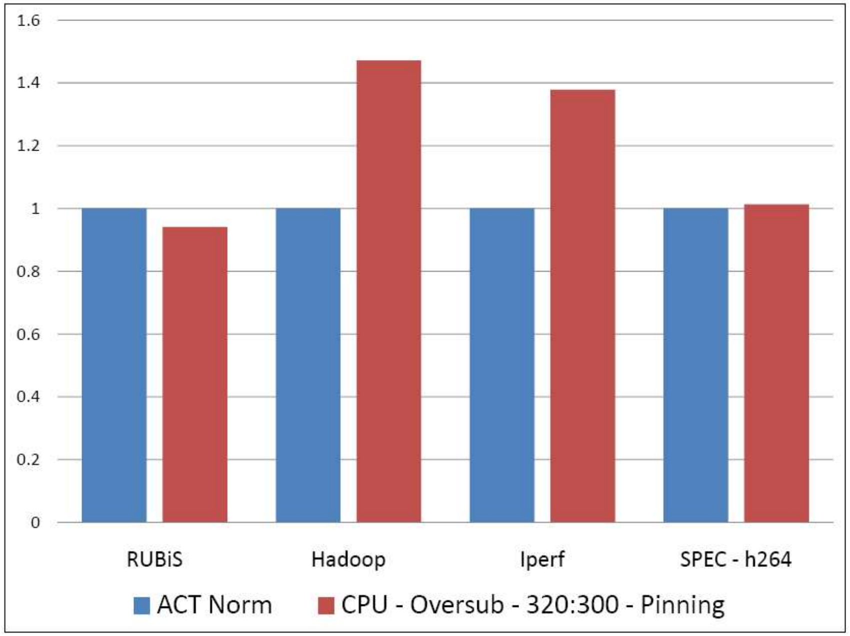
- Testbed:
  - Multiple dual-socket quad-core x86 nodes
  - Interconnect: Ethernet or InfiniBand fabric
    - In case of IB Xsigo VP780 I/O Director switch; Ethernet vnics exported to VMs; Ethernet – InfiniBand translation performed in control domain
- Management brokers:
  - CPU management through vCPU percentage limits and pinning
  - IO management through QoS limits to vnics enforced via Xsigo switch or through token buffer in dom0/vmkernel
  - Power management through DVFS (and C-states)

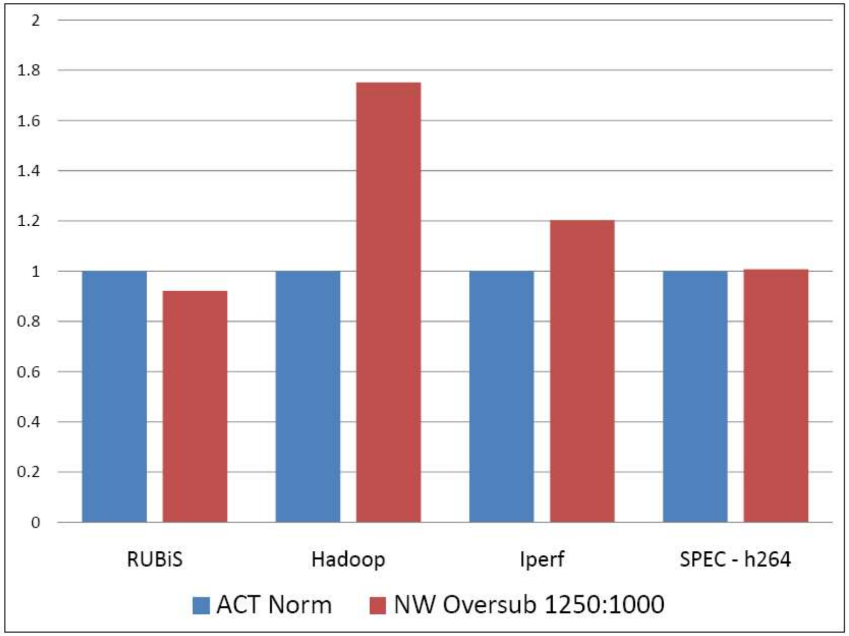
- Workloads:
  - Gold: 80% CPU, 200Mbps; Silver: 60% CPU, 125Mbps; Bronze: 40%, 75Mbps

  - RUBiS: All 3 VMs Gold. Requests per second – the more the better.
  - Hadoop: Master VM Gold, Slave VMs Bronze. Execution time – the lower the better.
  - Iperf: Silver VMs. Throughput – the more the better.
  - Spec-h264ref: Gold VM. Execution time – the lower the better.

Ability to distribute resource based on VMs importance

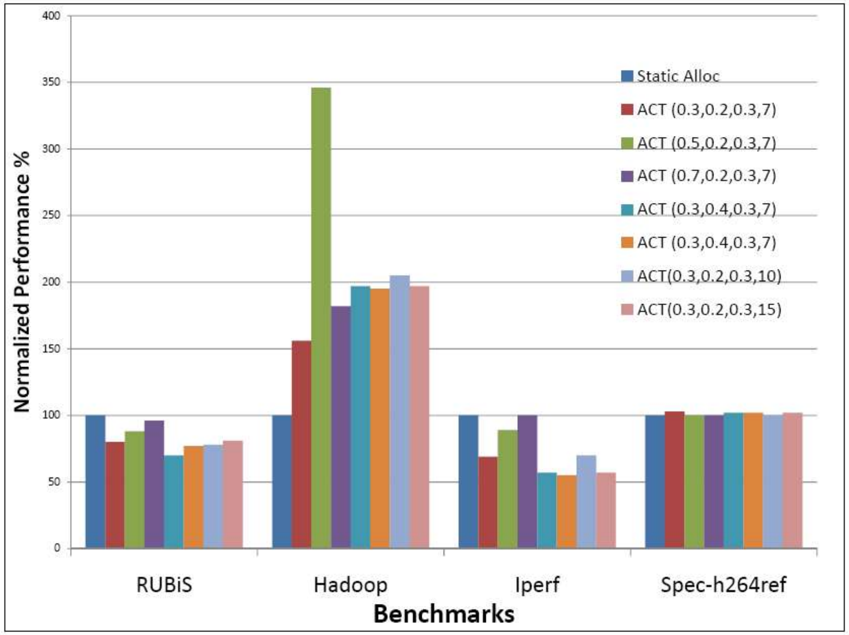
- In under-provisioned platforms performance penalty is shifted to VMs with lowest CoS

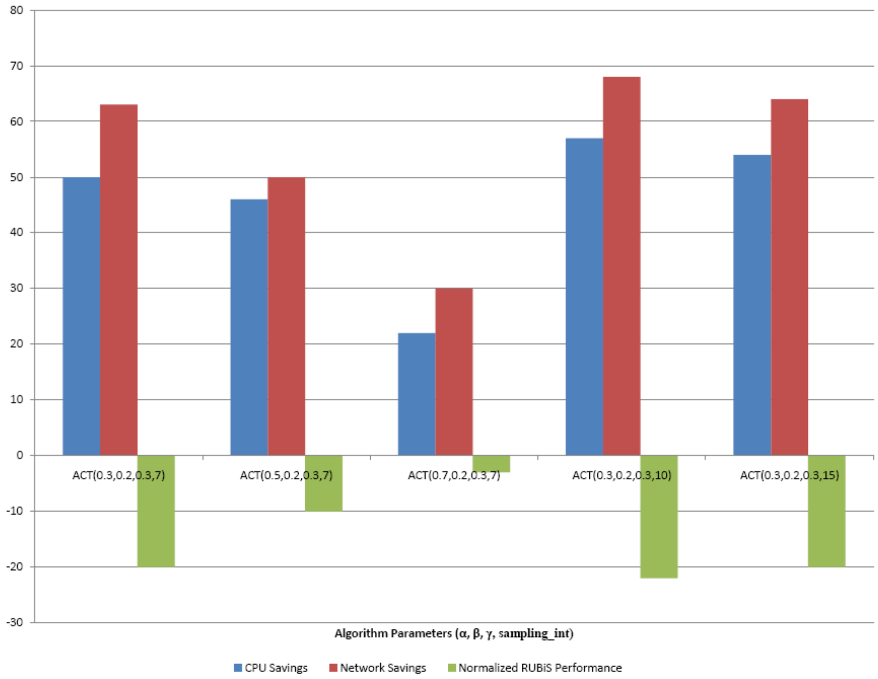




Tradeoffs between resource consolidation opportunities vs. attainable performance

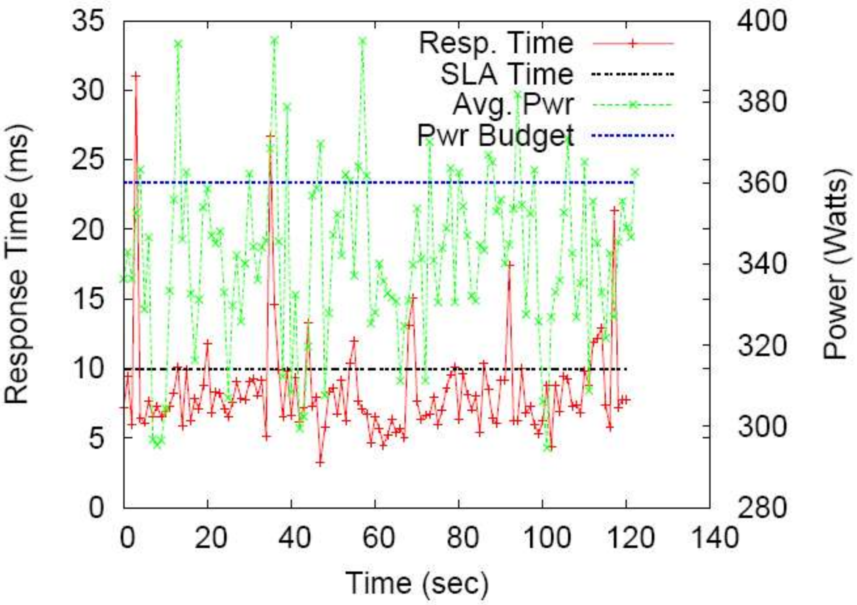
- Negligible level of SLA violation with 20-30% lower resource utilization
- With less than ½ of the platform capacity maximum 20% performance penalty for some requests
- Algorithm parameters used to control tradeoff

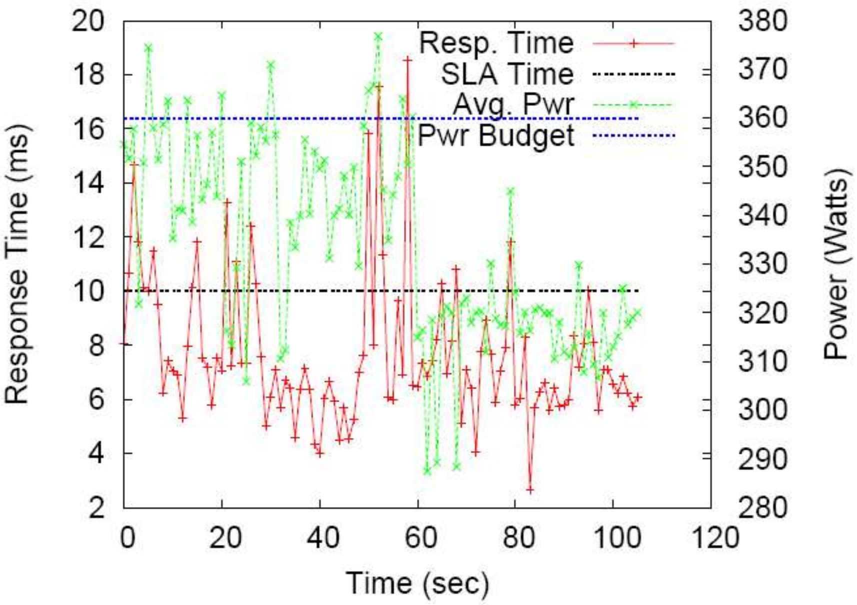




Coordinated CPU and Power Management

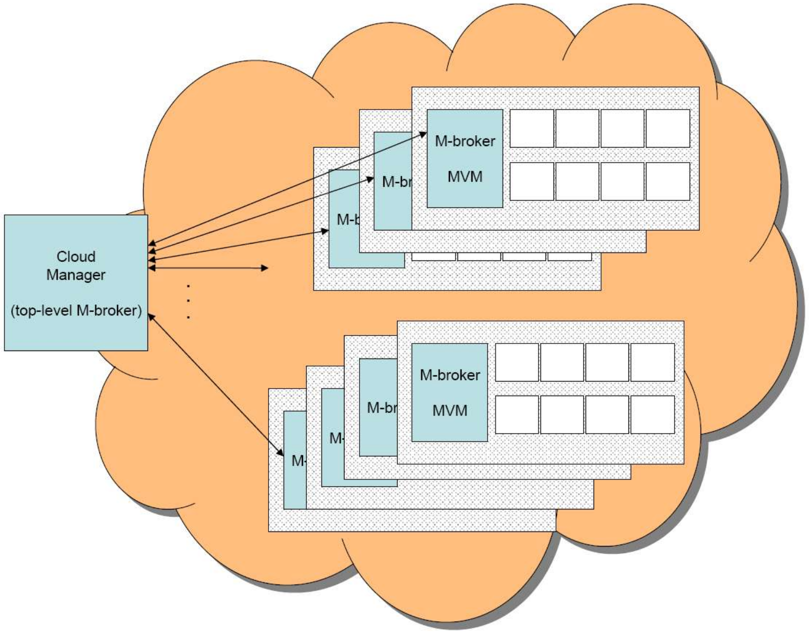
- Lack of coordination (left) triggers repeated oscillations in resource utilization
- Coordination reduces violations and helps determine migration thresholds





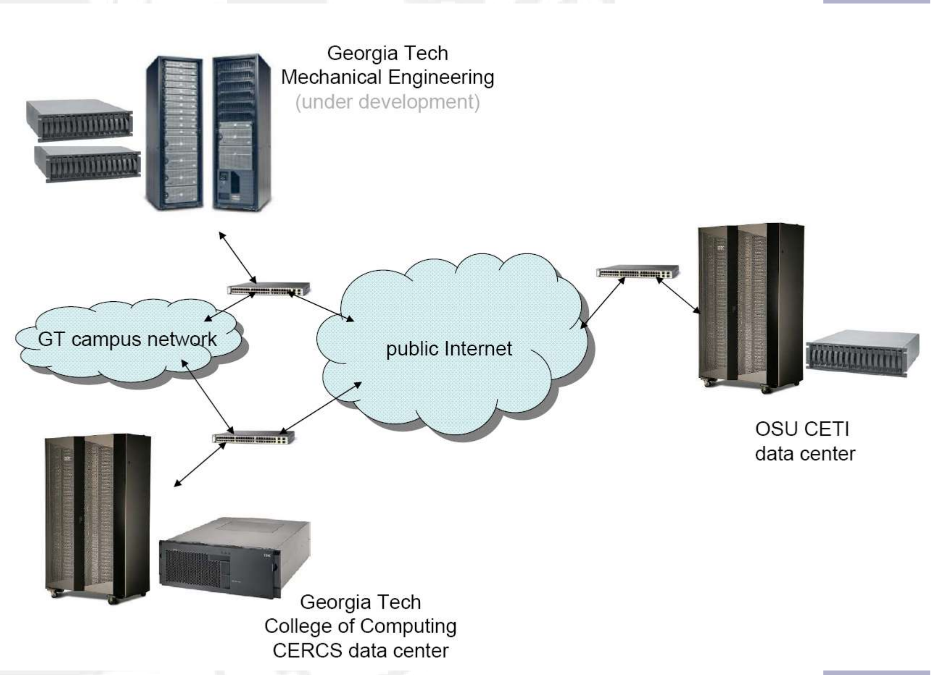
Towards Distributed Clouds

- Build overlays between platform-level management domains
  - leverage our group's high performance eventing middleware EVPath
- Placement of management logic
  - centralized at top level
  - distributed clustered hierarchies
  - localized at individual nodes for low-latency decisions
- Introduce statistical guarantees for allocation of shared resources:
  - e.g., guarantee bandwidth 150Mbps 95% of the time.



CERCS Distributed Cloud Infrastructure

- Enable efficient resource sharing by enterprise workloads with dynamic behaviors
- Critical Enterprise Cloud Computing System (CECCS)
  - infrastructure supported by IBM
  - CERCS Georgia Tech and OSU resources
  - additional GT locations
    - CEETHERM at ME
- Green Clouds
  - extend with additional monitoring and actuation capabilities
    - sensors
  - coordinated management of IT- and environmental facilities-level properties


Georgia Tech
Mechanical Engineering
(under development)
GT campus network
public Internet
OSU CETI
data center
Georgia Tech
College of Computing
CERCS data center