GT Cloud ISTC Research Overview

CERCS Research Center
Georgia Tech

Greg Eisenhauer
Ada Gavrilovska
Ling Liu
Calton Pu
Karsten Schwan
Matt Wolf
Chengwei Wang
Deepal Jayasinghe

http://www.istc-cc.cmu.edu/

Dynamically Elastic Platforms: Hardware-> Applications

Cloud ISTC@GT:

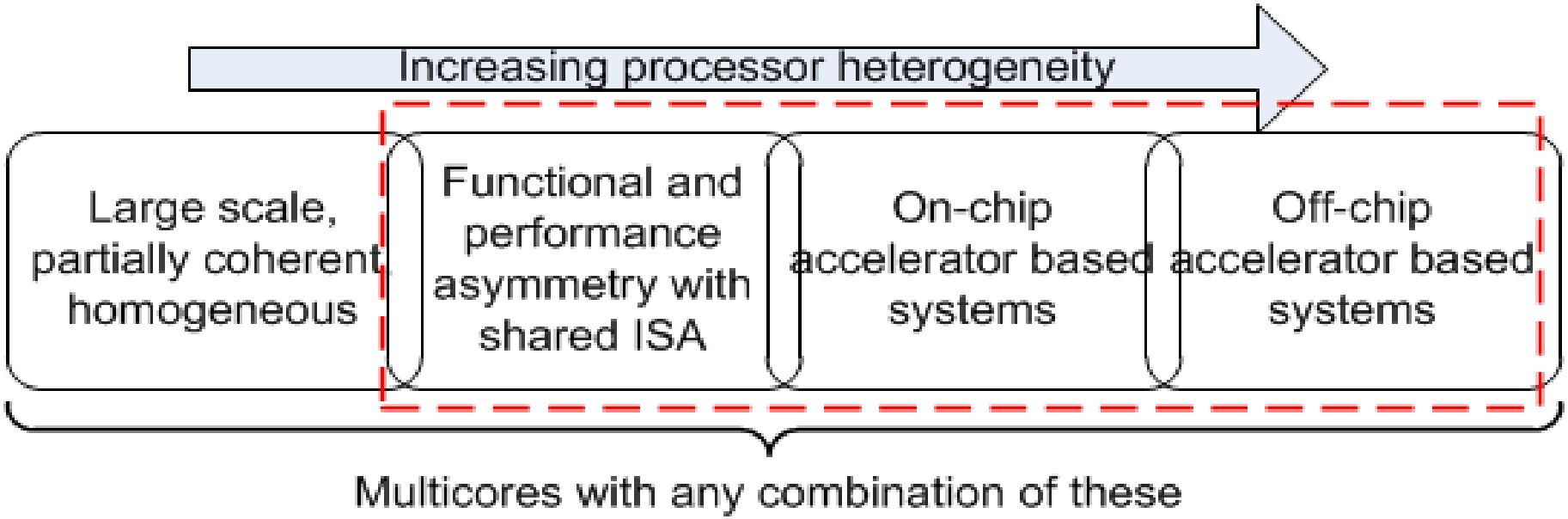
Formulate and carry out a research program that:

- Extends across the “Hardware –> Application” Stack, and
- Cross-links with other ISTC research at multiple levels of the stack, via the
- Jointly defined ‘Pillars’: Big Data, Automation, and Specialization.

Elastic Hardware Platforms I

Problem:

Future hardware has a spectrum of heterogeneity and scale



- Future hardware/software platforms that elastically:
  - service this range of heterogeneity,
  - cause minimal changes to applications over course of hardware evolution, and
  - Improve power/performance.

Elastic Hardware Platforms II

Examples:

- Heterogeneous and asymmetric multicore systems:
  - 10s & 100s of cores; typically virtualized; platform-level asymmetry & integrated accelerators; performance and functional asymmetries

- Our platforms: Current:
  - (1) Multicores 'made' heterogeneous; (2) PCI-attached accelerator systems – NSF Keeneland; (3) Intel's `QuickIA' Atom/Xeon experimental asymmetric platform (and successors)

- Our platforms: Future:
  - (1) MIC and Xeon Systems; (2) GPU-based multi-petaflop DOE machine (Titan – ORNL) – DOE Exascale effort; (3) Others???

GT faculty: Ada Gavrilovska, Karsten Schwan, Sudha Yalamanchili

Intel collaborators: Rich Uhlig's group: Rob Knauerhase, Sanjay Kumar, Jeff Jackson, Scott Hahn, Dulloor Rao, ...; George Cox; Ganapati Srinivasa

Elastic Hardware Platforms III

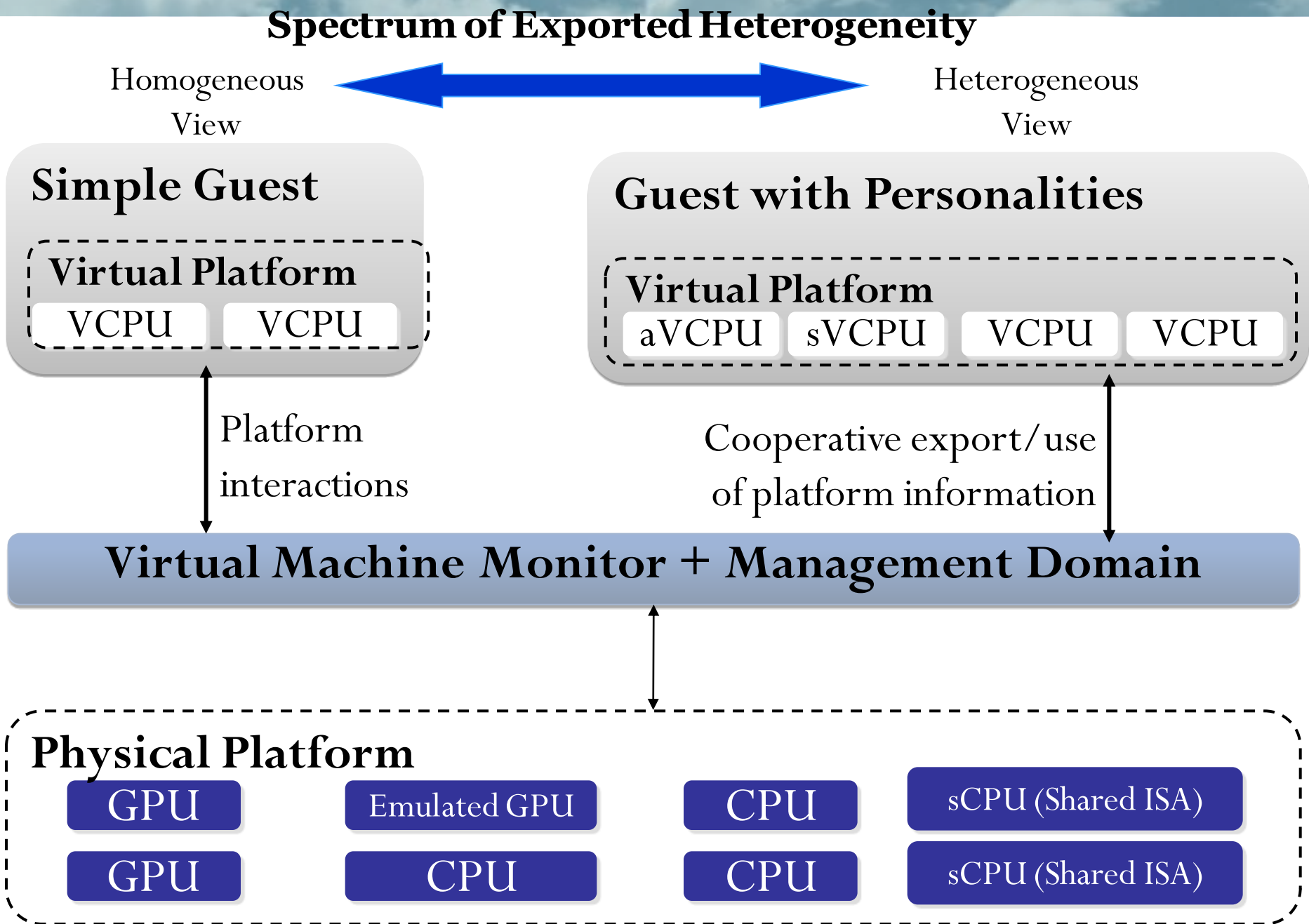
Approach:

- *Dynamic elasticity*: create hypervisor and tool support to support a dynamic range of applications on diverse heterogeneous platforms

  hypervisor as platform to experiment with elasticity

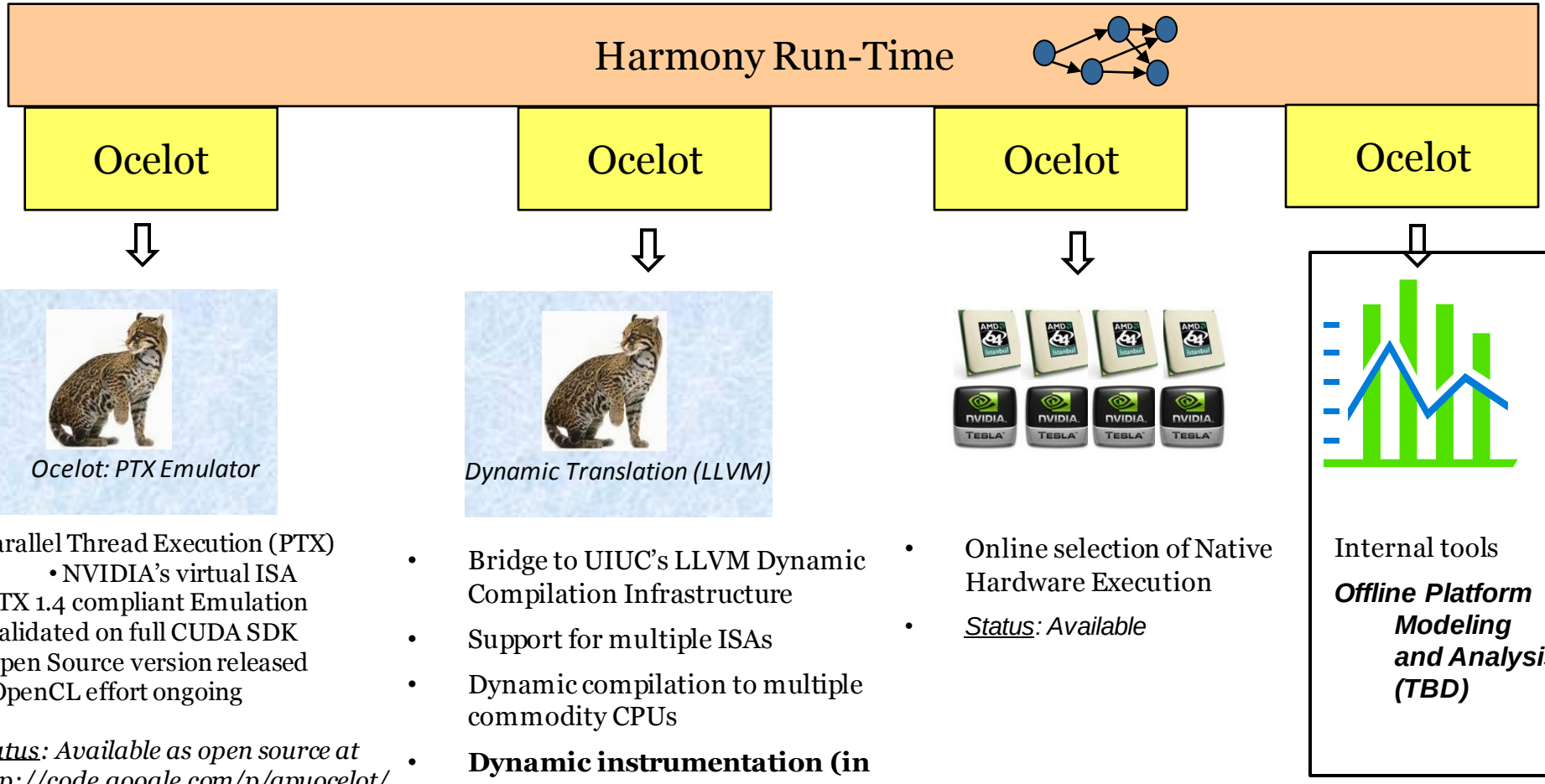
- Research Program:
  - Hypervisor support for increased platform 'fungibility' and 'diversity': VMs with multiple 'personalities' on heterogeneous platforms (slide!)
  - Tools to enable/improve + evaluate platform diversity: `Ocelot' runtime specialization via code generation and emulation (slide!)
  - Scheduling and resource management methods that deal with diversity, i.e., multi-personality VMs, support different optimization metrics, multiple resources (paper submitted)
  - System support for complex memory structures ('memory-based' scheduling; isolation and shared resources; persistent RAM) (ASPLOS 2012, addt'l papers early 2012)
  - 'At Scale': island-based resource management (in submission)

VMs with Multiple Personalities



Ocelot Dynamic Exection Infrastructure

Sudha Yalamanchili – Tool Support for runtime specialization



- Parallel Thread Execution (PTX)
  - NVIDIA's virtual ISA
- PTX 1.4 compliant Emulation
- Validated on full CUDA SDK
- Open Source version released
- . OpenCL effort ongoing

Status: Available as open source at http://code.google.com/p/gpuocelot/

- Bridge to UIUC's LLVM Dynamic Compilation Infrastructure
- Support for multiple ISAs
- Dynamic compilation to multiple commodity CPUs
- **Dynamic instrumentation (in progress)**

Status: Available December 2009 http://code.google.com/p/gpuocelot/

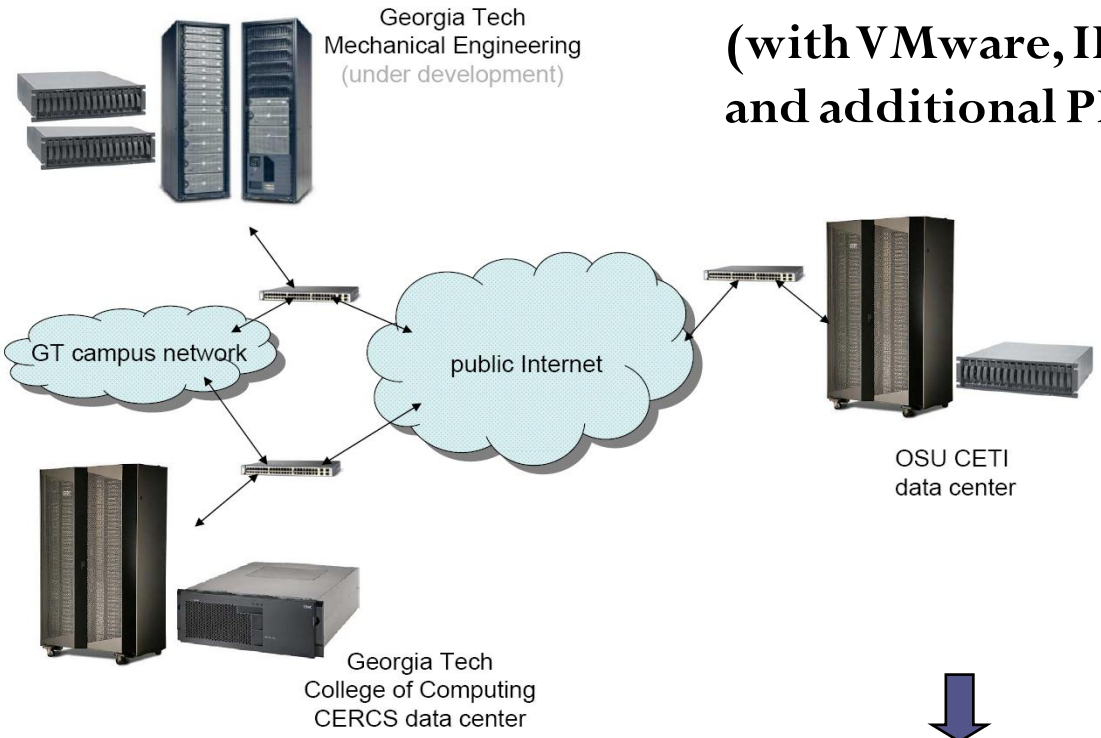
- Online selection of Native Hardware Execution
- *Status: Available*

Select Insights

- Coordination in personality scheduling is critical to application performance
- Ocelot tools can (i) 'explain' performance, through emulation + (ii) 'improve' it, through dynamic re-targeting (i.e., increased fungibility)
- Hypervisor 'fault and migrate' support can be efficient basis for heterogeneous core management, incl. for 'memory scheduling'
- Island-based resource management necessary when scaling to multi-domain platforms; opens up many challenging research problems

Elastic Platforms: Hardware -> Applications

Research Overview



GT `GreenIT' Private Cloud
(with VMware, IBM, Travelport, Yahoo, and additional PIs in ECE and ME)

OpenCirrus / ISTC
OpenStack / HPC
Specialization / Big Data:
Hadoop / Fawn / HDFS (with CMU
Automation:
Large-scale Management (with HP, WIPRO, ATT, Fujitsu, VMware, Amazon – hopefully)

Cloud@Home (with Motorola, Intel, Nokia) – Embedded ISTC

Service Delivery (e.g., Educational Services) (joint with Ohio State)

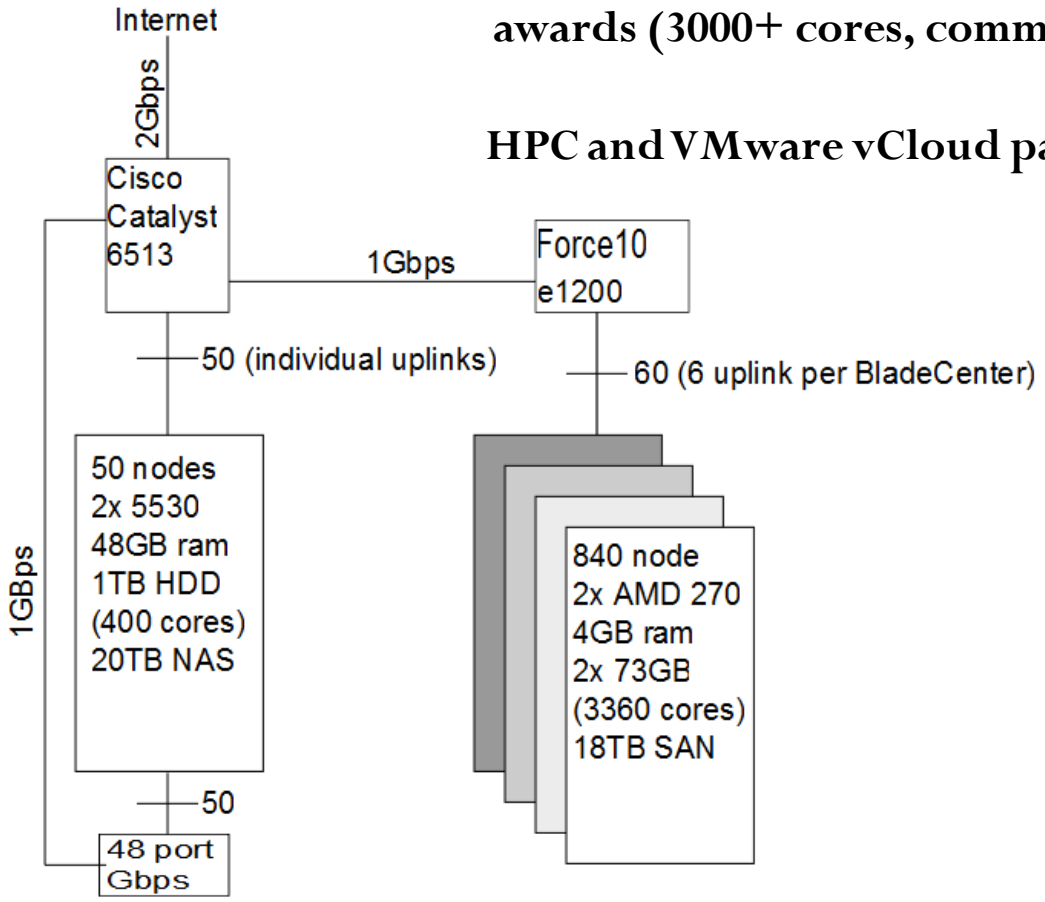
GT Facilities: OpenCirrus Cluster

`NextGen' Servers – INTEL donation (=> 1000 cores/IB interconnect)

Xen and OpenStack

`GreenIT' Private Cloud – Georgia Tech and NSF awards (3000+ cores, commodity interconnect)

HPC and VMware vCloud partitions



Measurement Repository

GT Facilities: GreenIT Private Cloud

- ~3000 cores
  - VMware VCloud
  - physical partitioning
- Fully instrumented
  - RPDUs; sensors for temperature, air velocity; fan speeds; …
  - software tools for per-core, per-…, => per-VM resource usage
- Dynamic controls on cooling infrastructure
- Instrumentation: (1) online data capture and analysis via Flume plus (2) data aggregation into HBase store for postmortem analysis

Ada Gavrilovska, Yogendra Joshi (ME) – VM Power Metering, …

Dynamically Elastic Software

Automation and Specialization (i.e., online management) require runtime models – roadblocks:

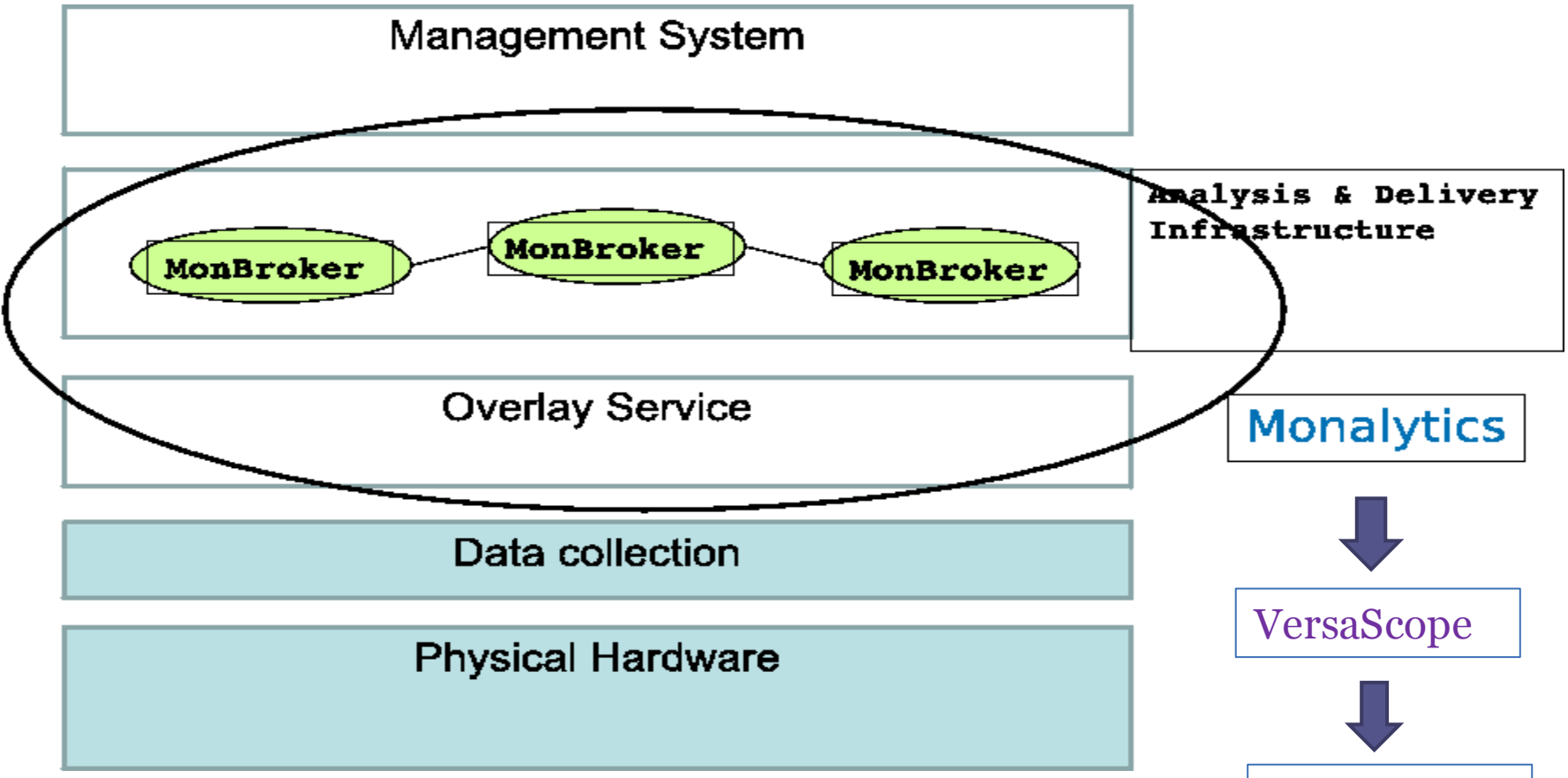
- Guest VMs run on 'virtual hardware' (i.e., shared infrastructure):
  - can lead to unexpected system behavior; e.g., network congestion control with virtual network device
- Administrators have no or insufficient knowledge about applications:
  - requires `black box' methods for recognizing application behaviors and workloads (may be unreliable)
  - requires experiment-driven and probabilistic models linking application/workload behaviors to resource consumption ; e.g., online VM power metering; multi-tier performance and failure dependencies
- Systems have management silos
  - HP's iLO, IBM's Director, IPMI and platform level management, IBM's Tivoli enterprise-level management, VMware's Virtual Center
- 'Scale': #machines, #managed entities, #levels of abstraction, #management domains
- Generalizing from 'big data' codes => multi-tier enterprise applications

=> Need for new approaches to systems monitoring, application and workload modeling, runtime analysis, and management

Greg Eisenhauer, Ada Gavrilovska, Ling Liu, Calton Pu, Karsten Schwan, Matt Wolf

Elastic Software: Automation/Specialization

Scalable Management for Cloud Systems and Applications - I



1. **Monalytics I** – Mahendra Kutare, Greg Eisenhauer, Chengwei Wang, Karsten Schwan, Vanish Talwar, ICAC 2010.
2. **Monalytics II** - ChengweiWang, Karsten Schwan, Vanish Talwar, Greg Eisenhauer, Liting Hu, Matthew Wolf, ICAC 2011.
3. **Anomaly Detection** -- Chengwei Wang, … , IM 2011.
4. **Monalytics III** – Chengwei – in submission.

Scalable Monitoring (Management) - II

State Monitoring in Data Centers: Ling Liu

Results to date: Accuracy/Efficiency:

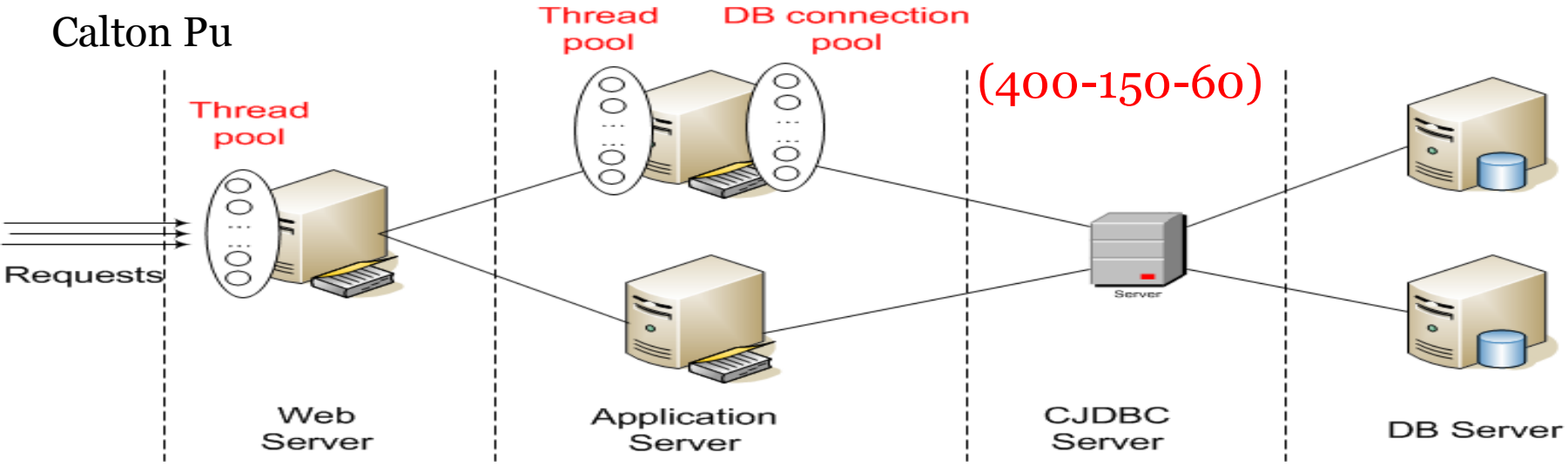
- Temporal Window based State Monitoring (icde2008, tkde2010)
- Multi-tenancy Enabled State Monitoring (ICDCS 2009, TPDS-sub)
- Violation Likelihood Based State Monitoring (icdcs12-sub)
- Fault Resilient State Monitoring (DSN12-sub)
- State Monitoring as a Service (MaaS) (TOIT-sub)

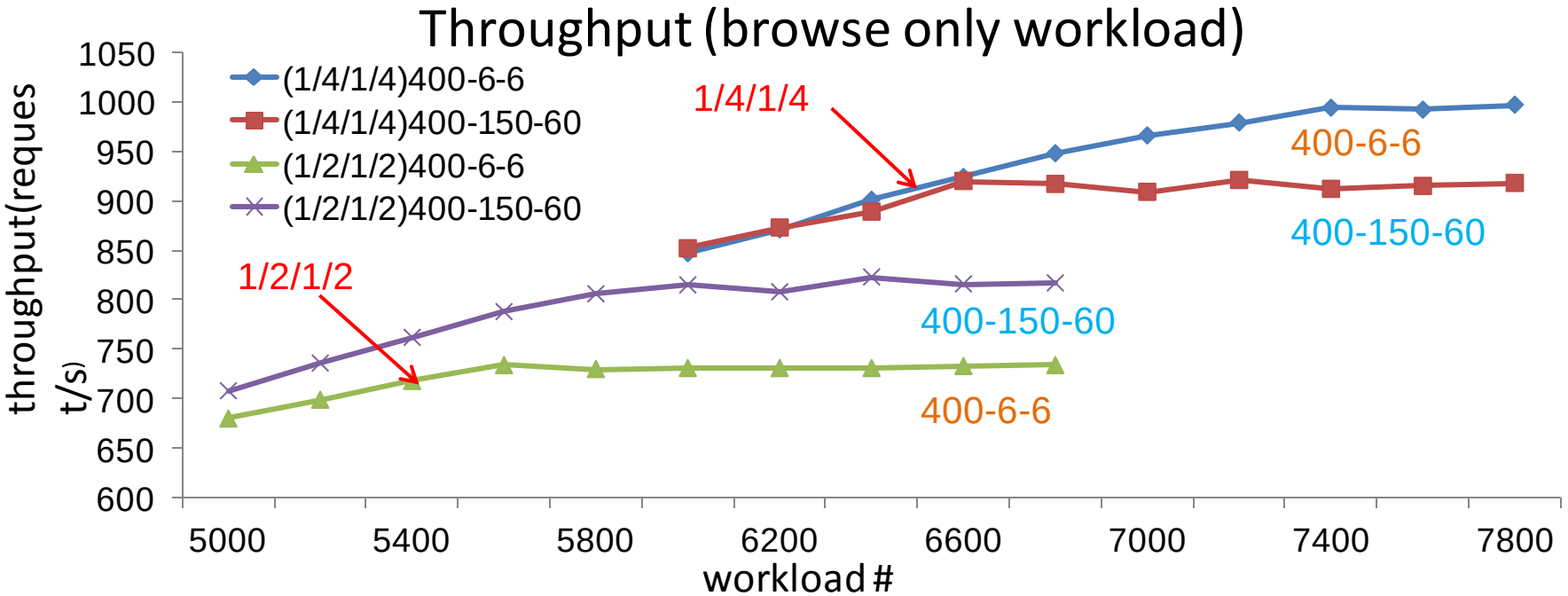
- Self-Scaling State Management (middleware'10, TOIT-sub)

- State Monitoring Enabled Cloud Deployment (USENIX12-sub)

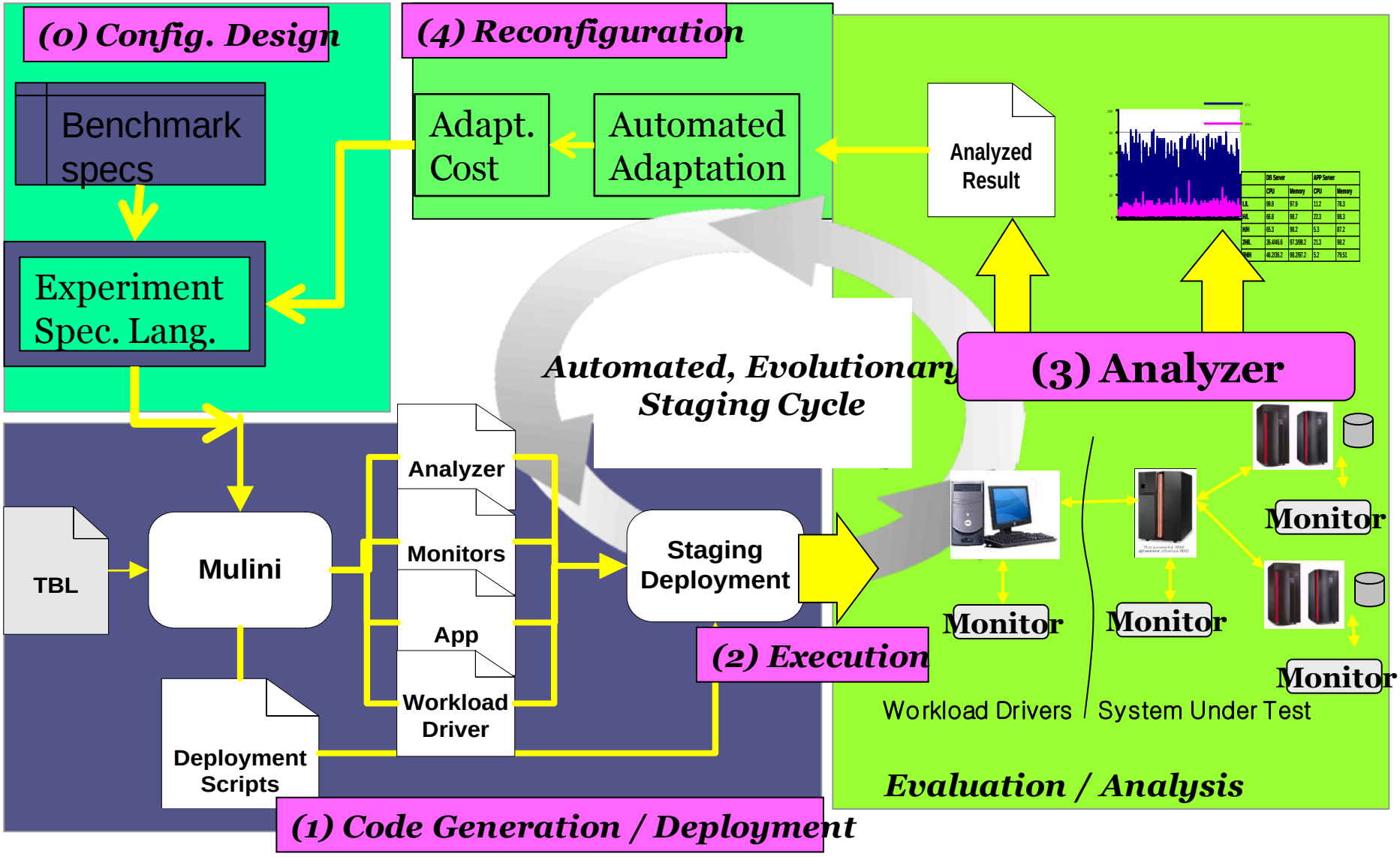
Measurements with RUBBoS: Soft Resource Over-Allocation

Calton Pu





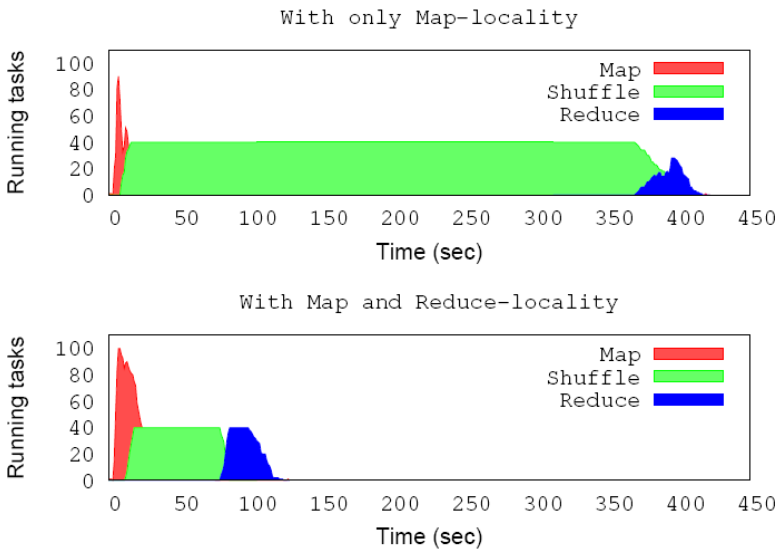
The ELBA System for Automated Configuration Management



Big Data I: Optimizing MapReduce Workloads in Clouds

Ling Liu

- Workload aware, data locality aware job scheduling
  - Load awareness in data placement
  - Job-specific locality-awareness
  - Job-specific data replication
- Approach
  - To optimize for data locality during both map and reduce phases
  - Benefits from both intelligent locality-aware data placement as well as VM placement
  - Achieves up to 50% reduction in execution time and 70% reduction in cross-rack network traffic
- Next:
  - Workload aware, data locality aware and cost-aware VM management
  - Enhancing MapReduce for Big Data analytics



Karsten Schwan

- Resource-constrained data intensive systems:
  - 'Hadoop' counterpart of resource constrained Key-Value Store (Fawn++) – with Intel Pittsburgh (Michael Kaminsky, Dave Anderson)

Big Data II: Data Hotspots

- Problem: Data Hotspots in HDFS – multiple applications using same HDFS store
- Solution: (1) Runtime methods for automated hotspot detection, followed by (2) Selective dataset replication (building on Rabbit, earlier work with Greg Ganger,…)

Conclusions

- Elasticity is needed at and must operate across many (or all) levels of the hardware => application stack; thus requires cross-stack support and interaction

- Automation requires:
  - Scalable, Accurate Monitoring and Detection
  - Runtime Modeling through Experimentation
  - Continuous Management and Runtime Specialization

- Many remaining challenges:
  - managing and modeling 'over time' (simple example: VM migration)
  - dealing with state space explosion for measurement-based management
  - dynamic change in management purpose and different/multiple simultaneous management goals
  - managing across multiple resources and resource types
  - how to exploit cross-stack linkages and interactions

- **ISTC engagement sought across the entire stack**
  (we also interact with Embedded ISTC)