THE UNIVERSITY OF ARIZONA®

Autonomic Comosable Data Center (ACDC): The Next Generation Paradigm for Developing Large Scale Data Centers

Salim Hariri, Director
NSF Center for Cloud and Autonomic Computing
The University of Arizona
nsfcac.arizona.edu
email: hariri@ece.arizona.edu

Cloud and Autonomic Computing Center

Autonomic Computing Lab

Presentation Outline

- Brief Overview of Ongoing CAC Research Activities
- Motivation – Why Composable Data Centers?
- What are challenges of Designing Composable Data Centers
- UA Approach to Build a Composable System:
  - Just iIn Time Architecture (JIA)
  - Prelinary Analysis and Evluation
- Conclusions

On Going UA CAC Projects

- Autonomic Cyber Security (ACS)
  - Tactical Cyber Immune System (TCIS)
  - Autonomic Monitoring, Analysis and Protection (AMAP)
  - Anomaly based Detection of Attacks on Wireless Ad Hoc Networks
  - Resilient Cloud Services
  - Hacker Web: Securing Cyber Space: Understanding the Cyber Attackers and Attacks via Social Media Analytics
  - IoT Security Framework
- Big Data Analytics
  - Big Data Cybersecurity
  - High Performance Machine Learning Framework (HPMLF)
  - Heart Modeling, Analysis, Diagnosis and Prediction
- High Performance Distributed Computing and Applications
  - Just-In-Time Architecture (JITA) for Composable High Performance Data Centers
  - Heart Cyber Expert System (HeartCyPert)
  - Oil Well Data Analytics and Protection (OWDAP)
  - Hurricane Continuous Modeling and Simulation Environment

Credit to

Dr. Chung-Sheng Li
IEEE Fellow &
IBM Academy of Technology Leadership Team
Director, Commercial Systems
IBM Research Division

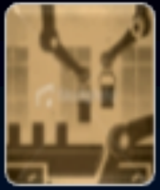
Cloud evolution – systems point of view

Homogeneous, Virtualized

Cloud 1.0

Software Defined Environments

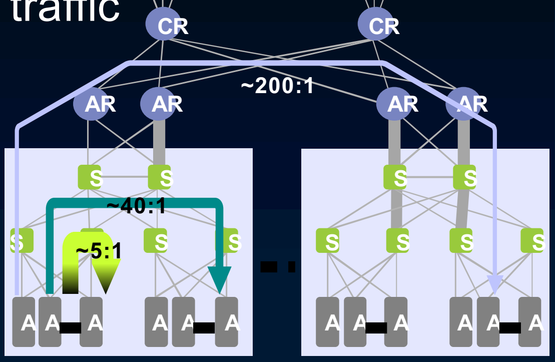
Cloud 2.0



Composable Systems

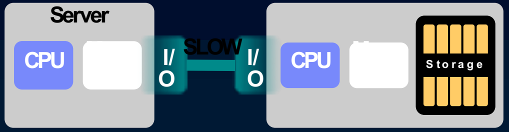
Cloud 3.0

Systems of Insight workloads create high east-west datacenter traffic



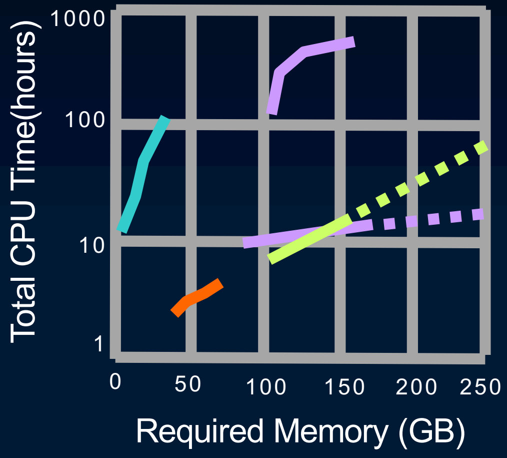
Systems of Insight workloads often require large, low latency storage

- Remotely attached storage incur long latency and throughput bottleneck



- Locally attached SSD & storage could be inflexible and expensive

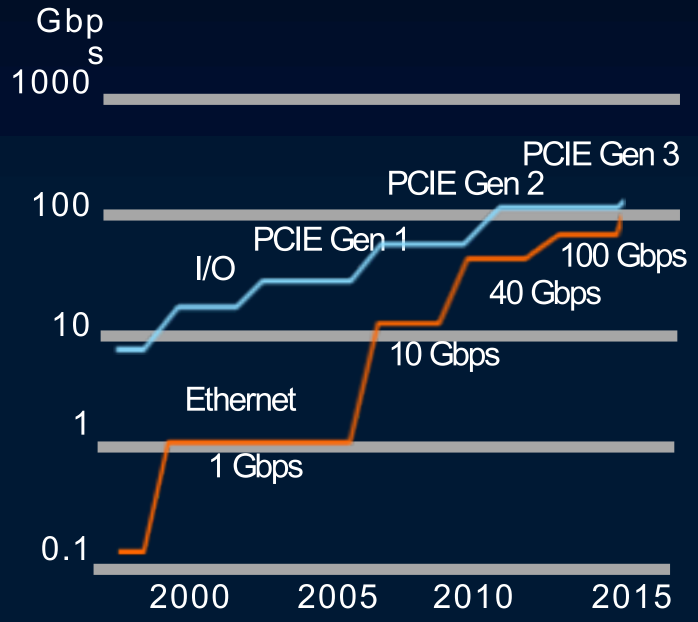
Systems of Insight workloads often have wide spectrum of memory requirements



Composable systems take advantage of rapid progress on network speed and acceleration

High bandwidth network and interconnect speed is expected to be comparable to PCIe speed by 2015-2017

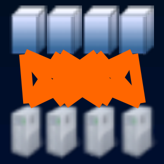
Network compared with System I/O



Increased focus on east-west traffic accelerate adoption of 2-tier (spine-leaf) and 1-tier DCN architectures

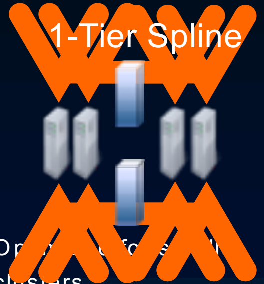
Network Design Choices

2-Tier Leaf-Spine



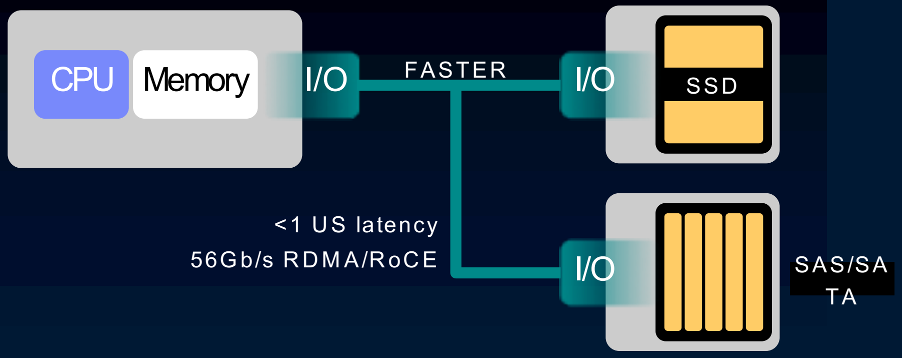
- Optimized for Scale & Growth – Cloud Model
- One network for all Apps / Tenants
- All nodes are equi-distant: 3-hops

1-Tier Spline



- Optimized for small clusters
- One network per Application
- All nodes are directly connected: 1 Hop

High speed network enables storage disaggregation with zero penalty to performance



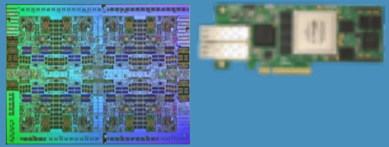
<1 US latency

56Gb/s RDMA/RoCE

Innovation platform: Agile, composable, disaggregated, heterogeneous, cloud-scale

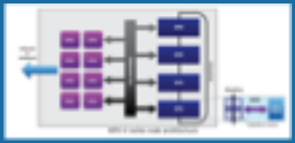
Enabled by significant reduction in cost of bandwidth and virtualization advances.

P8-CAPI (coherent insertion of accelerators)



GPU (Genomics, Healthcare)

TMS SSD (FSS, IoT)



Maxeler FPGA Accelerator (FSS, Natural Resources)

Active Storage (hyperconverged) Node

DRAM	P7/P8 CPU	AS Net
10GbE	FPGA	SAS/ SATA
Flash, MRAM, PCM		

JBO

Datacenter Scale “Computer”

Self tuned & Self Optimized

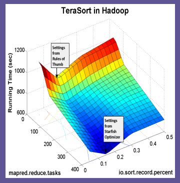
Software Defined Infrastructure Resource Abstractions for Composable Systems

High BW, Low Latency Network and Interconnect

Hyper-converged / Disaggregated Components

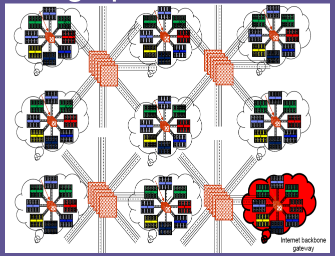
Building Blocks for Composable System

Self-tuning could achieve 75% of optimal performance within minutes



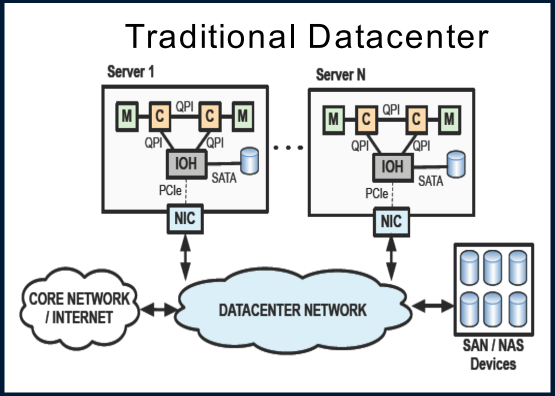
Disaggregated fully non-blocking spine-leaf data center network based on SDN is available now (2014)

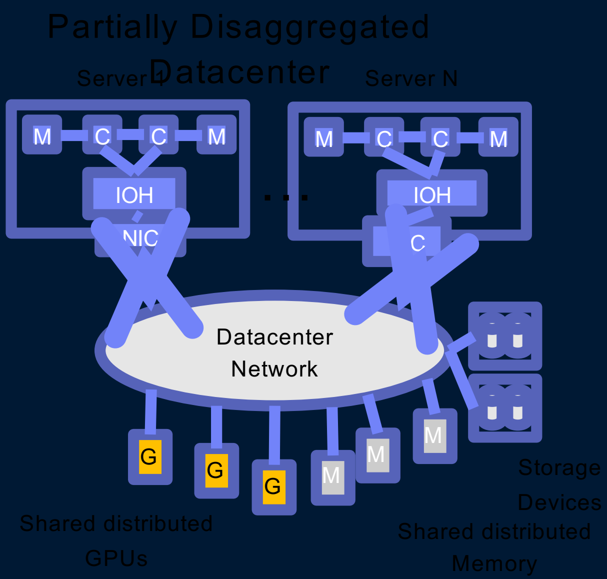
High bandwidth Si Photonics links for east–west direct connections rewired using optical switches

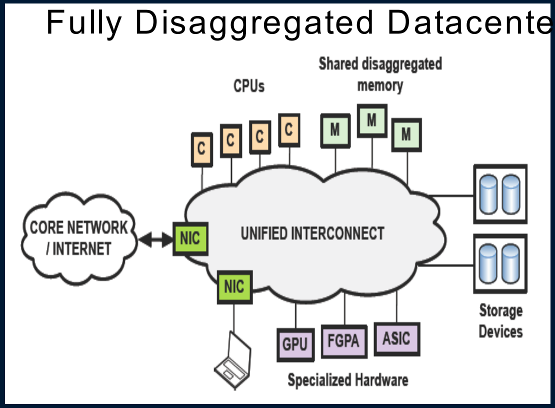


Why Disaggregation?
Resource Modularity

- Easier to build & evolve
  - Resources have different cycles/trends/constraints.
  - Disaggregation enables independent evolution, the biggest driving force from vendor's viewpoint
- Fine-grained resource provisioning
  - Current practice: replace/buy an entire server, rack, or even datacenter.
  - Go buy some CPU blades at Best Buy® and plug them in.
- Operational efficiency
  - Datacenter as a single giant computer
  - Higher utilization with statistical multiplexing
- Reduces the need to optimize for "locality" of data to processing and hence lessens the need for careful placement of data & workload
- **Physical resource pooling**: allows ***fail in place*** and reduce/lessen the need for field maintenance (especially when coupled with software defined everything)


Traditional Datacenter

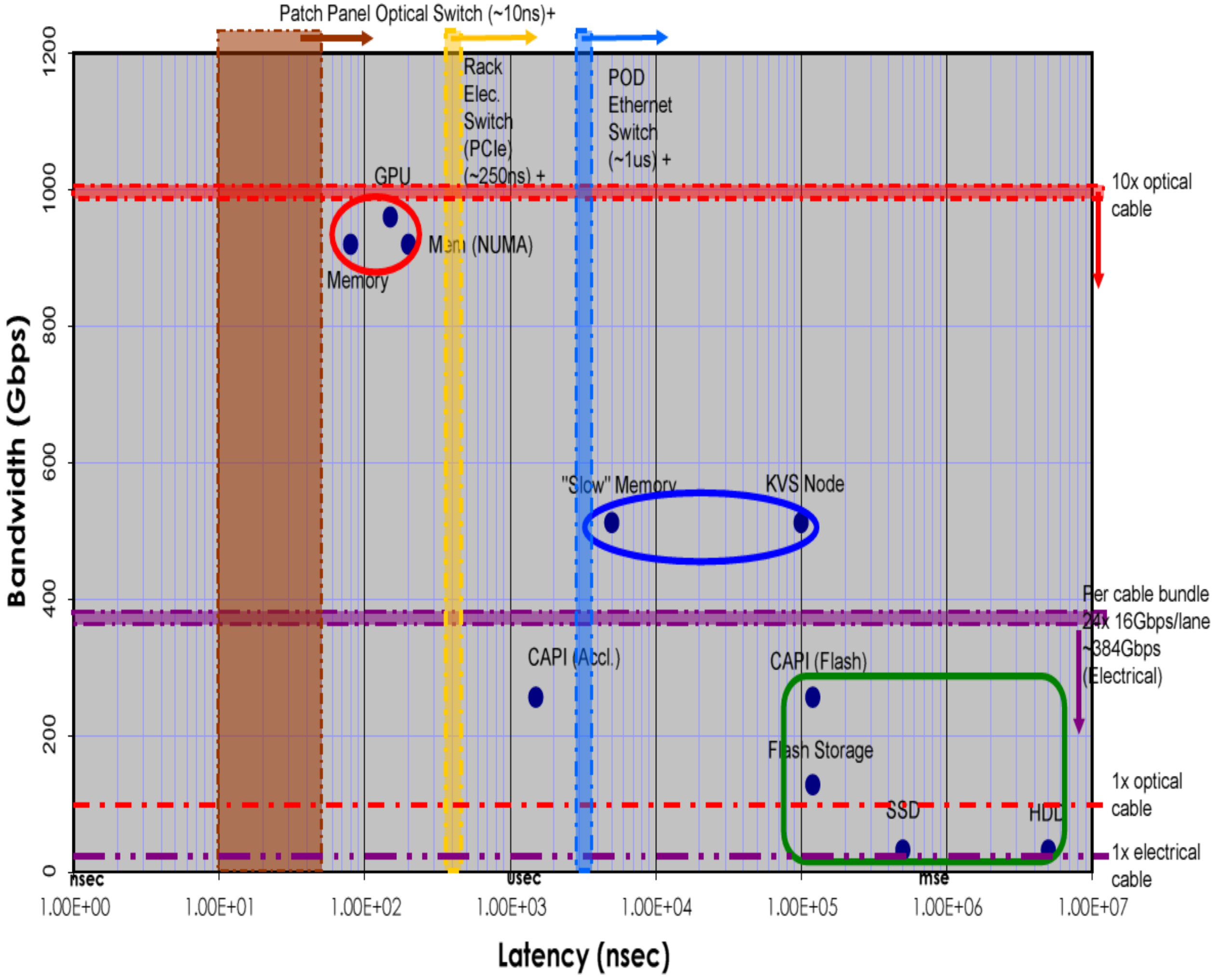

Partially Disaggregated Datacenter

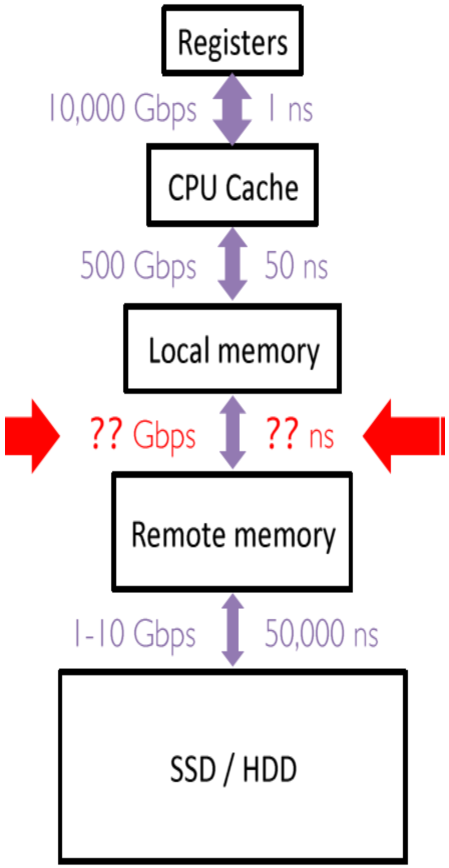

Fully Disaggregated Datacenter

What are the challenges?

- **Network**: How fast should the network be? How much latency could workload tolerate?
- **Scalability**: What is the right (sweet spot) scale of the disaggregation? (chassis, rack, pod, datacenter)
- **Quality of Service/Resiliency:** What is the impact on the RAS? Are there new opportunities resulting from physical resource pooling?
- **Circuit switching vs. Packet Switching:** Can we leverage optical circuit switching (OCS)?
- **Unified control plane/scheduler:** How can we make sure the scheduling and placement of workload do not create conflicting data flow within the network due to disaggregation?

What are the appropriate interconnect technologies for disaggregation?





Registers

10,000 Gbps 1 ns

CPU Cache

500 Gbps 50 ns

Local memory

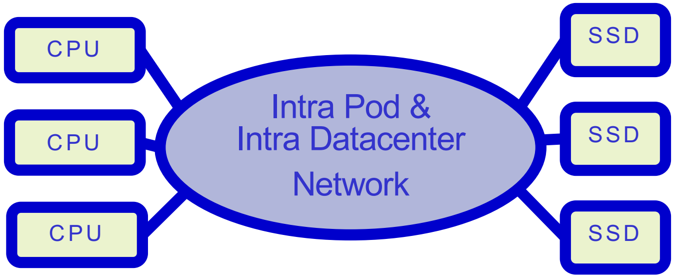
?? Gbps ?? ns

Remote memory

1-10 Gbps 50,000 ns

SSD / HDD

Amin Vahdat (Google) in his keynote at 2014 Open Network Summit presented the case that the cross-sectional BW needs to be 100+ Tb/s and end-to-end latency < 10 us to support disaggregated SSD and large MapReduce workloads

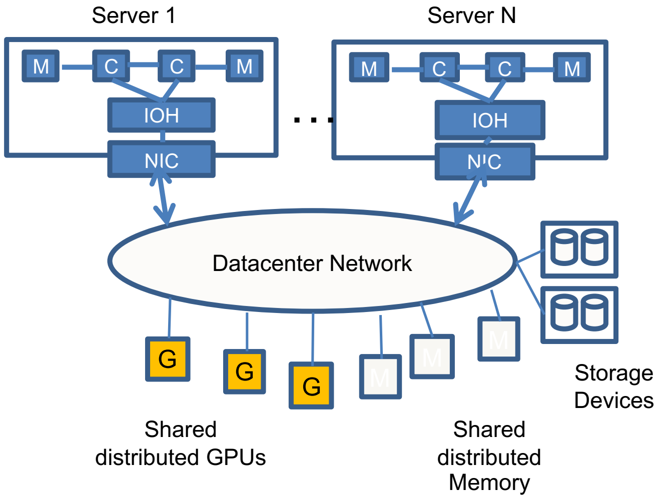


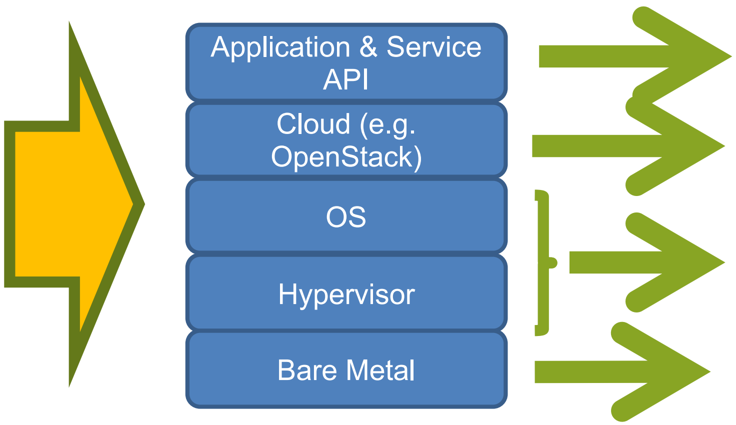
- Amdahl’s rule of thumb: every MHz of CPU needs to pair with 1 Mb/s of I/O
  - 16 core @2.5GHz ➔ 40 Gb/s
  - 32 core @2.5GHz ➔ 80 Gb/s
  - SSD: 100K+ IOPS, 100 us access latency (cf. HDD: 50 IOPS, 10ms access latency)
- Implications: 1000 VMs require 40 Tb/s bisection, 10 us access latency (port to port)
- MapReduce/Hadoop and large graph implementations within BigData, Analytics, and NoSQL generate large volume of east-west traffic among Hadoop clusters
- Cross-sectional BW: Azure Pb/s, GCE 100 Tb/s

Network requirements: Cross-sectional BW: 100+ Tb/s, end-to-end latency < 10 us

Integration Methodology for Disaggregated Physical Resource in the system Stack

Partially Disaggregated Datacenter





RAMCloud

Memory as a Service, Flash as a Service

Transparent to app (e.g. Swap RamDisk, RDMA)

Transparent to app/OS/hypervisor

- **Hardware based**, transparent to applications and OS/hypervisor
  - Access as an I/O device based on direct integration through PCIe over Ethernet
  - Global shared memory for disaggregated memory
  - Direct attached memory through Centaur (Power), CAPI (Power), and QPI (Intel)
- **Hypervisor/container based**: transparent to applications and guest OS
  - getMemory: e.g. remote swap RamDisk
  - getGPU: e.g. through PCIe over Ethernet
- **Microservice/Application based:** expose disaggregation details and resource remoteness directly to applications
  - Resources exposed via high-level APIs (e.g. put/get for memory) using built-in processing element
    - **GetMemory** (e.g. Memory as a Service) as one of the OpenStack service
      - Openstack service sets up channel between host and memory pool service over RDMA.
    - **GetGPU** instance
      - Locate available GPU from GPU pool & host from host pool
      - Establish channel between host and GPU through RDMA/PCIe and expose to applications via library or virtual device.
  - Cloud-born applications already built using such APIs

UA Approach to Develop Composable Datacenters: JITA – Just in Time Architecture

Collaborators

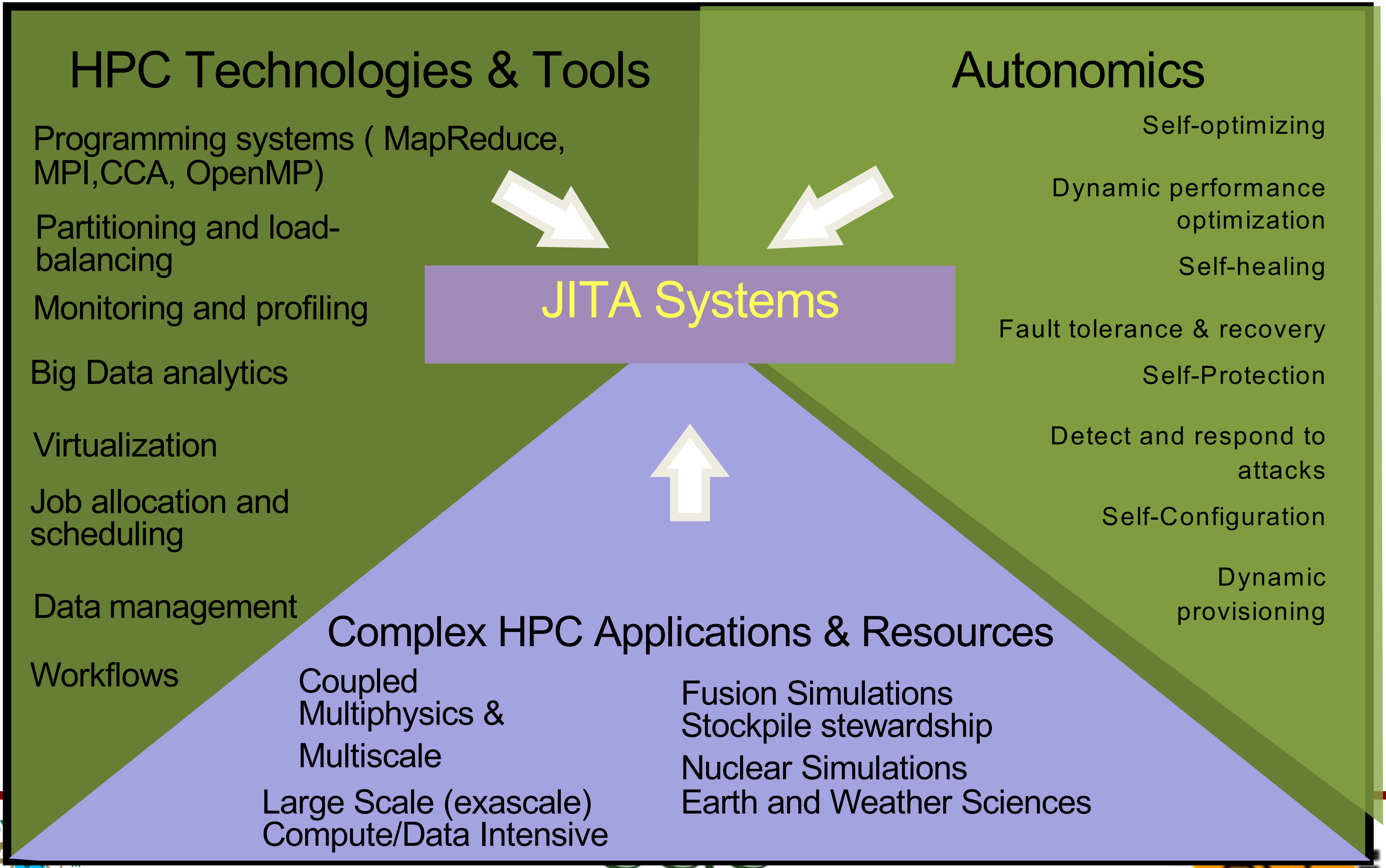
UA: Ali Akoglu, Ivan Djordjevic, and Cihan Tunc

Colorado State University: H. J. Siegel

Research Issues

- How to build disaggregated or composable data centers on the fly?
- How to develop software architecture and resource management that can be customized dynamically to meet application SLO?
  - Virtual Data Center (VDC)
- How to leverage emerging optical interconnect technologies?
- How to model and validate the performance of composable data centers?

JITA Technologies and Tools



HPC Technologies & Tools

Programming systems (MapReduce, MPI,CCA, OpenMP)

Partitioning and load-balancing

Monitoring and profiling

Big Data analytics

Virtualization

Job allocation and scheduling

Data management

Workflows

Autonomics

Self-optimizing

Dynamic performance optimization

Self-healing

Fault tolerance & recovery

Self-Protection

Detect and respond to attacks

Self-Configuration

Dynamic provisioning

JITA Systems

Complex HPC Applications & Resources

Coupled Multiphysics & Multiscale

Large Scale (exascale) Compute/Data Intensive

Fusion Simulations
Stockpile stewardship

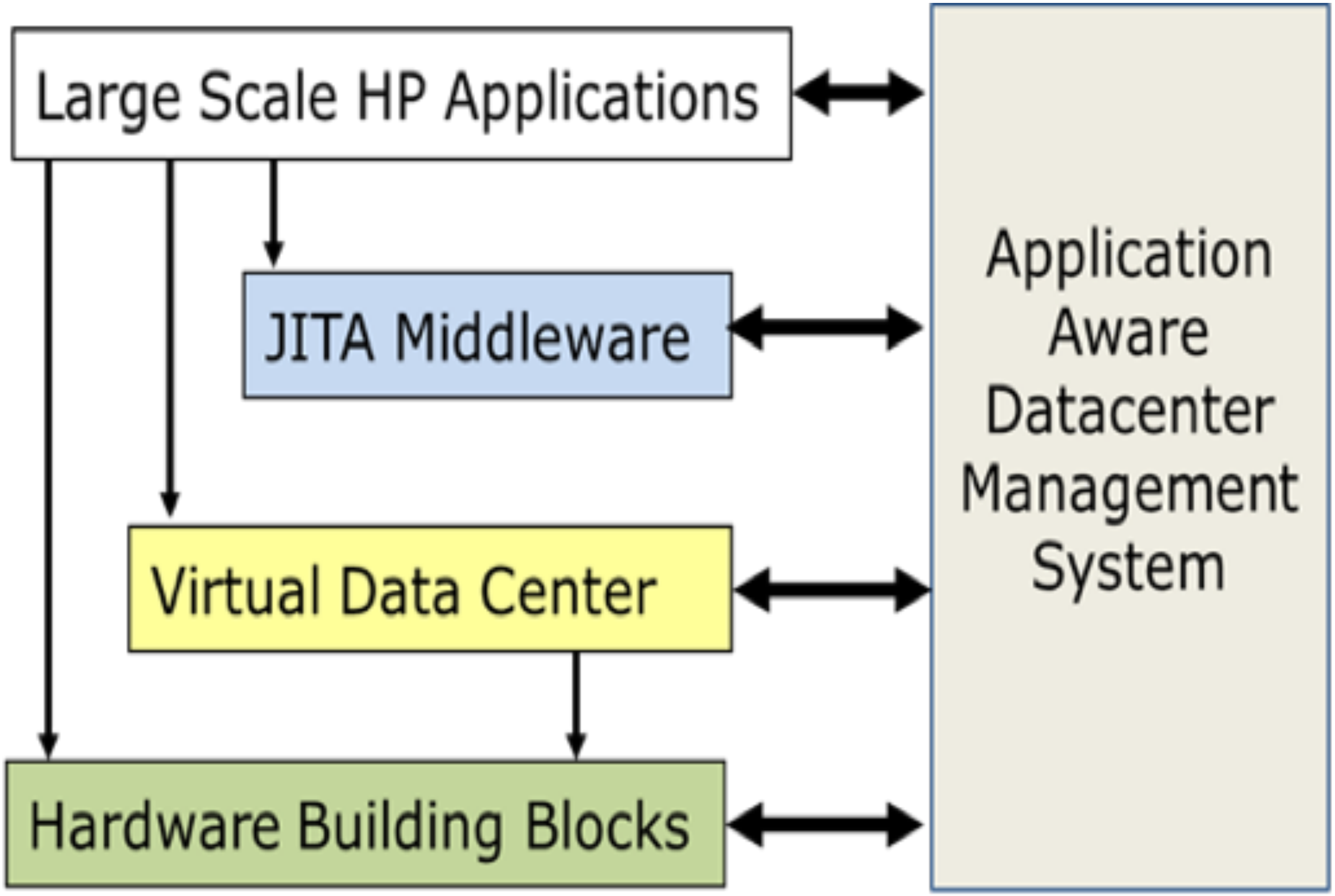
Nuclear Simulations
Earth and Weather Sciences

Research Thrusts

- Thrust 1: JITA Design Approach
- Thrust 2: Optical Interconnect Infrastructure
- Thrust 3: Modeling, Analysis, and Simulation of JITA

Thrust 1: JITA Design Approach

Just-In-Time Architecture (JITA)



Large Scale HP Applications

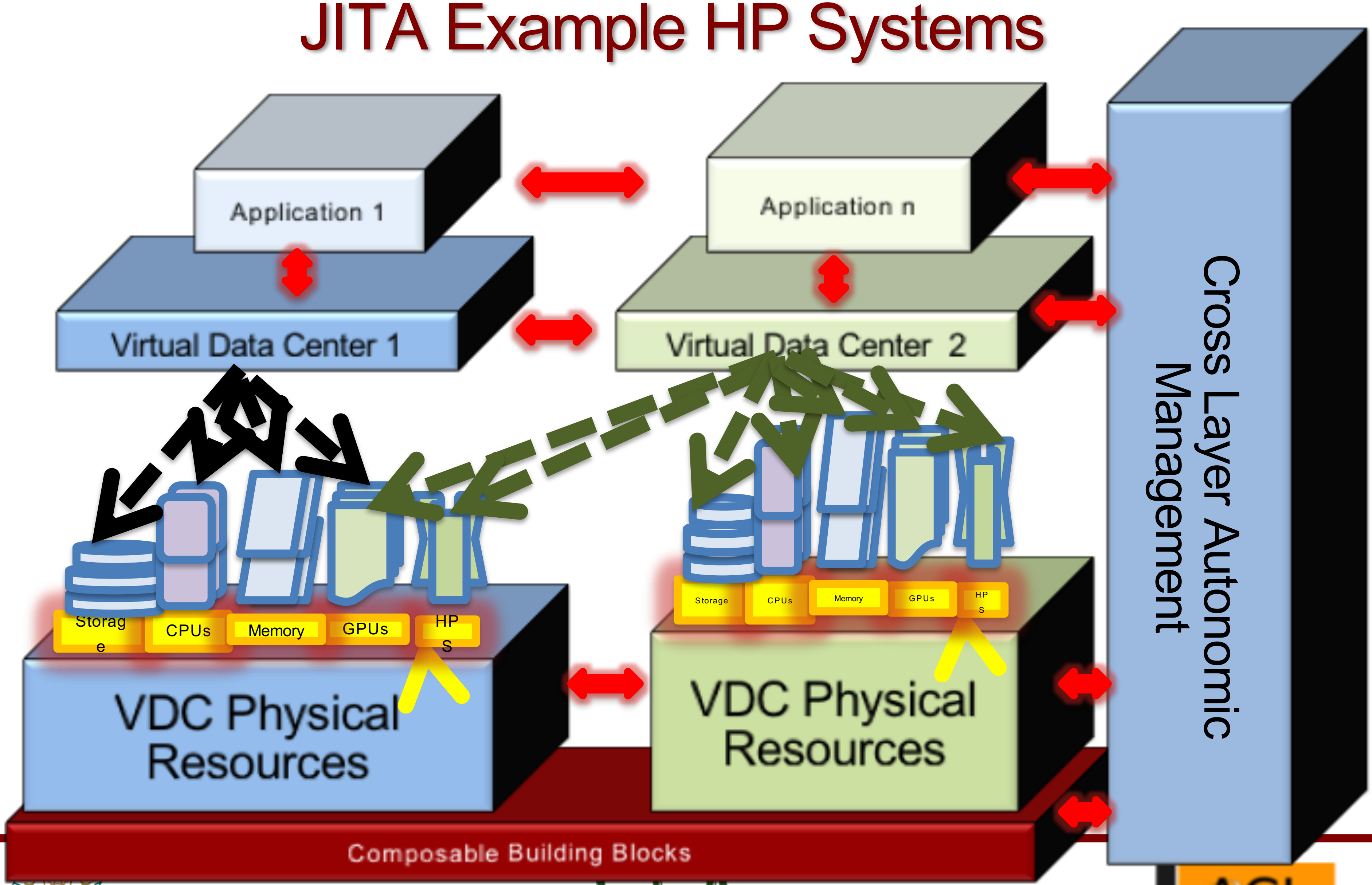
JITA Middleware

Virtual Data Center

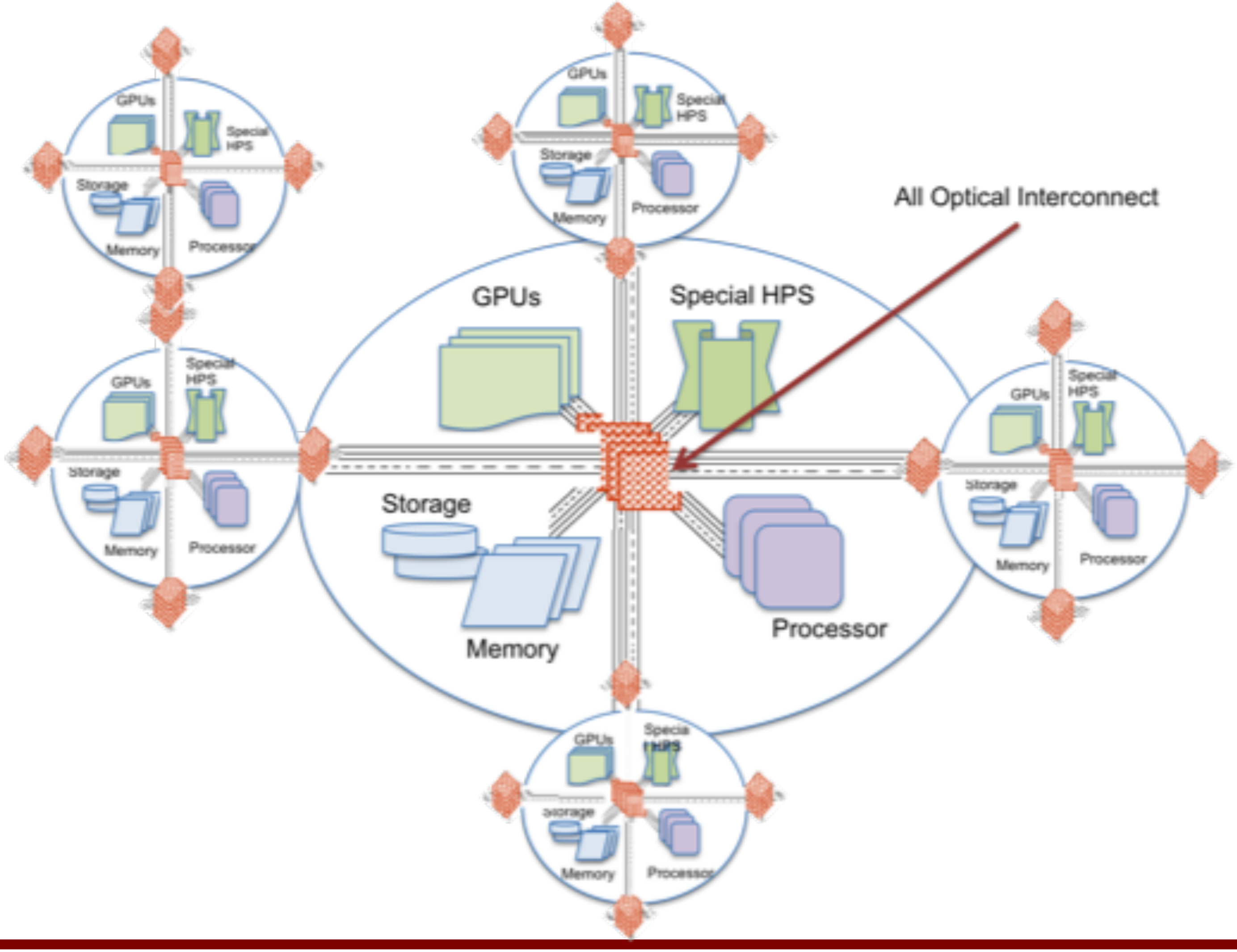
Hardware Building Blocks

Application Aware Datacenter Management System

JITA Example HP Systems

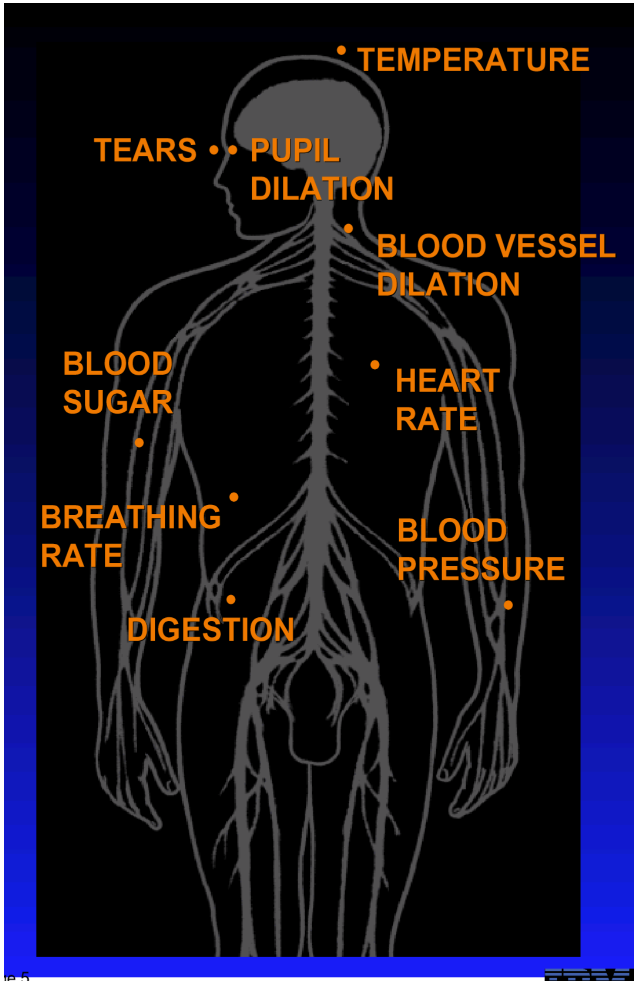


Scalable Architecture

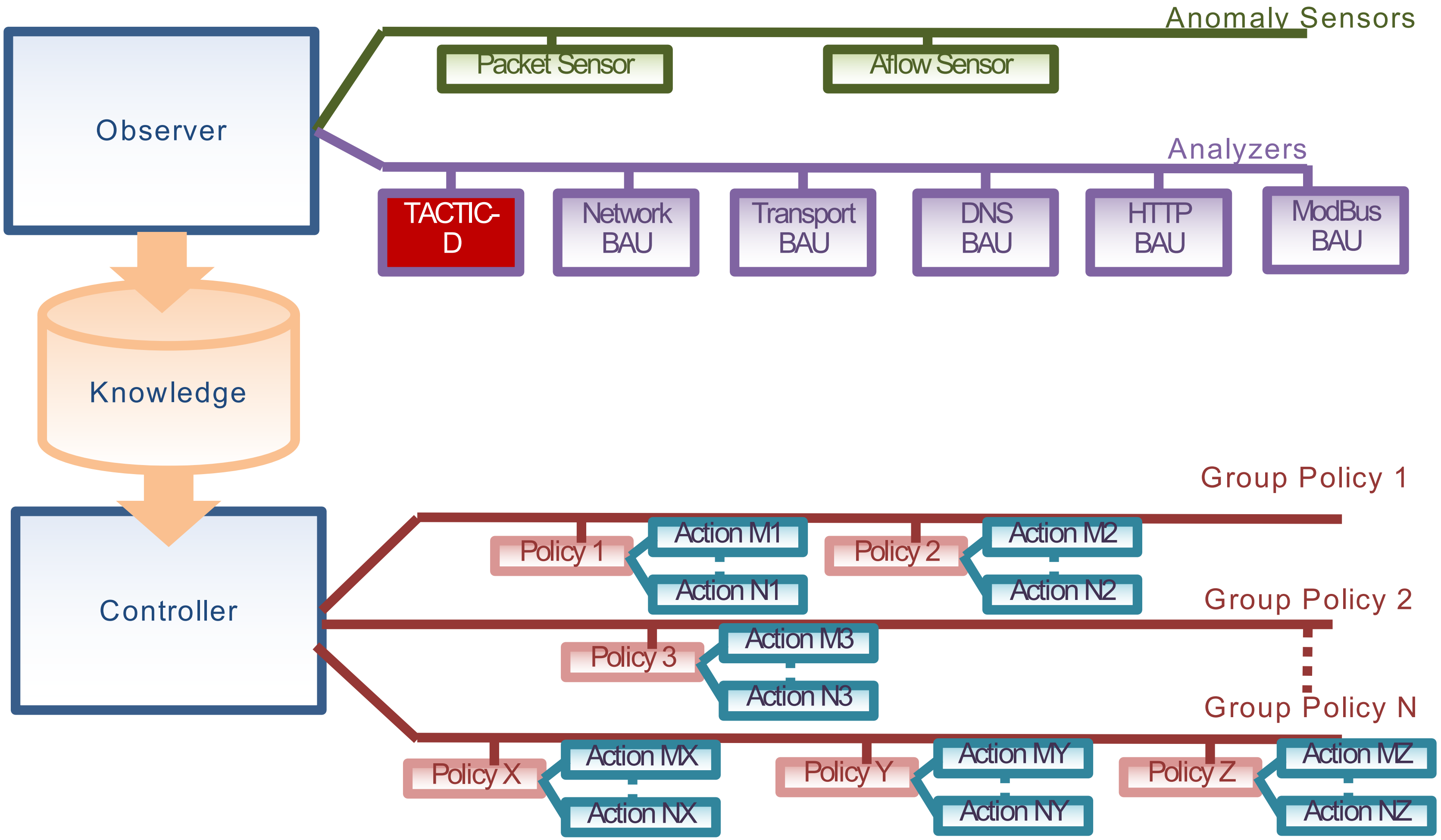


Autonomic Computing

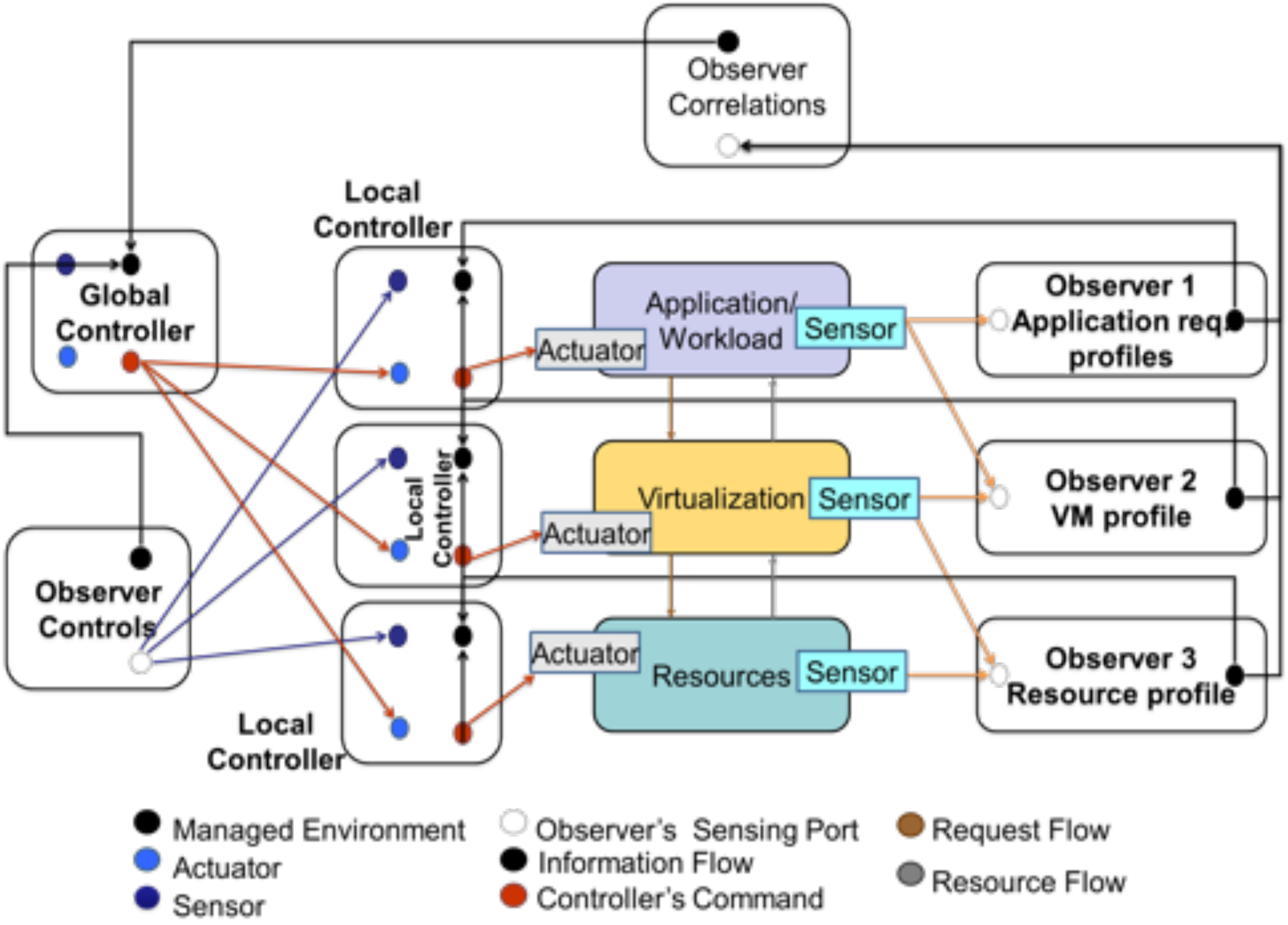
- Analogous to Human autonomic nervous system
- AC continuously monitors, analyzes, and diagnoses the managed system behavior and then takes proactive actions



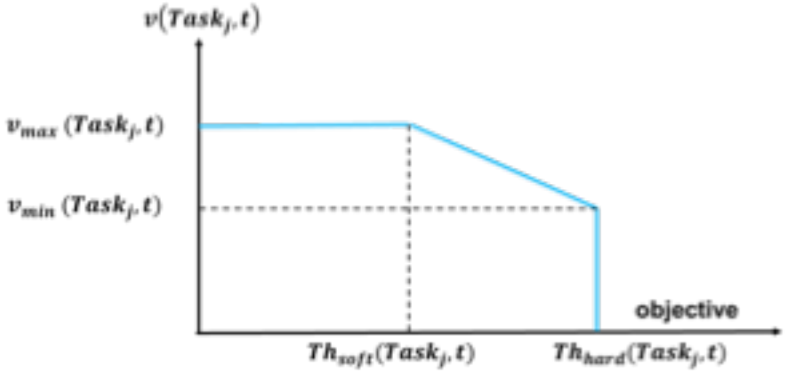
Autonomic Component Architecture



Cross-layer Autonomic Management



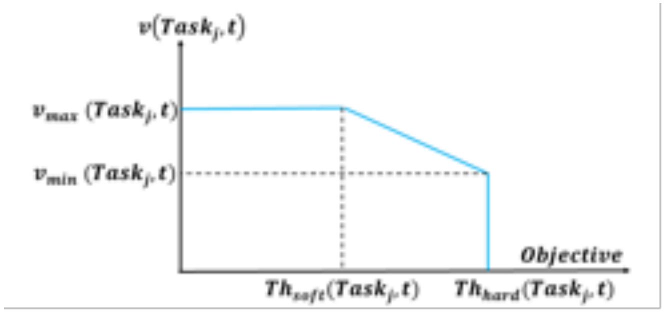
Value of Service (VoS)

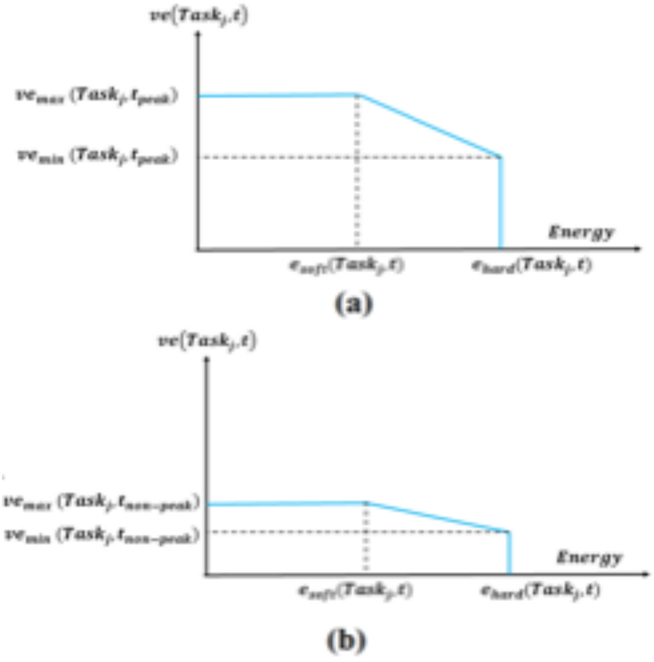


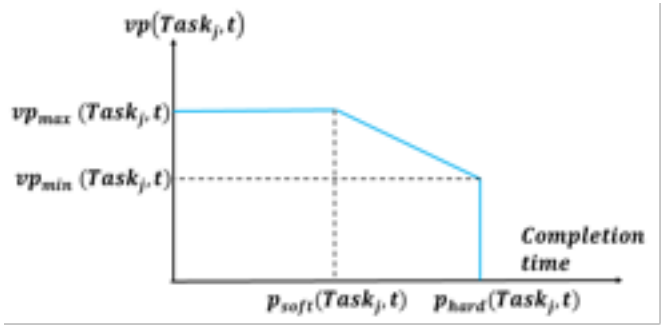
- Utility functions have been shown to be effective metrics in resource management, especially in an oversubscribed environment.
- A primary difference of our VoS metric from utility techniques is the fact that the value metric allows us to consider the value of performing resource management at a particular time of the day or night as well as the actual operational costs of using the allocated resources at a given time.

VoS Examples

- Value of Service (VoS) with respect to Performance and energy







Energy value vs energy consumed
(a) Peak time, (b) Non-peak time

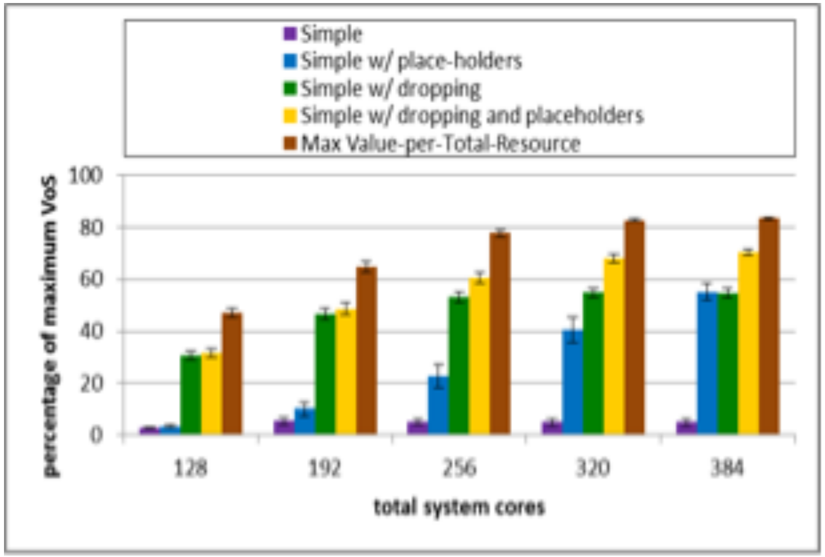
JITA Scheduling Algorithm

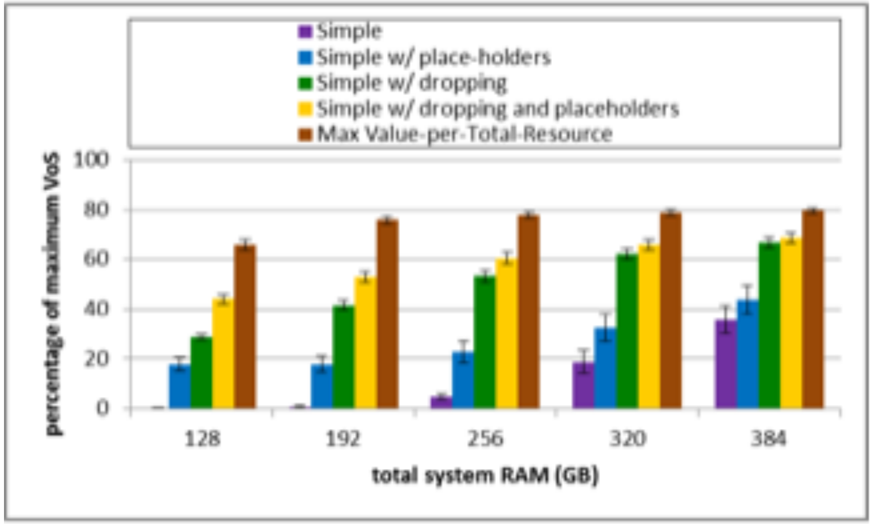
Our algorithm is based on the resource allocation choices that provide the highest task value divided by the amount of resources used, to better utilize the resources *Maximum Value-per-Total Resource (Maximum VPTR).*

Algorithm 1. Pseudo-code for the Max VPTR heuristic.

```
while the set of mappable tasks is not empty
    for each task in the set of mappable tasks
        find the allowable VM configuration maximizing task VPTR
    select task/VM pair that gives the highest VPTR
    if selected task can start execution immediately
    then
        assign selected task to VMs
    else
        create a place-holder for selected task using its resource allocation choice
    remove selected task from mappable tasks
end while
```

JITA Scheduling Simulation Results

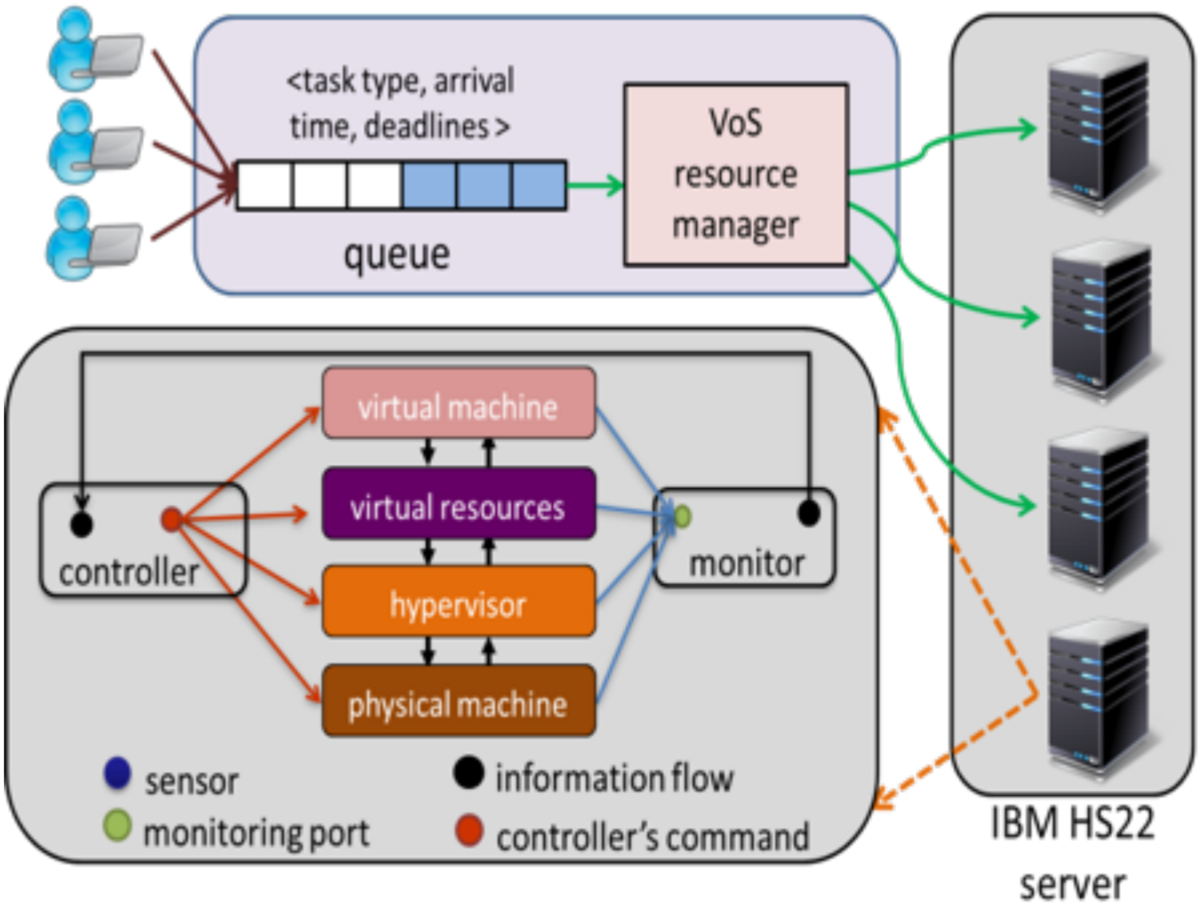




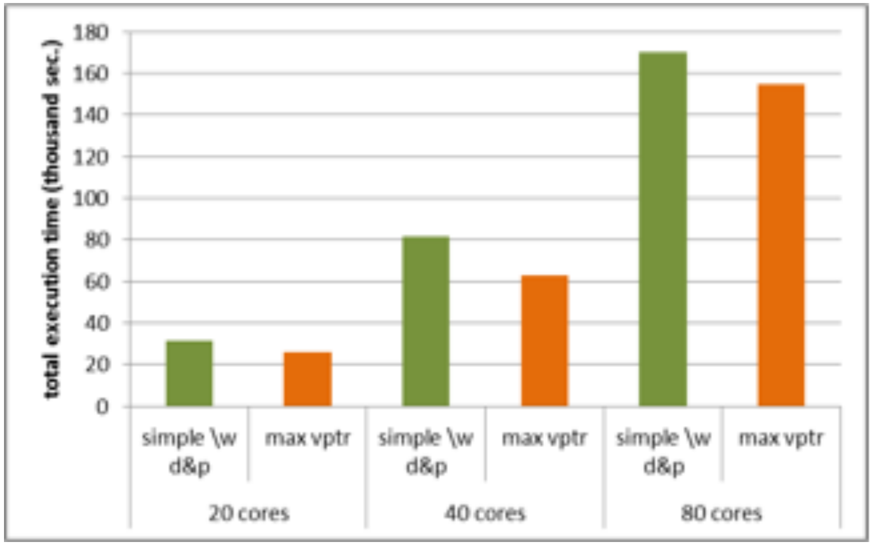
The percentage of maximum VoS earned by the heuristics in environments where the number of cores in the system is varied from 128 to 384 and the amount of memory is fixed at 256 GB.

The percentage of maximum VoS earned by the heuristics in environments where the amount of memory in the system is varied from 128 to 384 GB and the number of cores is fixed at 256.

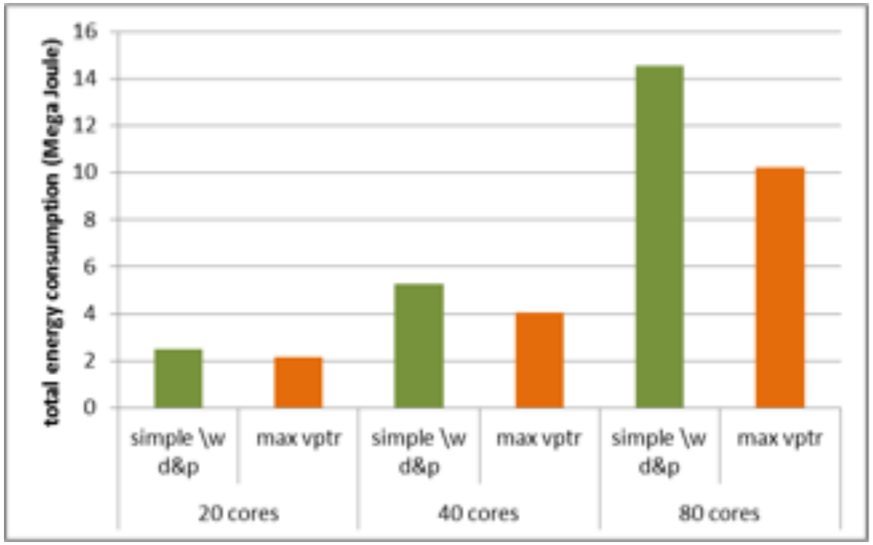
JITA Experiment Results



JITA Experimental Results



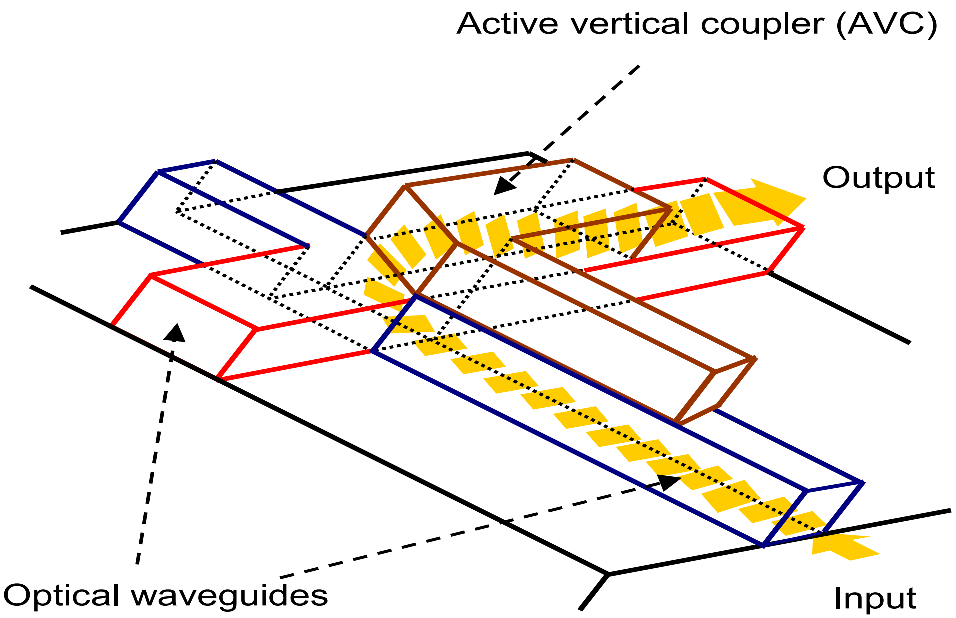
Total task execution time for workload 1 (thousand seconds).



Total energy consumption by the executed tasks for workload 1 (in Mega Joules)

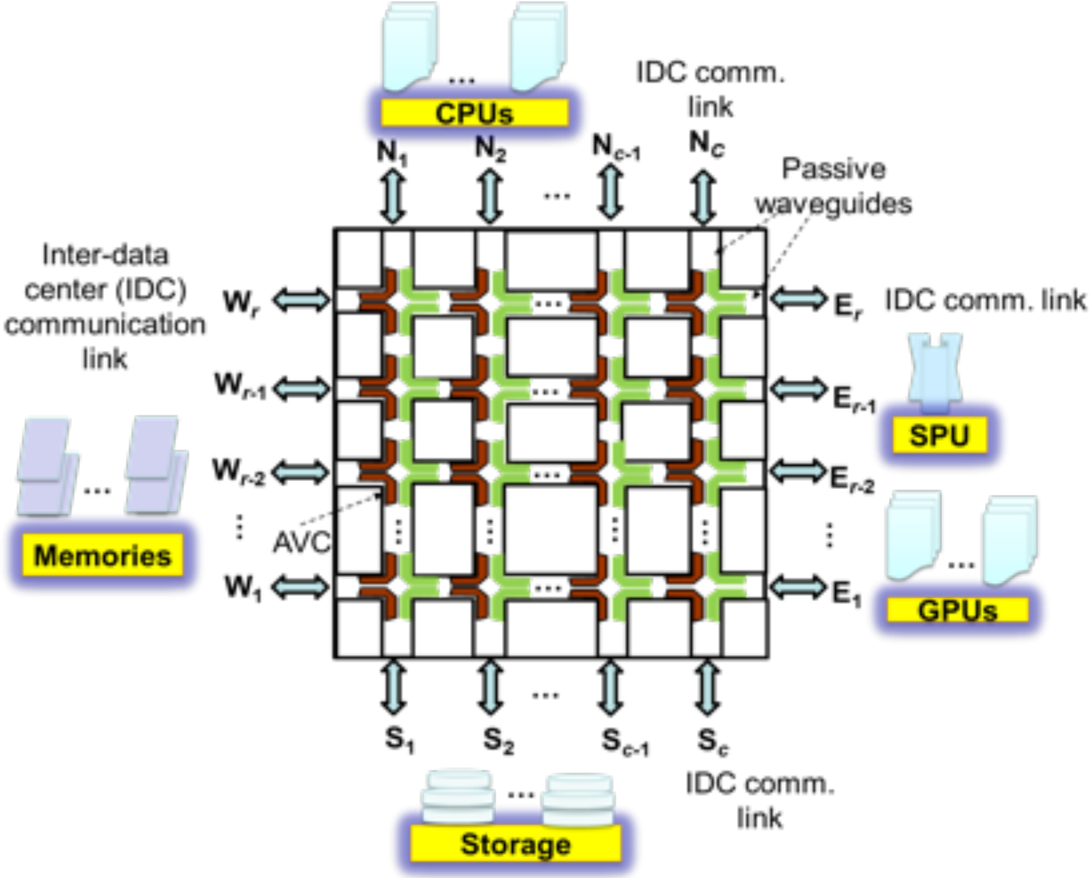
Thrust 2: Optical Infrastructure Design Approach

Optical Cell Design

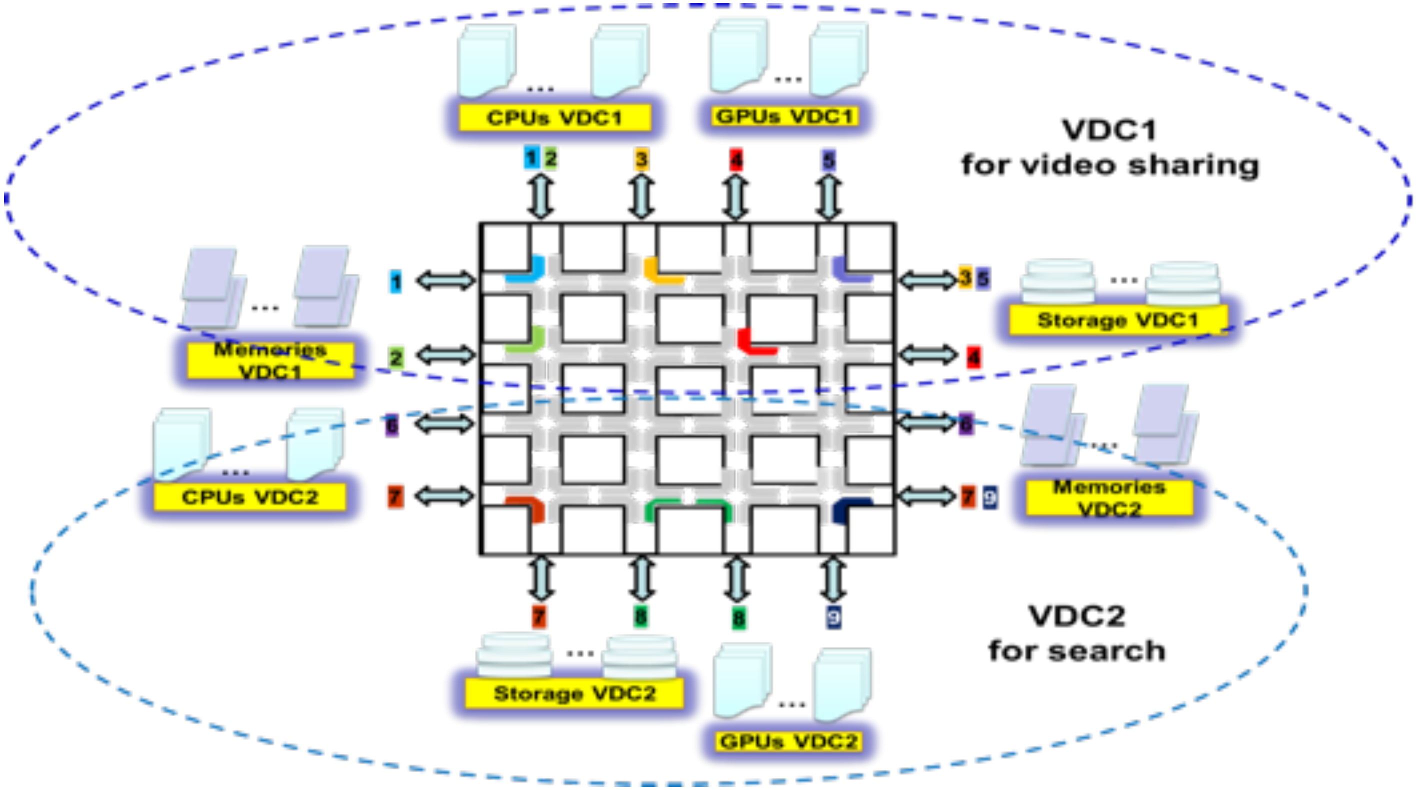


Switching cell operation principle

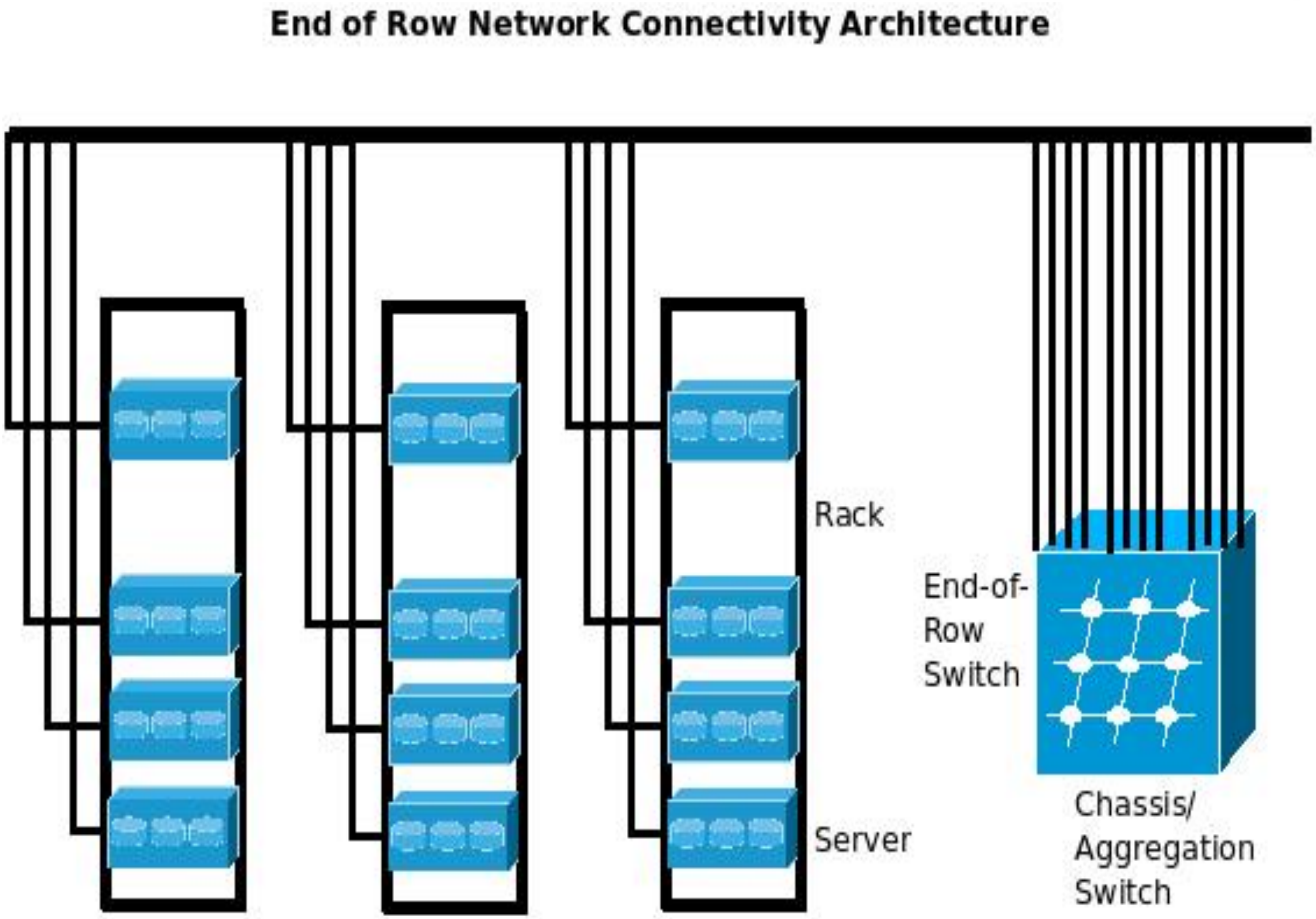
Optical Space Switch



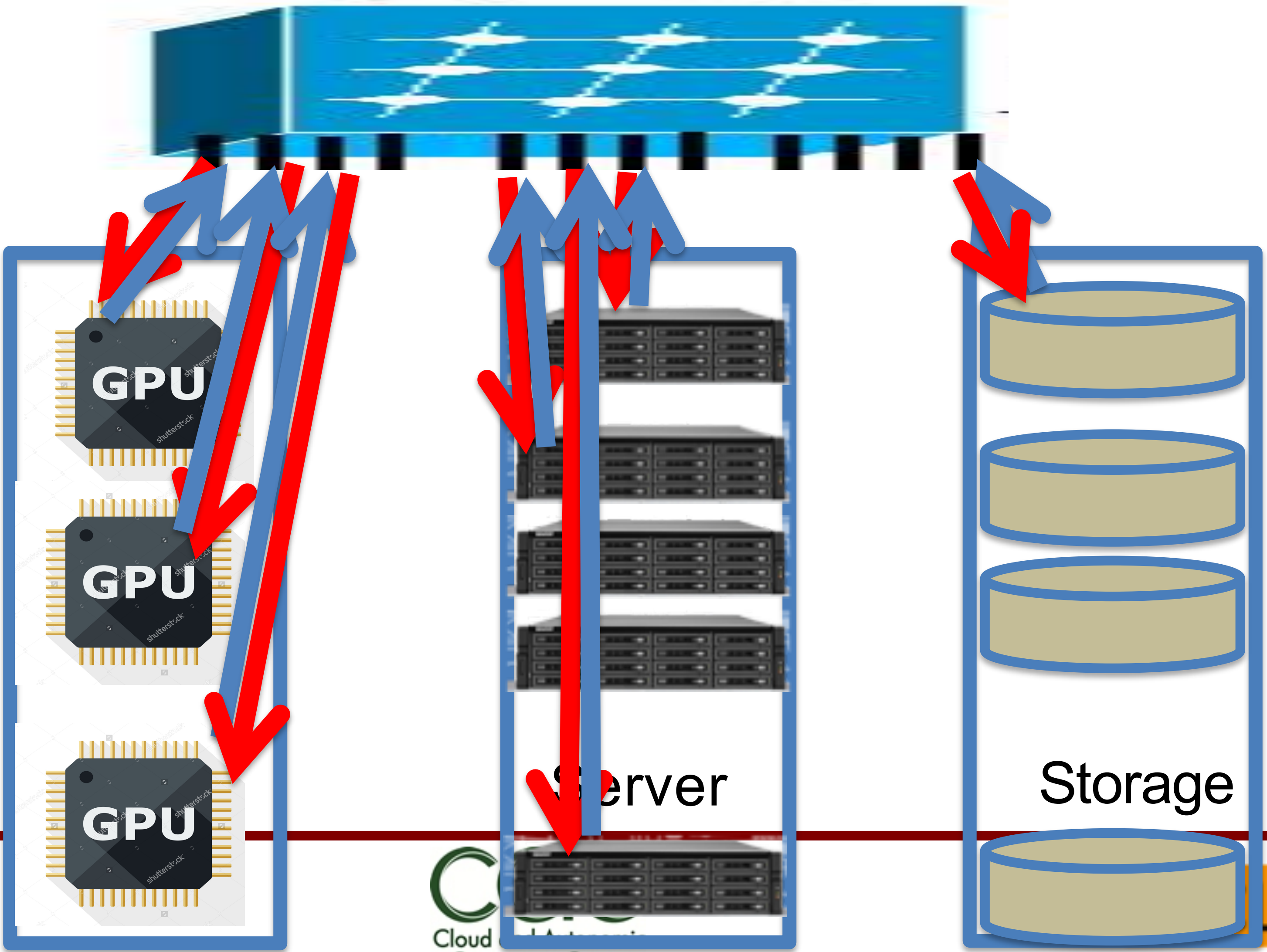
JITA Optical Interconnect



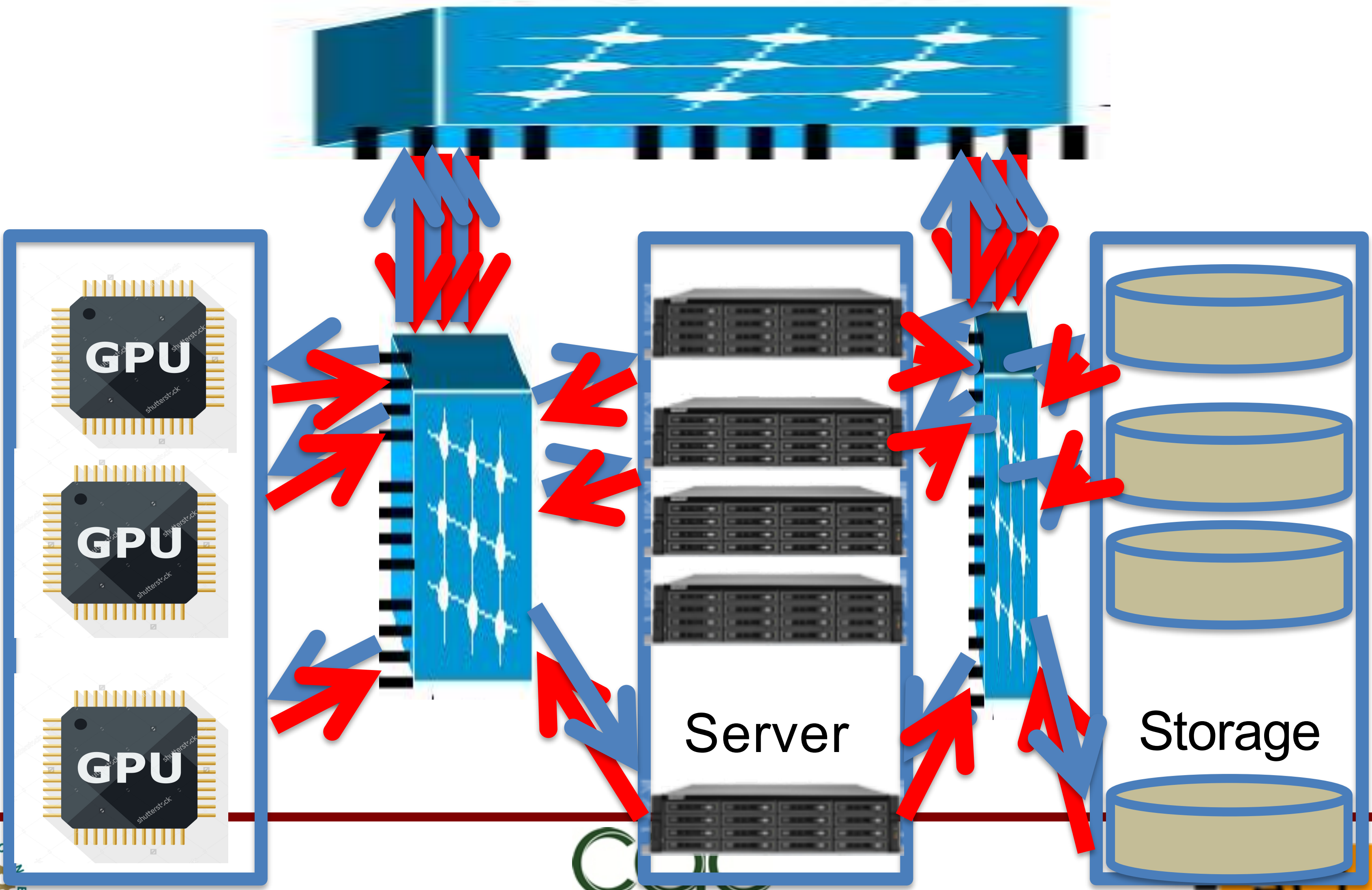
End of Row (EoR) Topology


End of Row Network Connectivity Architecture
Rack
End-of-Row Switch
Server
Chassis/ Aggregation Switch

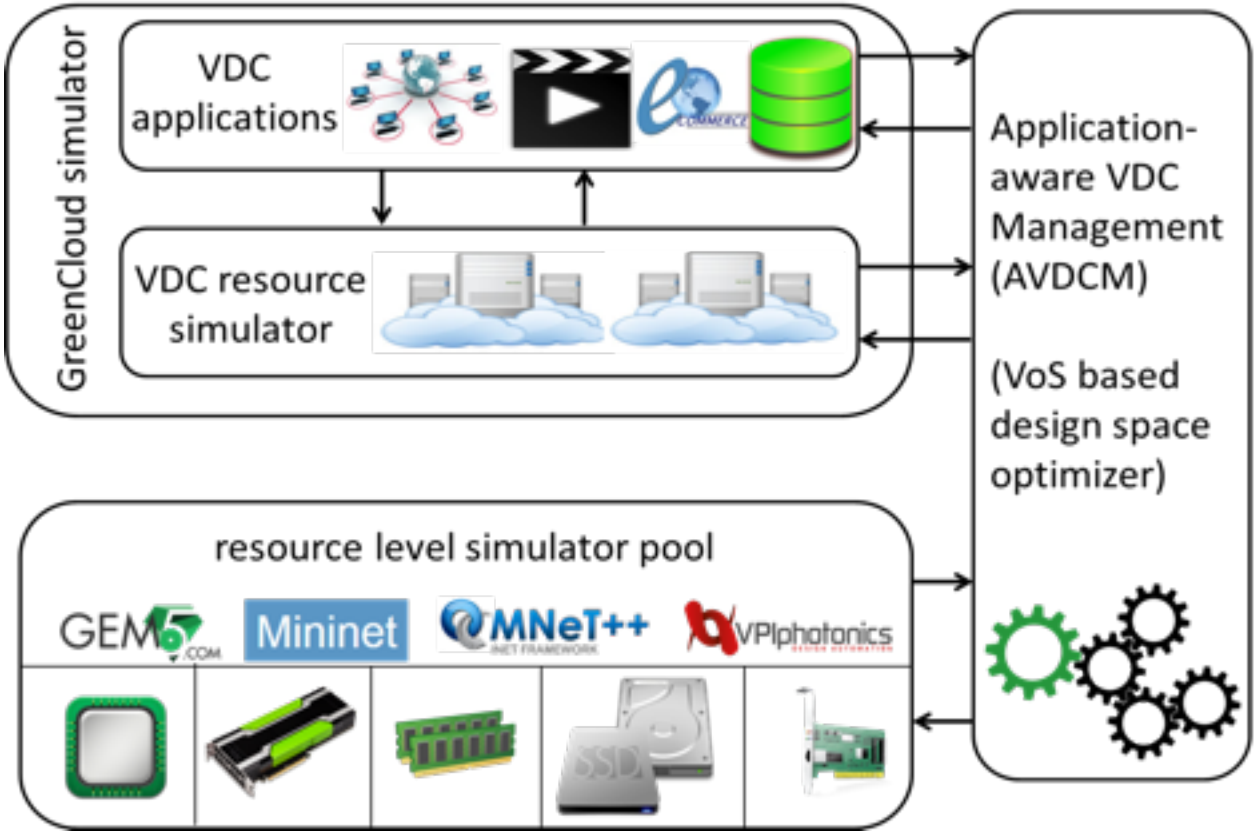
Optical End of Row (OEoR) Topology



Optical Top of Rack (OToR) Topology



Performance Modeling, Analysis and Simulation



Summary: Composable datacenter scale systems expose many more system knobs and need to be self-optimized

Many areas requires performance tuning

Hardware Configurations

CPU & Cache
- Adopt SMT4 for Terasort
- Prefetch from L2/L3/memory to D-L1
- Large on-chip cache, memory and IO bandwidth

Storage
- Software RAID over LVM to reduce storage layer overhead
- Symphony round-robin scheduling algorithm to utilize disk arrays

JVM
- GC and jitting policy
- Heap size
- Enable Huge Page

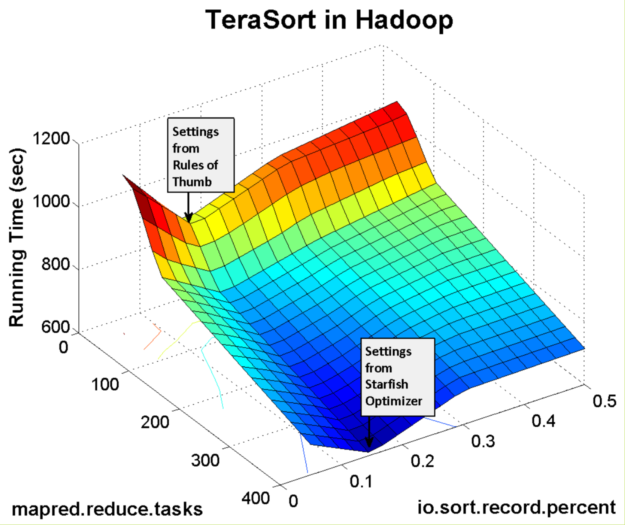
Platform Symphony
- Buffer related to reduce IO
- Smart scheduling
- Task granularity
- Resource Allocation

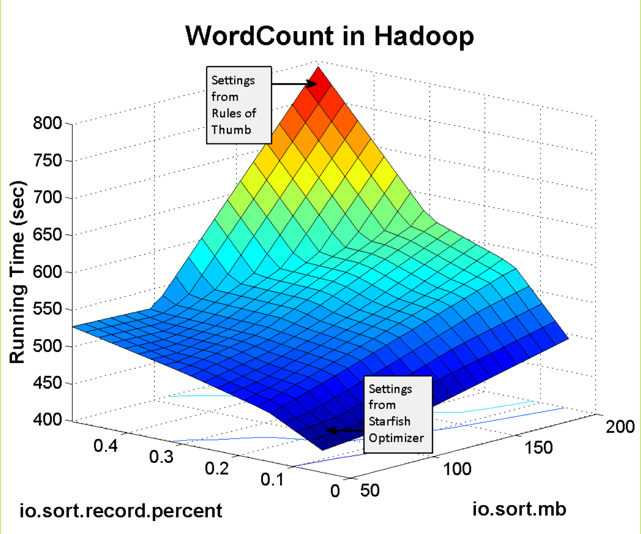
Compression Algorithm
- Gzip → LZO →SNAPPY →LZ4

Manual optimization of Terasort took 18 months

	Bottleneck
02/10/2012 **47 minutes**	**CPU**
27 minutes	**Disk IO**
22 minutes	**Memory**
19 minutes	**Disk IO**
15 minutes	
13 minutes 48 seconds (on p730)	**CPU/Memory, but software stack inefficent**
07/03/2012 **8 minutes 44 seconds (on 7R2)**	
02/05/2013 **7 minutes 50 seconds (on 7R2)**	**CPU/ Memory**
04/15/2013 **6 minutes 41 seconds (on 7R2)**	**CPU/ Memory**

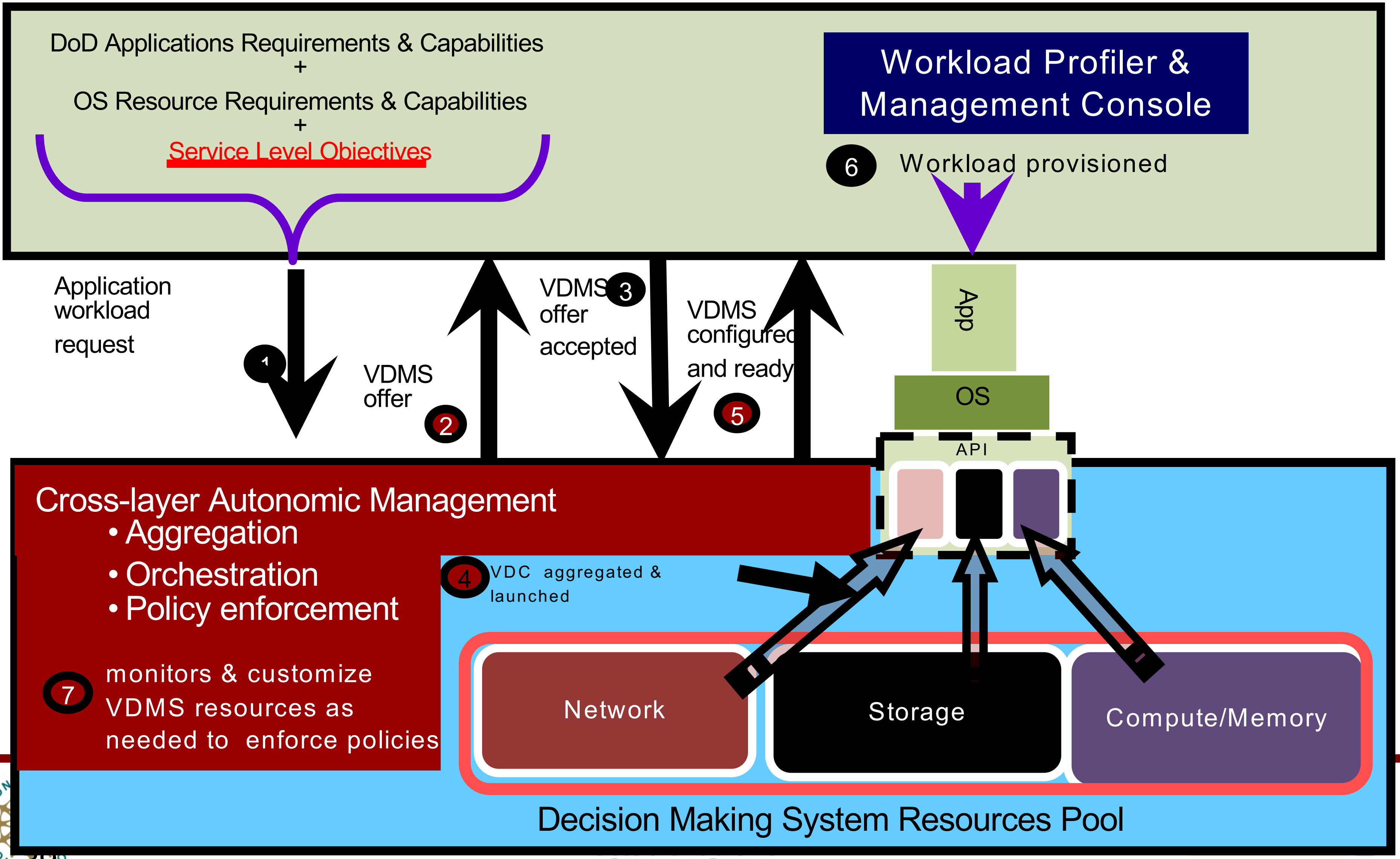
Self-tuning could achieve 75% of optimal performance within minutes


TeraSort in Hadoop


WordCount in Hadoop

Source: Duke Univ.

JITA Example Workload Profile


DoD Applications Requirements & Capabilities
+
OS Resource Requirements & Capabilities
+
Service Level Objectives
Workload Profiler & Management Console
6 Workload provisioned
Application workload request
1
VDMS offer
2
VDMS offer accepted
3
VDMS configured and ready
5
App
OS
API
Cross-layer Autonomic Management
• Aggregation
• Orchestration
• Policy enforcement
4 VDC aggregated & launched
7 monitors & customize VDMS resources as needed to enforce policies
Network
Storage
Compute/Memory
Decision Making System Resources Pool

Conclusions

- Autonomic computing can paly an important role in designing composable data centers
- Software Defined Infrastructures are a key technology to be leveraged in the development of software architecture and middleware
- Optical Interconnect technology must be leveraged
- Automated configuration and tuning are key design parameters

Questions?

Contact Dr. Hariri at hariri@ece.Arizona.edu