

Smart Edge: A Software-Defined Infrastructure for Future Application Platforms

Raouf Boutaba

David R. Cheriton School of Computer Science
University of Waterloo - Canada

CNSM 2014, Rio De Janeiro (Brazil), November 18, 2014

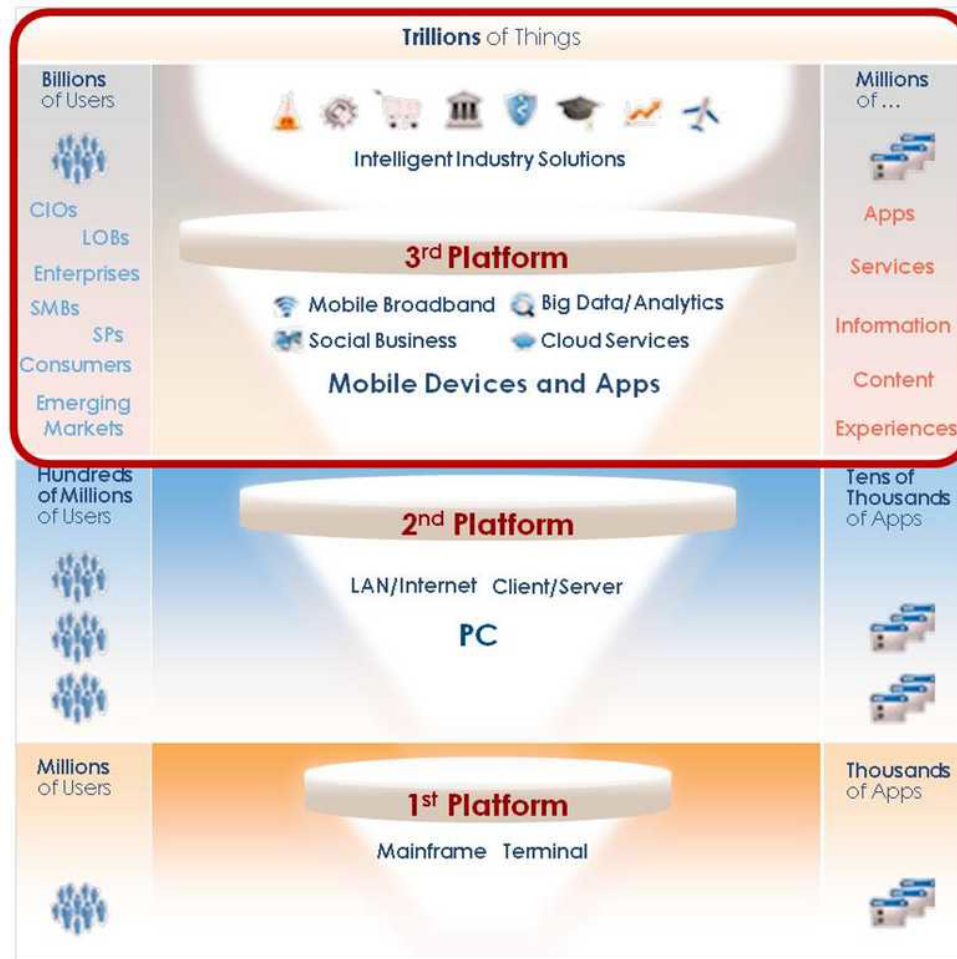
Outline

- ▣ Future Application Platforms: trends and challenges
- ▣ The SAVI Project
- ▣ SAVI Smart Edge
- ▣ Sample Research Contributions
 - ▣ VDC Planner
 - ▣ Venice
 - ▣ Greenhead
- ▣ Conclusion

ICT Innovation

**ICT Innovation
2005–2020+**

**ICT Innovation
1985–2005**



IDC top 10 predictions
report, 2013

3rd Platform's 4 Pillars

- Mobile broadband
- Cloud-based services
- Big Data analytics
- Social media



Mobile Broadband

- High connectivity
- High transmission speed
- Low latency
- 5G ?

Cloud Services

- Economical
- Scalable
- Elastic
- Flexible

Big Data Analytics

- Ability to harness and analyse large and complex sets of data
- The 3 Vs
 - Volume
 - Velocity
 - Variety
- More Vs ?

Social Media

- ▣ Social Networking
- ▣ Social Business solutions
- ▣ Other innovative applications
 - ▣ Social media search
 - ▣ Social gaming
 - ▣ ...

Future Application Platform

- ▣ Mobile broadband
- ▣ Cloud-based services
- ▣ Big Data analytics
- ▣ Social media



➔ SAVI Project

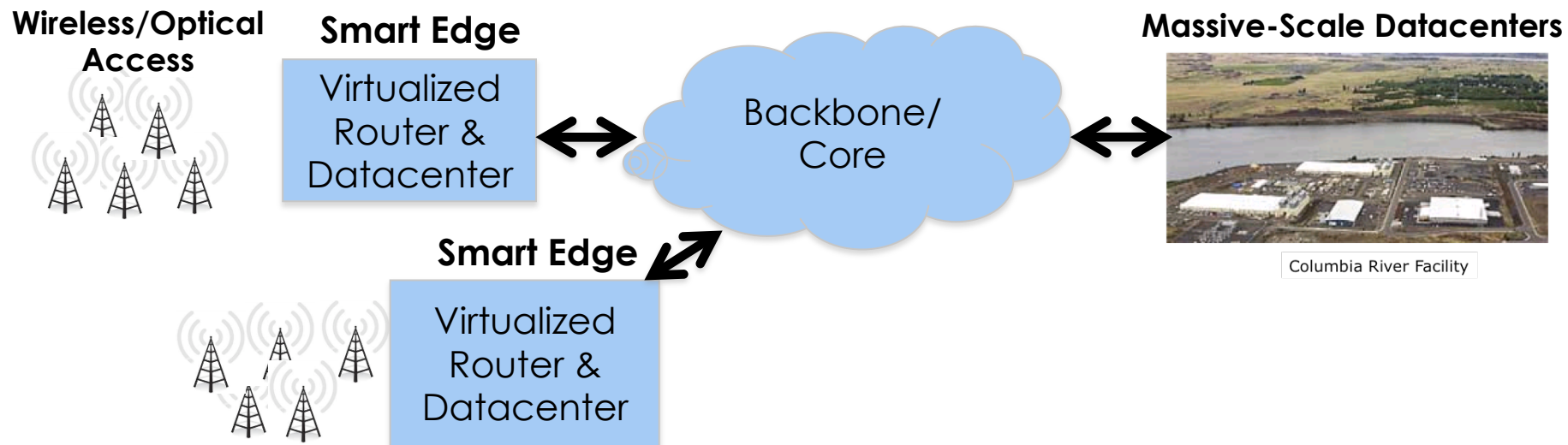
Outline

- ▣ Future Application Platform: trends and challenges
- ▣ The SAVI Project
- ▣ SAVI Smart Edge
- ▣ Sample Research Contributions
 - ▣ VDC Planner
 - ▣ Venice
 - ▣ Greenhead
- ▣ Conclusion

The SAVI Project

- SAVI : Smart Applications on Virtual Infrastructures
 - An NSERC Strategic Research Network
 - 9 Universities (15 professors & > 50 graduate/postgraduate students)
 - 13 Industry Partners (IBM, Cisco, Ericsson, Juniper, TELUS, ...)
- SAVI Research Themes:
 - Smart applications
 - Extended Cloud Computing
 - Smart Converged Edge
 - Integrated Wireless Optical Access
 - SAVI Application Platform Testbed

SAVI Vision

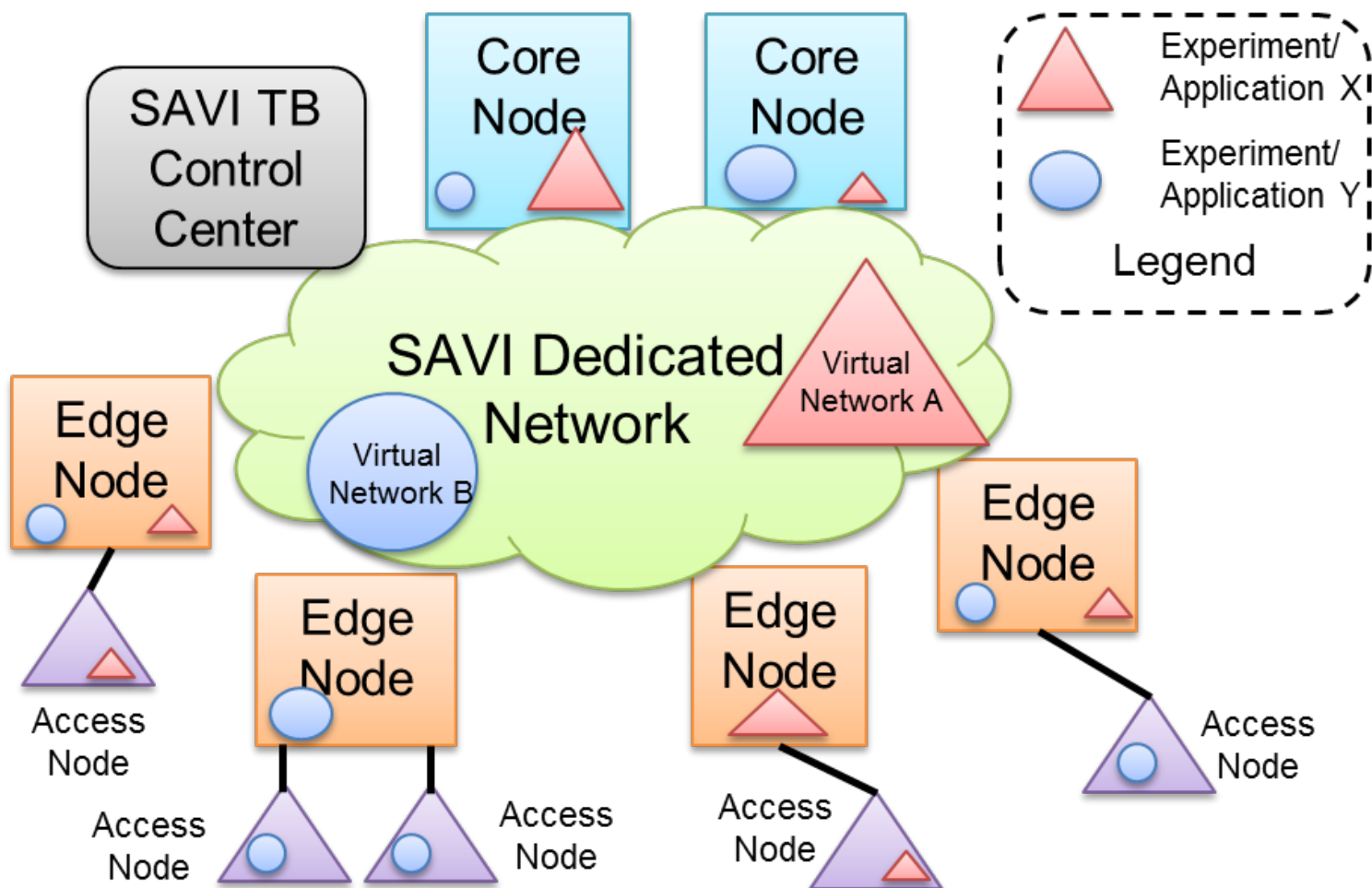


Research Scope

- Extended computing cloud that includes smart edge
- Application enablement leveraging very-high bandwidth access and low-latency services in the smart edge and massive remote cloud resources
- Integrated wireless/optical access controlled by the smart edge
- Control & Management system to enable experimentation with service applications and Future Internet architectures
- SAVI Testbed

SAVI Testbed

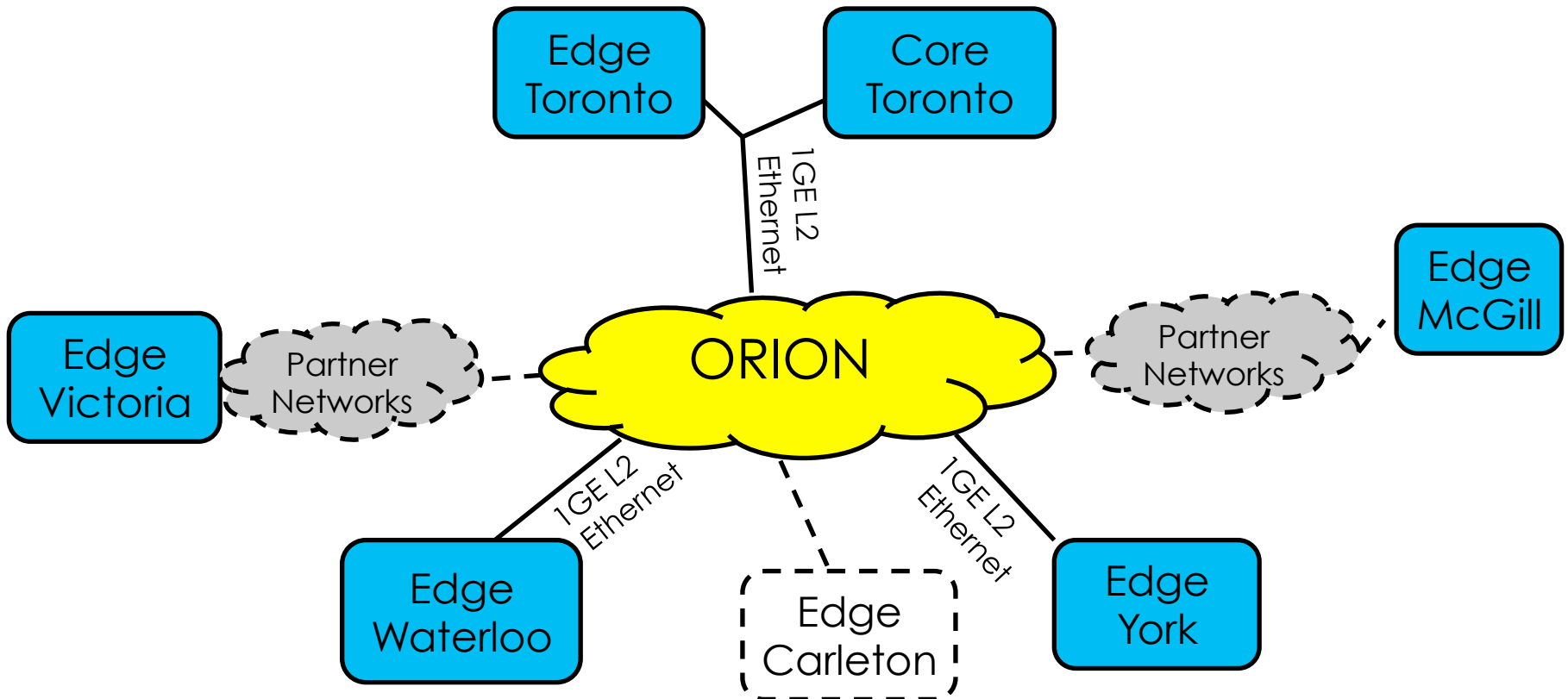
- Experimenters request slices of resources
- Interconnected to form virtual infrastructures



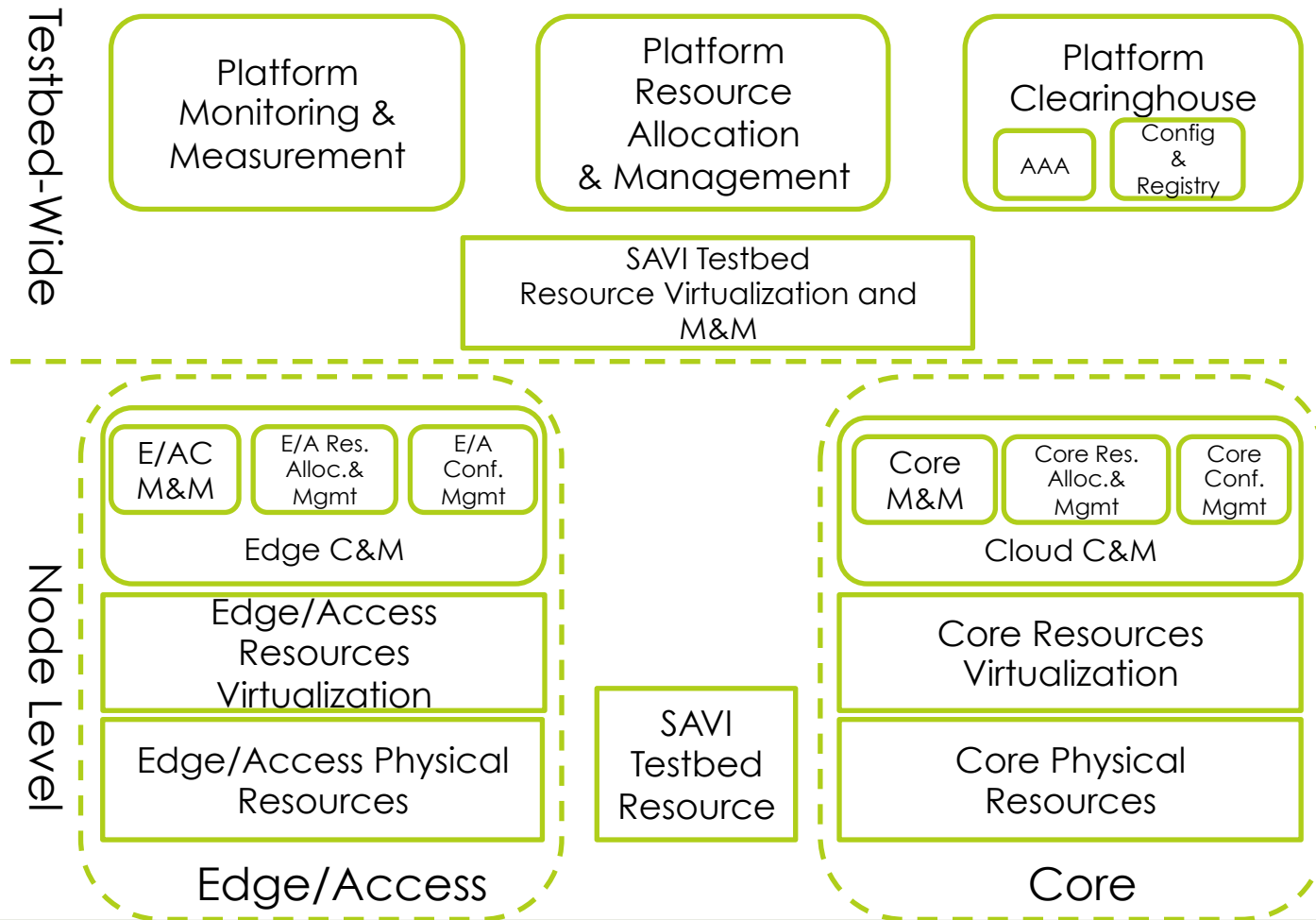
SAVI Testbed Resources

- ▣ 550+ cores
- ▣ 10+ FPGA systems
- ▣ 6+ GPU systems
- ▣ 50+ TB storage
- ▣ 10/1 GE fabrics (OpenFlow)
- ▣ 1GE dedicated backbone: ORION

SAVI Testbed Dedicated Network



SAVI TB Control and Management



The Smart Edge - Goals

- To develop a virtualized network and computing infrastructure that provides responsive and high-capacity virtualized resources and services close to the user.
- To provide converged computing, networking, programmable hardware processing, and storage that complement the resources provided in remote datacenters
- To implement edge routers using virtual resources with a focus on energy efficiency, scalability and programmability.

Smart Edge Characteristics

- Converged

- Provides access to heterogeneous resources

- Computing, storage and networking (data center, WAN)

- Programmable hardware (GPUs, FPGAs, NetFPGAs, BEE boards)

- On-Premises

- Can be isolated from rest of network while accessing local resources

- M2M, e.g., security/safety systems

- Proximity

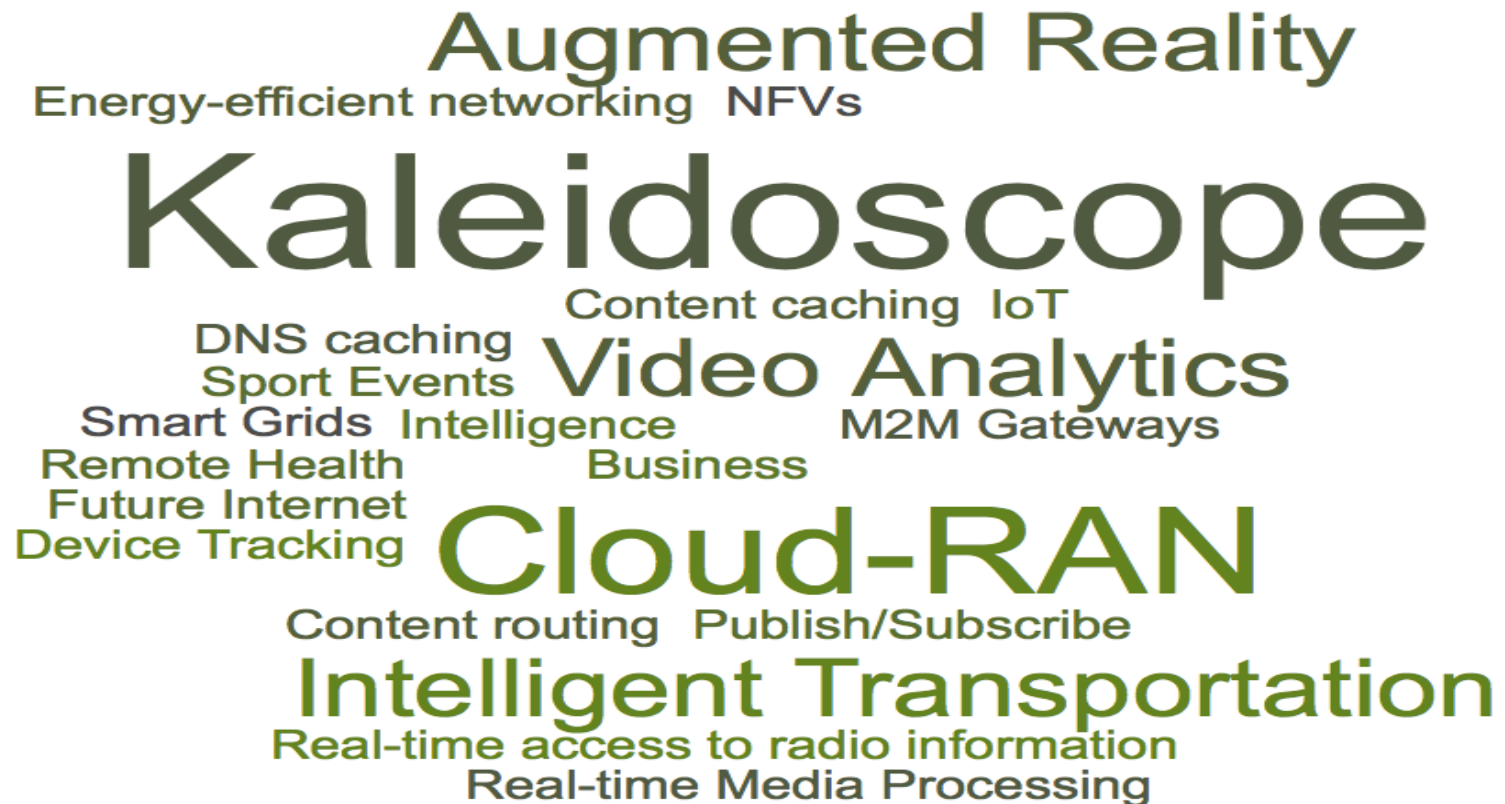
- Close to source of information, can capture key information for Big Data Analytics

- Direct access to devices, e.g., can be leveraged by business specific apps

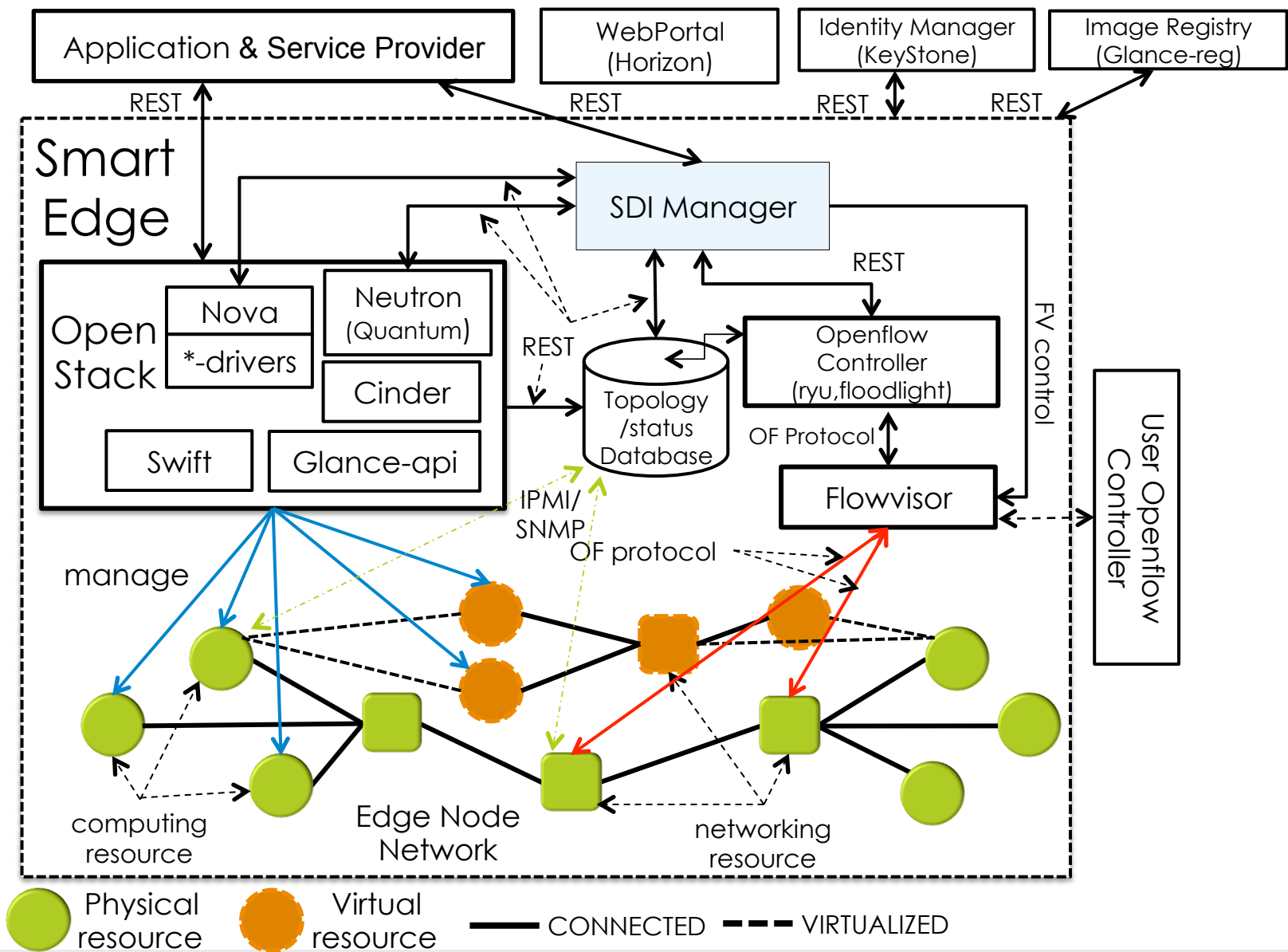
Smart Edge Characteristics (cont)

- Low-latency
 - Close to end user, reducing latency considerably
 - React faster, improve QoE, minimize congestion elsewhere
- Location awareness
 - User/device tracking
 - Location-based services (local points of interest), analytics, etc.
- Network context information
 - Real-time network data (e.g., radio condition, network stats)
 - Context-aware services/apps for improved QoE
- Programmability
 - Software-Defined Infrastructure: Combines cloud computing technologies and software-defined networking under a single management system

Use Cases



Smart Edge Integrated C&M

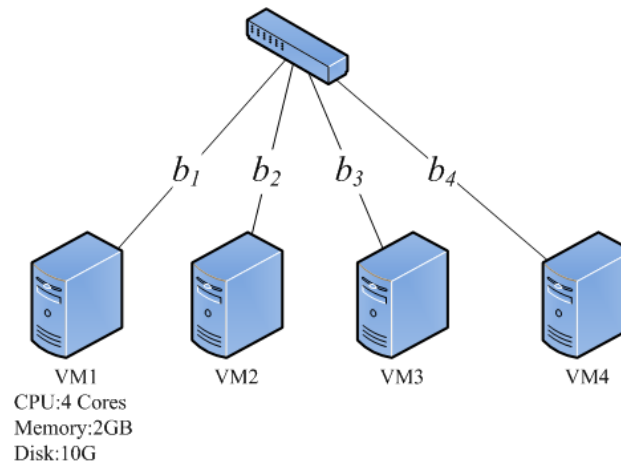


Outline

- ▣ Future Application Platforms: trends and challenges
- ▣ The SAVI Project
- ▣ SAVI Smart Edge
- ▣ Sample Research Contributions
 - ▣ VDC Planner
 - ▣ Venice
 - ▣ Greenhead
- ▣ Conclusion

Virtual Data Centers

- Currently cloud providers provides only computing resources but do no provide guaranteed network resources
- Goal: Provide both guaranteed computing and network resources
 - Virtual Data Centers (VDCs): virtual machines, routers, switches and links



VDC Embedding

■ Objectives

- Map VDCs onto physical infrastructure (Computing + networking resources)
- Maximize acceptance ratio/revenue
- Minimize energy costs
- Minimize the scheduling delay
- Achieve all of the above objectives dynamically over-time

■ VDC Planner*

- A migration-aware virtual data center embedding framework
- VDC embedding, VDC scaling
- Dynamic VDC consolidation.

*M. F. Zhani, Q. Zhang, G. Simon, R. Boutaba. VDC Planner: Dynamic Migration-Aware Virtual Data Center Embedding for Clouds. IFIP/IEEE IM'13. Ghent (Belgium), May 27-31, 2013.

VDC Planner Architecture

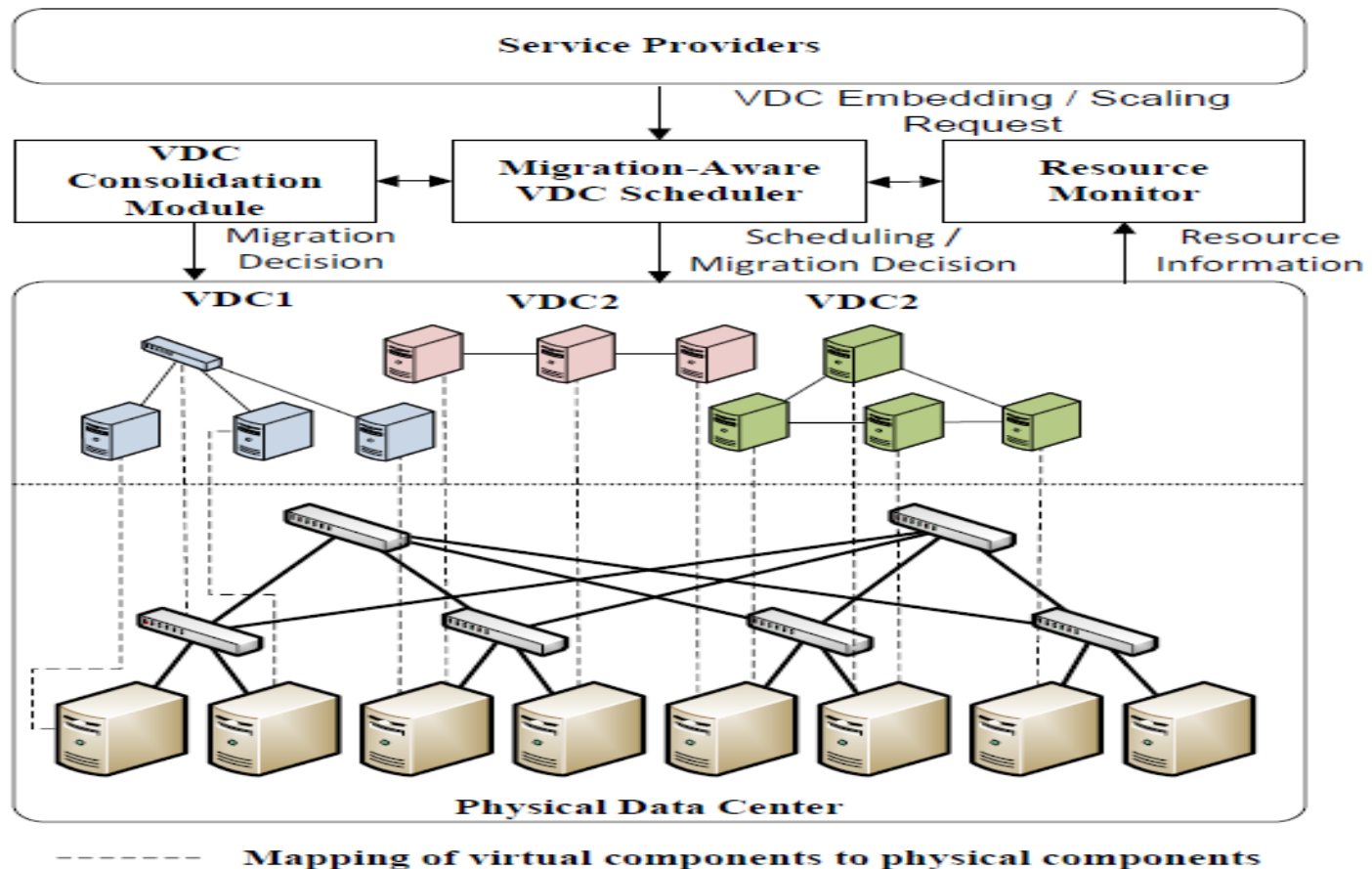
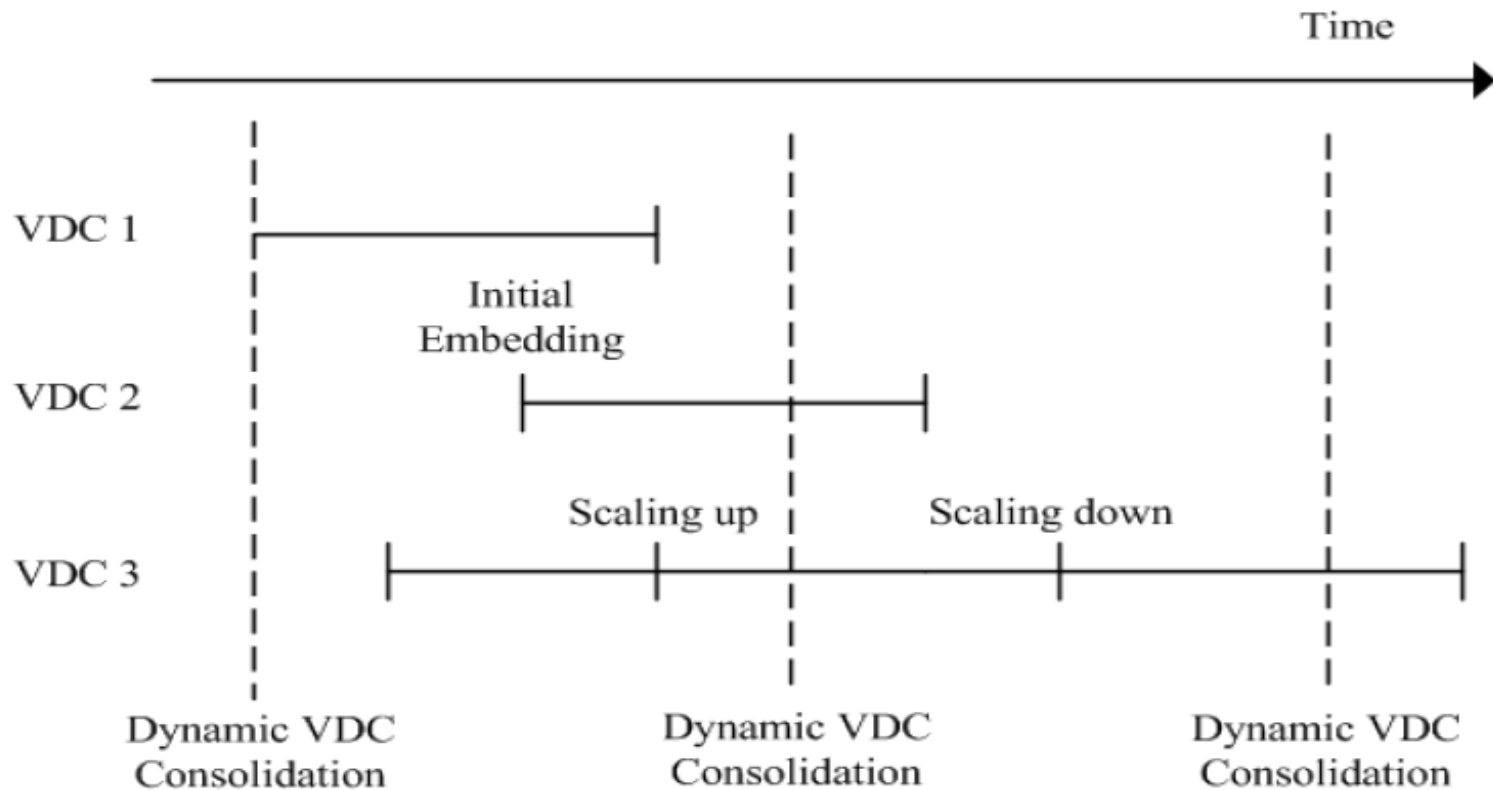


Figure 1: VDC Planner Architecture

VM Migration: Usage Scenarios



Problem formulation

■ Objective function

$$\min \underbrace{\sum_{\bar{n} \in \bar{N}} y_{\bar{n}} p_{\bar{n}}}_{\text{Operational costs}} + \underbrace{\sum_{i \in I} \sum_{n \in N^i} \sum_{\bar{n} \in \bar{N}} \gamma_n x_{n\bar{n}}^i g_{n\bar{n}}^i}_{\text{embedding cost}}$$

$y_{\bar{n}}$ a Boolean that indicates that \bar{n} is active

$x_{n\bar{n}}^i$ a Boolean that indicates that n is embedded in \bar{n}

■ Embedding cost

$$g_{n\bar{n}}^i = \begin{cases} \text{mig}(n, \bar{m}, \bar{n}) & \text{if } \bar{n} \neq \bar{m} \\ 0 & \text{if } \bar{n} = \bar{m} \\ 0 & \text{if } n \text{ is currently not embedded} \end{cases}$$

Migration-Aware VDC Embedding

- Sort the VMs by their size

$$size_n^i = \sum_{r \in R} w^r c_n^{ir}$$

- Compute the embedding cost (for each VM and physical node)

$$\begin{aligned} cost^i(n, \bar{n}) = & \gamma_n(mig(n, \bar{m}, \bar{n}) + MigOther(n, \bar{n})) \\ & + \sum_{n' \in N^i: (n', n) \in L^i} d(\bar{n}', \bar{n}) \cdot b_{(n', n)} \end{aligned} \quad (19)$$

- Embed the VM in the physical machine with the minimal embedding cost

Dynamic VDC Consolidation

- Sort the physical nodes in increasing order of their utilizations

$$U_{\bar{n}} = \sum_{r \in R} \sum_{i \in I} \sum_{n \in N^t: n \in loc(\bar{n})} \frac{w^r c_n^{ir}}{c_n^r},$$

- Migrate the VMs hosted in low-utilization machines (using Migration-Aware VDC Embedding Algorithm)
- If all VMs are successfully migrated, the machine is turned off.

Experiments

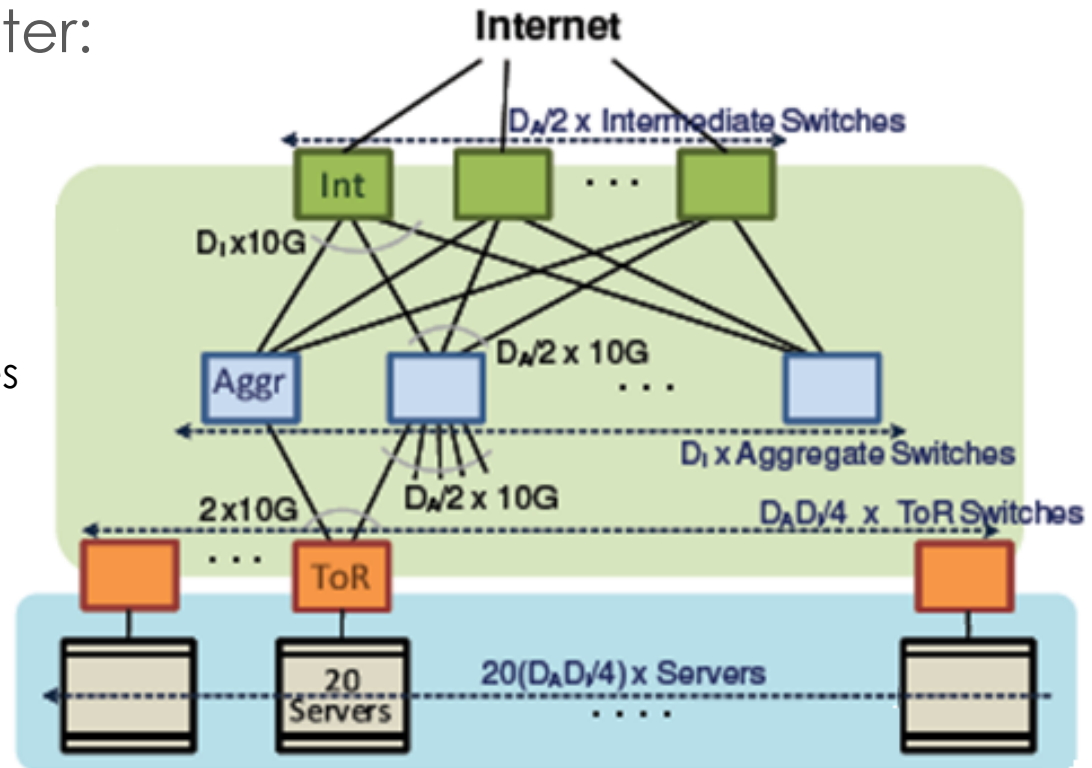
■ Physical data center:

4 core switches

4 aggregation switches

4 top-of-rack switches

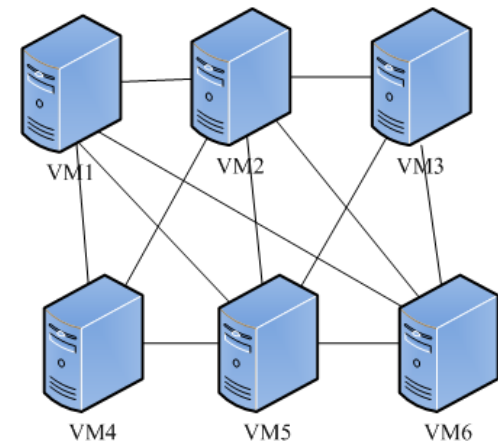
400 physical machines
(8 Cores, 8GB, 100 GB disk).



VL2 Topology

Experiment Set up

- VDC requests:
 - Number of VMs/VDC: [1-20]
 - VM requirements:
 - 1 – 4 cores
 - 1 – 2GB of RAM
 - 1 – 10GB of disk space
 - Virtual link capacity: [1-10 Mbps]
 - Arrival: Poisson distribution
 - 0.01 request/second during night time
 - 0.02 request/second during day time
 - VDC lifetime: exponential distribution (~3 hours)
 - Maximum waiting time: 1 hour



Performance Metrics

- Gain in acceptance Ratio

$$A_{m/n} = 100 \times \frac{A_m}{A_n} - 100$$

- Gain in revenue

$$G_{m/n} = 100 \times \frac{R_m}{R_n} - 100$$

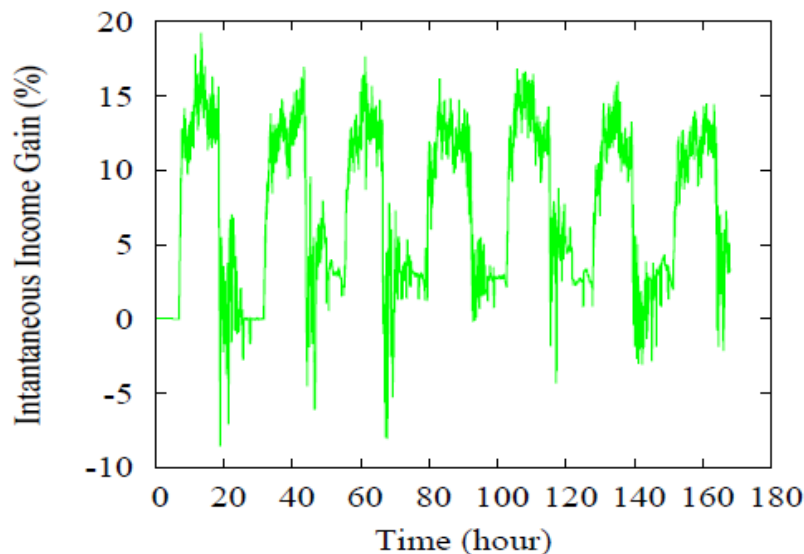
- Gain in number of active machines

$$M_{m/n} = 100 \times \frac{M_m}{M_n} - 100$$

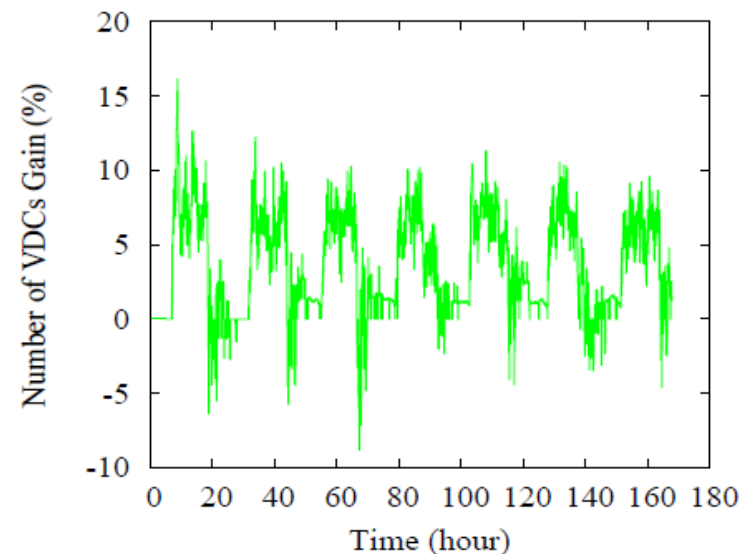
- Request scheduling delay

Results: Revenue & Acceptance Ratio

■ Migration-aware Embedding vs. Baseline



(a) Instantaneous income gains

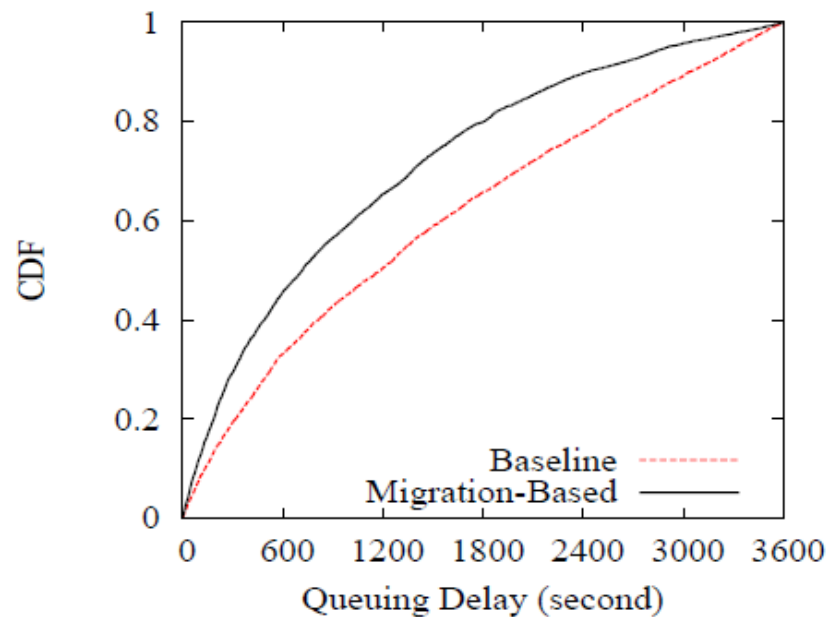


(b) Gain in acceptance ratio

- VDC planner achieves up to 17% gain in revenue and up to 10% gain in acceptance ratio.

Results: Queuing Delay

■ Migration-aware Embedding vs. Baseline

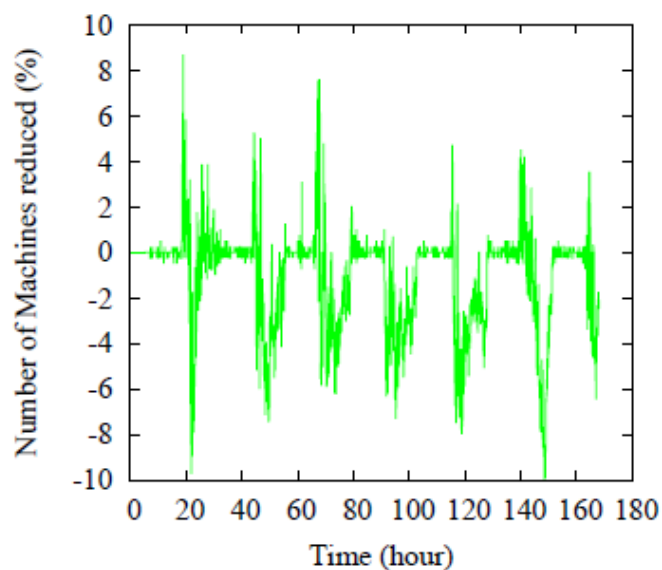


(c) Gain in terms of queuing delay

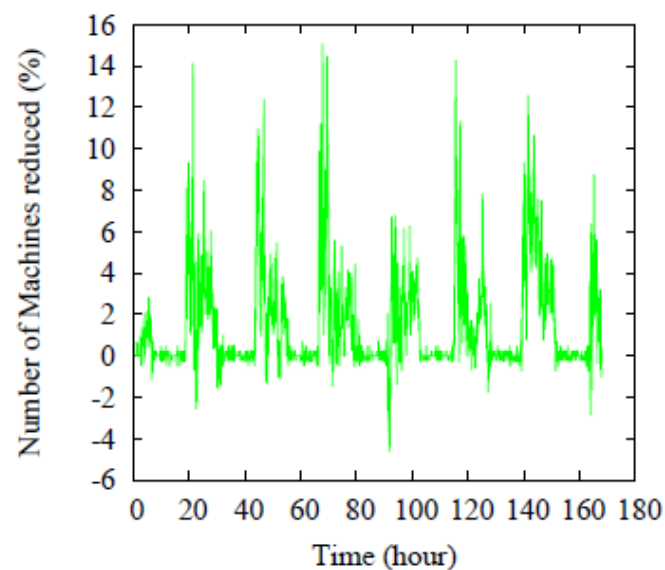
■ VDC planner reduces the scheduling delay by up to 25%.

Results: With Consolidation

■ Migration-Aware embedding + Consolidation



(a) Migration-aware algorithm



(b) Migration-aware embedding + consolidation

- VDC planner uses up to 14% less machines than the Baseline.

Outline

- ▣ Future Application Platforms: trends and challenges
- ▣ The SAVI Project
- ▣ SAVI Smart Edge
- ▣ Sample Research Contributions
 - ▣ VDC Planner
 - ▣ Venice
 - ▣ Greenhead
- ▣ Conclusion

VDC Reliability/Availability

- Reliability is a major concern of service providers
 - A service outage can potentially incur high penalty in terms of revenue and customer satisfaction
- Availability is a common reliability metric specified in SLAs
- VDC availability is dependent on
 - Service priority
 - VDC topology and replication groups
 - Hardware availability

Understanding Data Center Failures

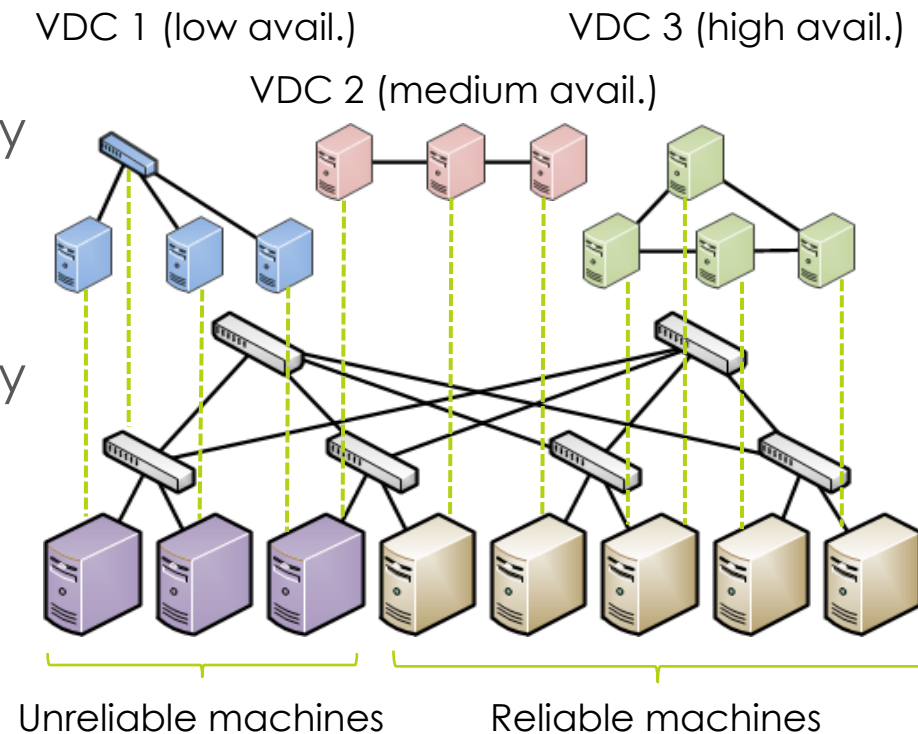
- Heterogeneous server failure rates*
 - Server that has experienced a failure is likely to fail again in the near future
- Network failure characteristics **
 - Failure rates of network equipment is type-dependent
 - Load balancers have high probability of failure ($\geq 20\%$),
 - Switches often have low failure probability ($\leq 5\%$).
 - Number of failures are unevenly distributed across equipment of the same type
 - E.g. Load balancer failures dominated by few failure prone devices
 - Correlated network failures are rare
 - More than 50% of link failures are single link failures, and more than 90% of link failures involve less than 5 links

* Vishwanath et al. "Characterizing Cloud Computing Hardware Reliability", SoCC 2010.

** Gill et. al. "Understanding network failures in data centers: measurement, analysis, and implications", SIGCOMM 2011.

Venice

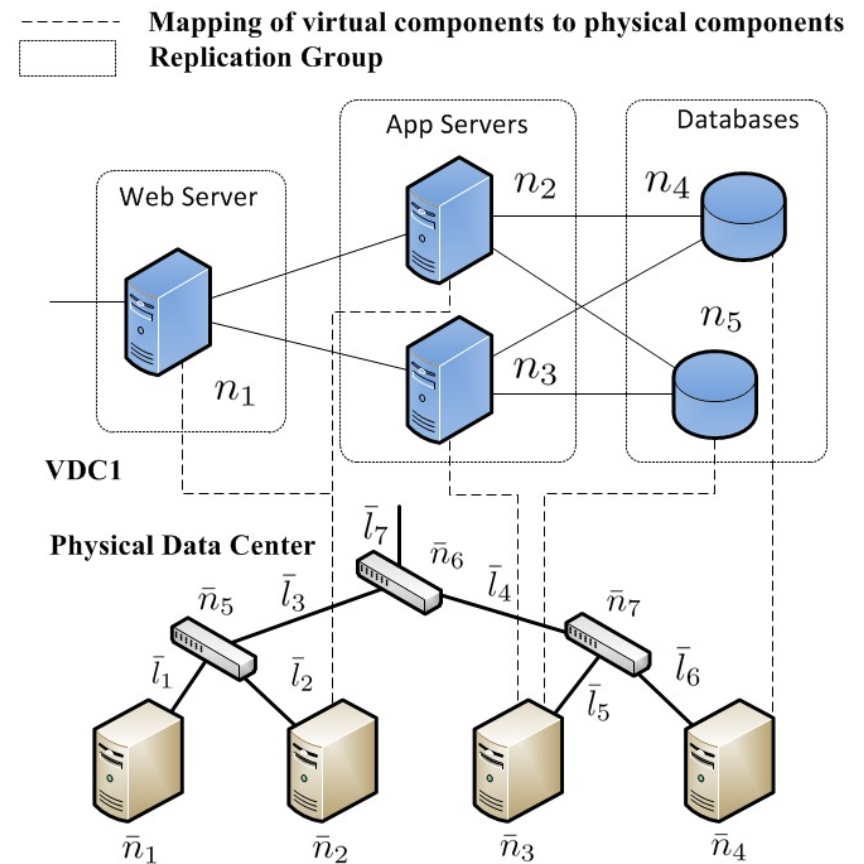
- VDCs have heterogeneous availability requirements
- Resources have heterogeneous availability characteristics
- Place VDCs with high availability requirement on reliable machines



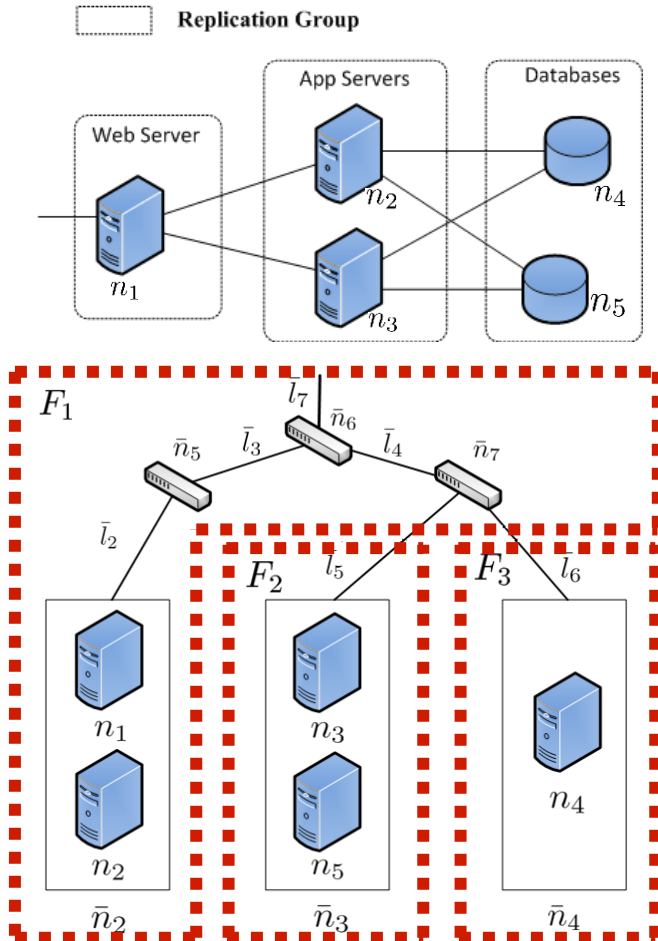
Computing VDC Availability

- Example of 3-tier application
- Availability of device j :

$$A_j = \frac{MTBF_j}{MTBF_j + MTTR_j}$$
- How to compute the availability of this VDC?



Computing VDC Availability (cont)



- Identify all possible failure scenarios S^k and compute the availability

Case 1: F1 unavailable,

$$A_{F_1} = 0$$

Prob. of occurrence: $P(F_1) = 1 - \prod_{i \in F_1} A_i$

Case 2: F1 available, F2 unavailable

$$A_{F_1} = \prod_{i \in F_2} A_i$$

Prob. of occurrence: $P(F_2) = (\prod_{i \in F_1} A_i) (1 - \prod_{i \in F_2} A_i)$

Case 3: F1 available, F2 available

$$A_{F_1} = 1$$

Prob. of occurrence: $P(F_2) = \prod_{i \in F_1 \cup F_2} A_i$

Using conditional probability, the availability of VDC_1 can be computed as:

$$A_{VDC_1} = \sum_{i=1}^3 P(F_i) A_{F_i}$$

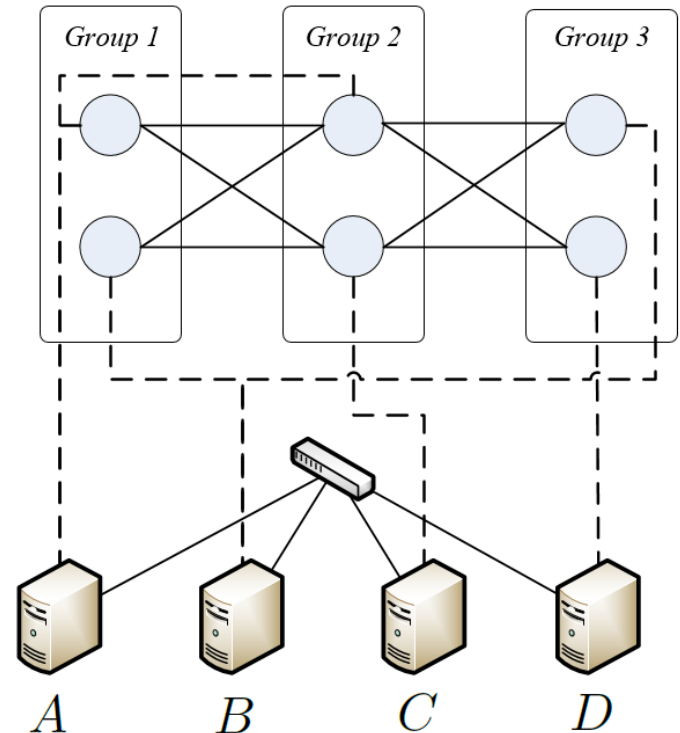
Computing VDC Availability (cont)

Theorem: VDC availability cannot be computed in polynomial time in the general case

Proof: Reduction from the counting monotone 2-Satisfiability problem

... Need to consider an exponential number of scenarios in the worst case!

$$f(A, B, C, D) = (A \vee B) \wedge (A \vee C) \wedge (B \vee D)$$



Computing VDC Availability (cont)

- Observation: it is unlikely to see large simultaneous failures
 - Given 3 nodes, each with availability $> 95\%$, the probability of seeing all 3 nodes fail simultaneously is at most $(1-0.95)^3 < 0.00013$
- A fast heuristic:
 - Compute availability using scenarios S^k that involve at most 3 simultaneous failures
- Fast heuristic provides a lower bound on VDC availability

Problem Formulation

- Objective function:

$$\min C_E + C_M + C_A$$

- Where

$$C_E = \sum_{\bar{n} \in \bar{N}} y_{\bar{n}} p_{\bar{n}} \quad (\text{Resource cost})$$

$$C_M = \sum_{i \in I} \sum_{n \in N^i} \sum_{\bar{n} \in \bar{N}} \gamma_n x_{n\bar{n}}^i g_{n\bar{n}}^i \quad (\text{Migration cost})$$

$$C_A = \sum_{i \in I} (1 - A_i) \pi_i + \sum_{\bar{n} \in \bar{N}} F_{\bar{n}} C_{\bar{n}}^{restore} + \sum_{\bar{l} \in \bar{L}} F_{\bar{l}} C_{\bar{l}}^{restore} \quad (\text{Failure cost})$$

Greedy Scheduling Algorithm

- For each received VDC request
 - Initial embedding: embed one node from each replication group.
 - Repeat
 - For each remaining component compute a score as the availability improvement - resource cost
 - Embed the component with the highest score
 - Until the VDC availability is achieved or all nodes are embedded
 - Embed the remaining components greedily based solely on resource cost

Experiments

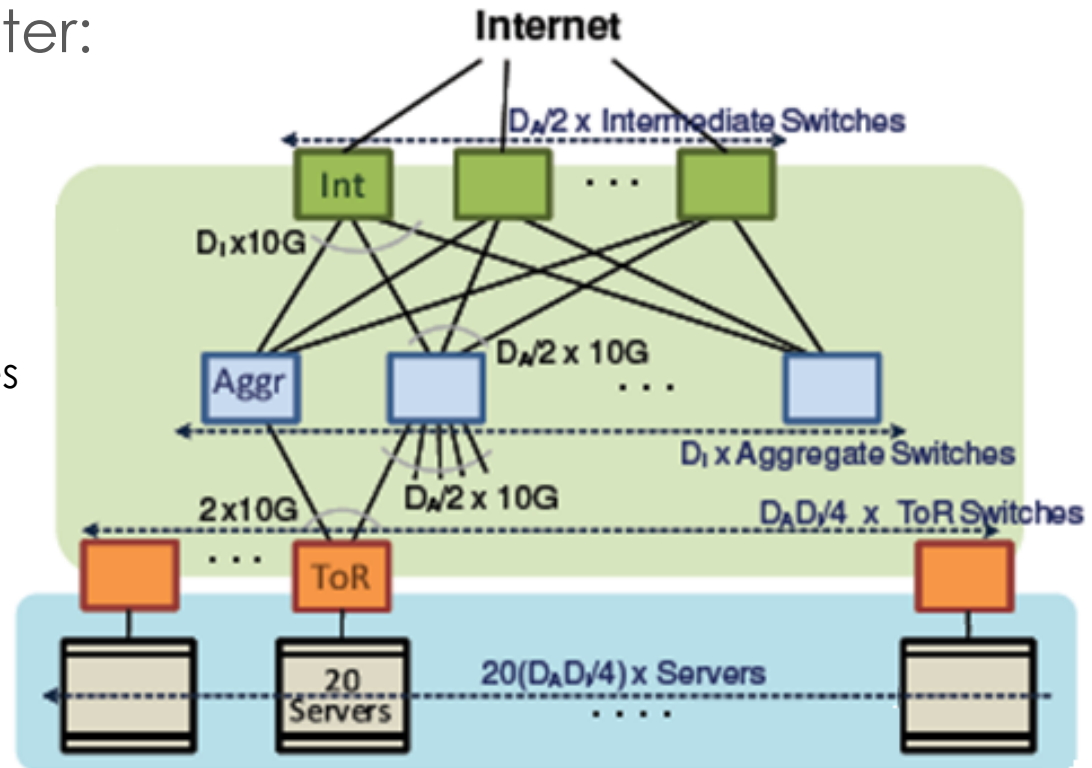
Physical data center:

4 core switches

4 aggregation switches

4 top-of-rack switches

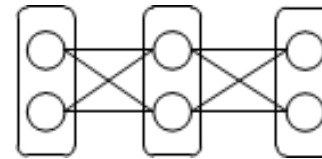
400 physical machines
(8 Cores, 8GB, 100 GB disk).



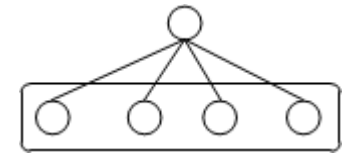
VL2 Topology

Experiment Setup

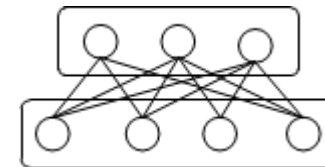
- VDC request formats
 - From 1 to 10 VMs per group
 - Different availability requirements
- VDC Planner used as a baseline for comparison



(a) Multi-tiered



(b) Partition-Aggregate

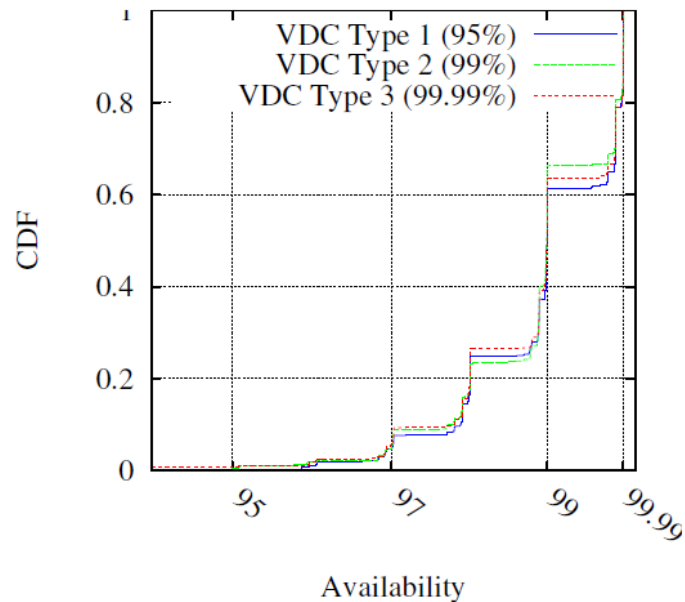


(c) Bipartite

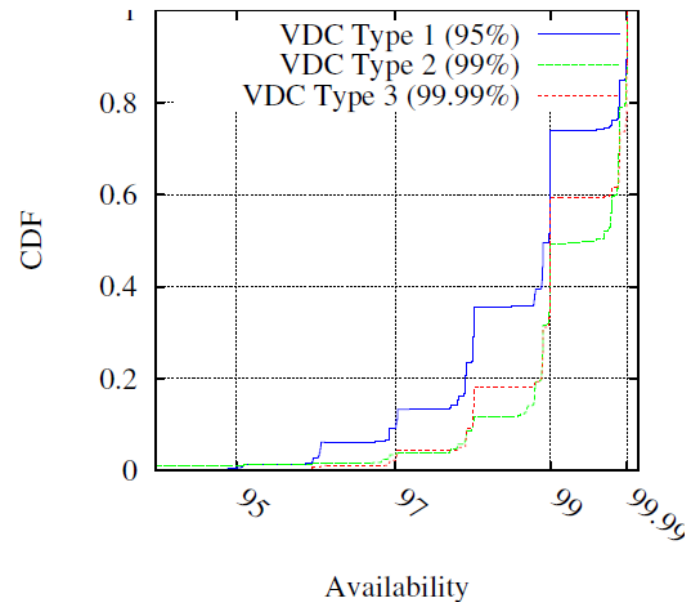
TABLE I: VDC Availability requirements

VDC Type	Minimum Required Availability (%)	Acceptable daily downtime
1	95.00	1h:12mn
2	99.00	14mn:2s
3	99.99	08.64s

Results: Availability



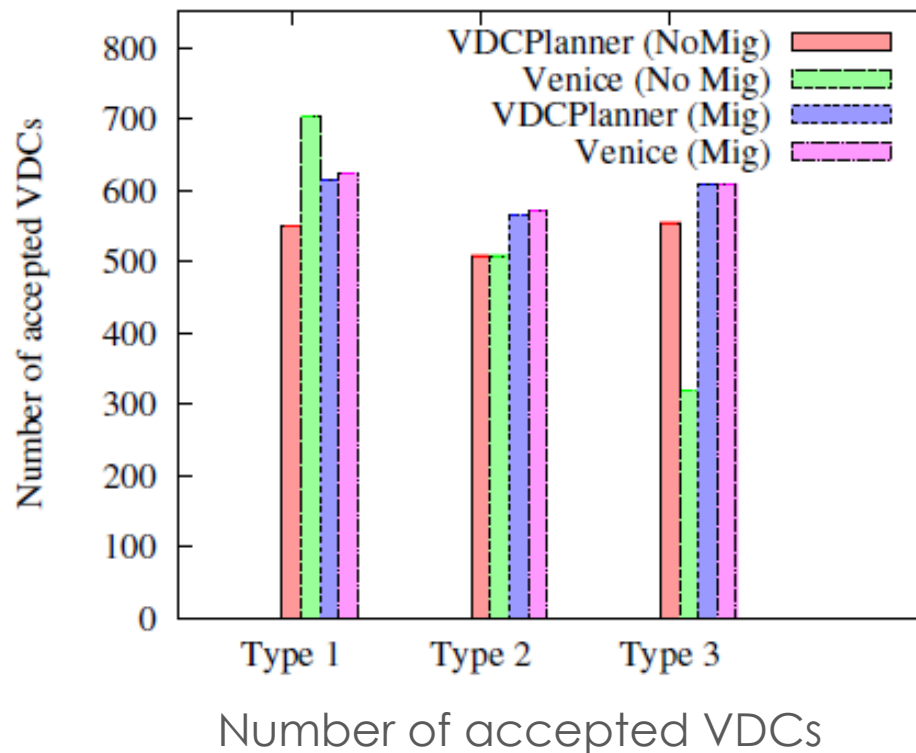
(a) VDC Planner (using migration)



(b) Venice (using migration)

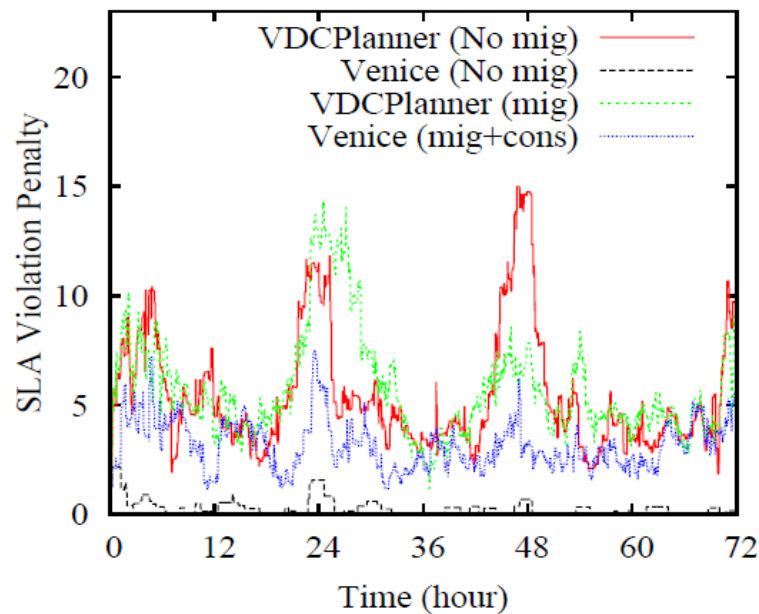
- Venice increases the number of VDCs satisfying availability requirements by up to 35%

Results: Acceptance Ratio

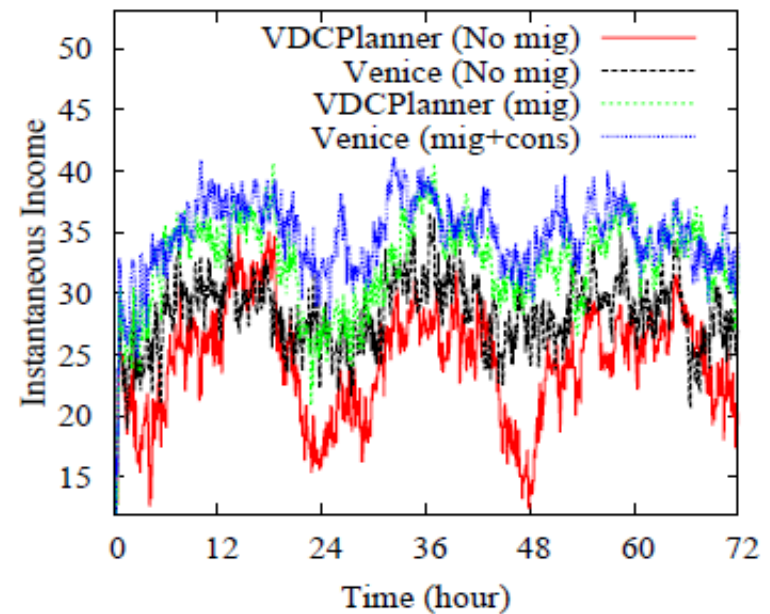


- With migration, the number of accepted VDCs is comparable to that of VDC Planner

Results: Revenue



SLA Violation Cost



Instantaneous Income Rate

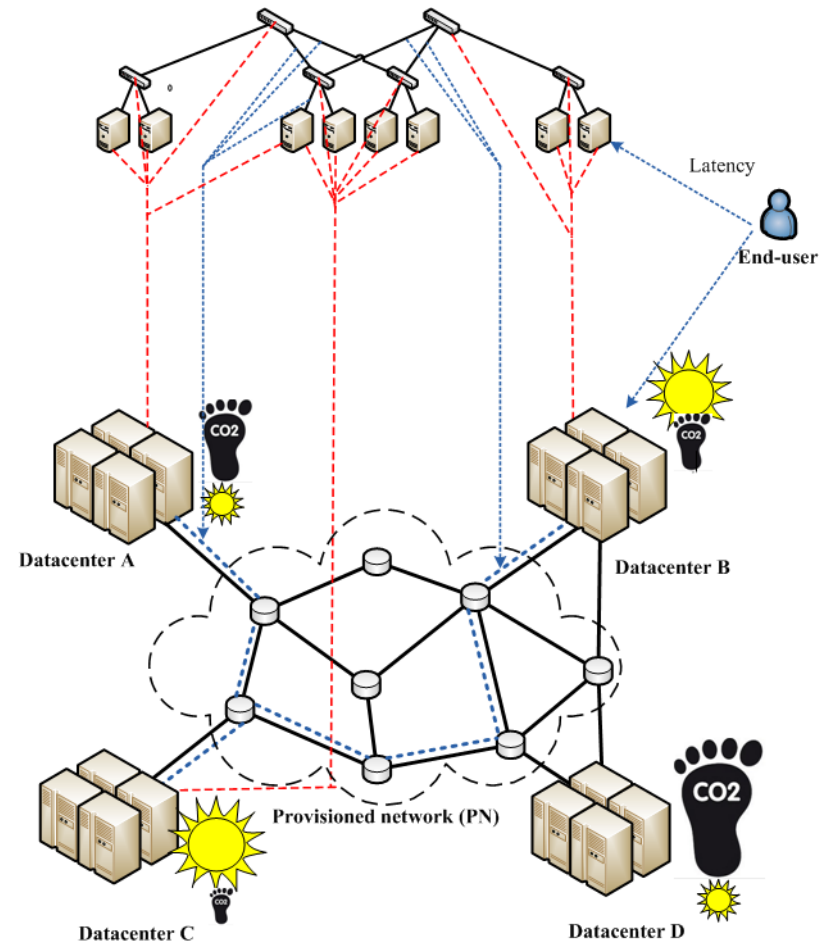
- Venice achieves 15% increase in revenue compared to VDC Planner

Outline

- ▣ Future Application Platforms: trends and challenges
- ▣ The SAVI Project
- ▣ SAVI Smart Edge
- ▣ Sample Research Contributions
 - ▣ VDC Planner
 - ▣ Venice
 - ▣ Greenhead
- ▣ Conclusion

Business Model

- Infrastructure Providers (InPs) build geographically distributed data centers
- InPs rent resources in the form of Virtual Data Centers (VDCs)
- VDCs geo-distributed to reduce energy cost and carbon footprint and/or to satisfy location constraints



InP Operational Objectives

- ▣ Satisfy performance requirements (e.g., latency)
- ▣ Reduce energy cost
 - ▣ Use data centers with low electricity prices
 - ▣ Reduce use of power from the grid
- ▣ Reduce carbon footprint
 - ▣ Use local renewable energy available at the data centers
 - ▣ Use source of power with the minimal carbon emission
- ▣ Reduce traffic in backbone network

Challenges

- Sources of clean energy for data centers:
 - Locally-available renewable energy sources (e.g., solar, wind)
 - Limited and dependent on location and weather conditions
- Electric grid:
 - Fluctuating prices of electricity and high carbon footprint
- Price and carbon footprint differ from one location to another

Greenhead Problem Formulation

- ILP decision variables:

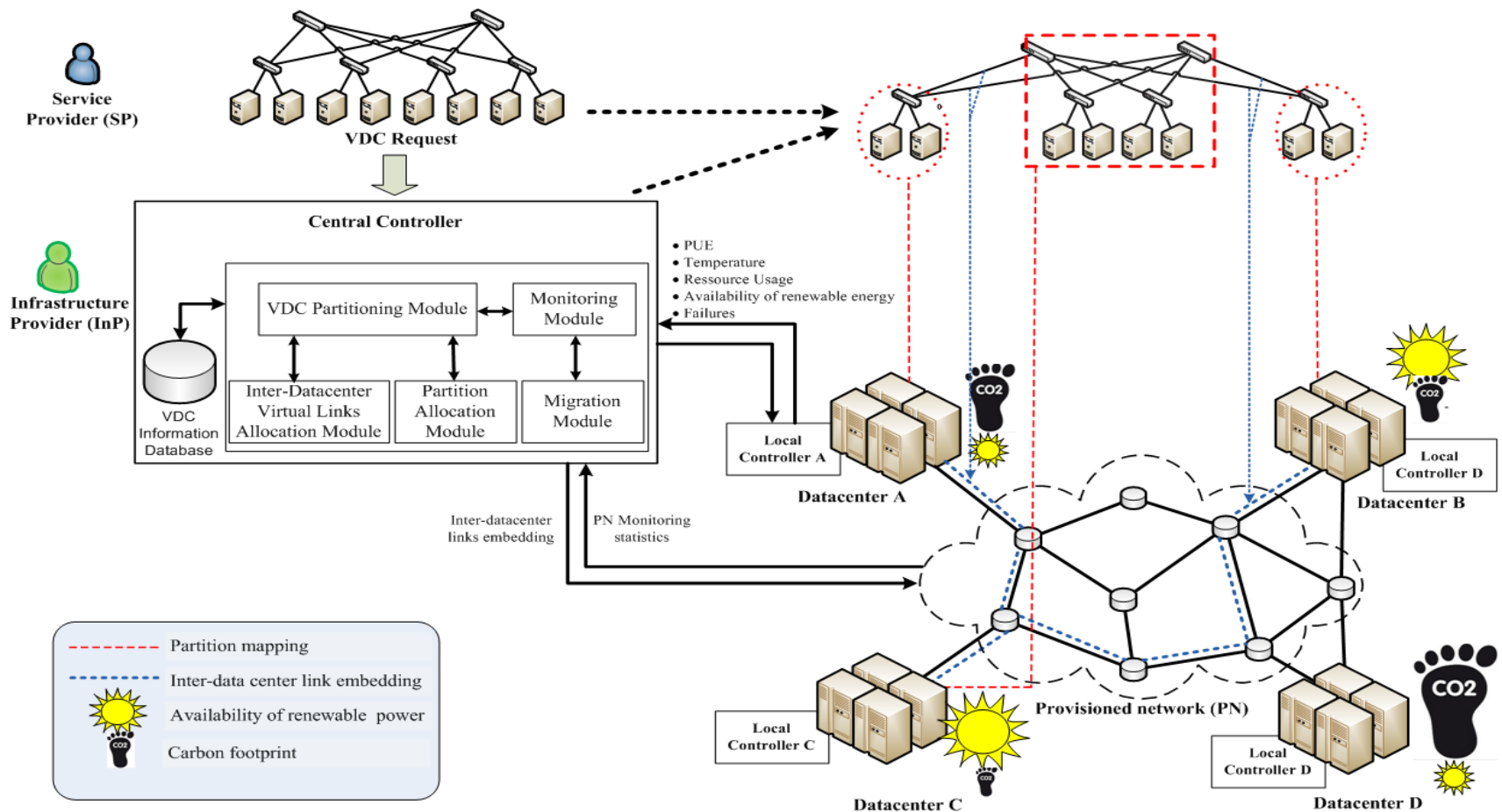
$$x_{ik}^j = \begin{cases} 1 & \text{If the VM } k \text{ of the VDC } j \text{ is} \\ & \text{assigned to data center } i \\ 0 & \text{Otherwise.} \end{cases} \quad f_{e,e'} = \begin{cases} 1 & \text{If the physical link } e \in E \text{ is used to} \\ & \text{embed the virtual link } e' \in E^j \\ 0 & \text{Otherwise.} \end{cases}$$

- Objective Function:

$$\text{Maximize } R^j - (D^j + P^j)$$

- R^j : Revenue for VDC request j
- D^j : Embedding cost for request j in data centers
- P^j : Embedding cost for request j in the backbone network

Greenhead Framework



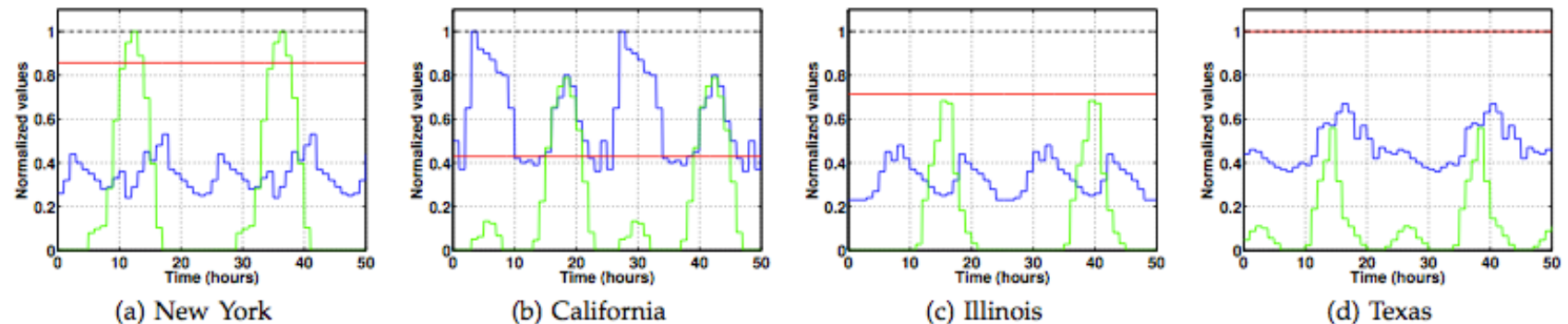
Greedy Algorithm

- Step 1 - VDC Partitioning: Location Aware Louvain Algorithm
 - Split the VDC requests into partitions with high intra-partition bandwidth demand and low inter-partition bandwidth demand
 - Minimize inter-data center traffic
- Step 2 – Partition embedding: Greedy partition assignment to data centers
 - Assign each partition to the data center that:
 - Satisfies location and delay constraints
 - Minimizes electricity costs and carbon footprint

Experiments

Physical infrastructure:

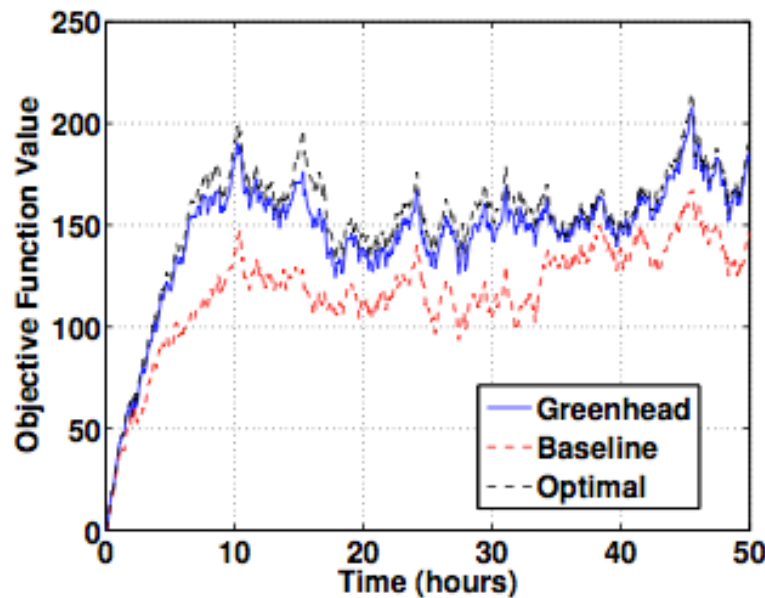
- 4 data centers located in New York, California, Illinois and Texas
- Backbone network: NSFNet network



Experiments Setup

- VDC requests:
 - Number of VMs randomly generated between 5 and 10 for small-sized VDCs and between 20 and 100 for large-sized VDCs
 - Virtual links randomly created with a bandwidth demand between 10 and 50 Mbps
 - Poisson arrivals (8 requests per hour) and exponential lifetime (average 6 hours)
 - 15 % of the VMs have location constraint
- Baseline approach: Greenhead without partitioning

Results: VDC Embedding



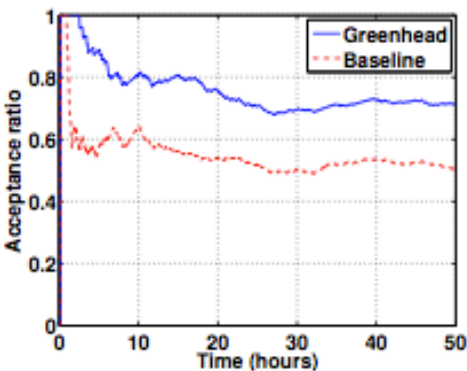
Objective function

VDC size	Greenhead			Baseline	ILP Solver
	Partitioning	Embedding	Total		
5-10 VMs	0.214	0.061	0.079	0.275	13540
20-100 VMs	31.41	0.28	31.69	2.2	-

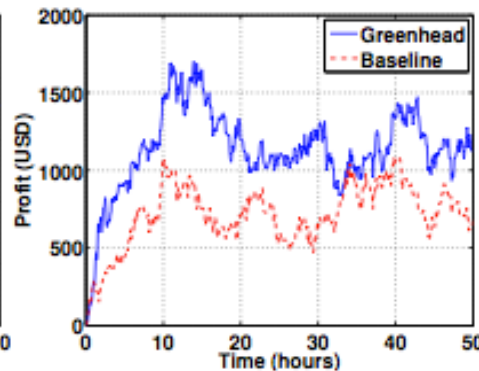
Computation Time (ms)

- Greenhead provides near-optimal solution within a reasonable time frame

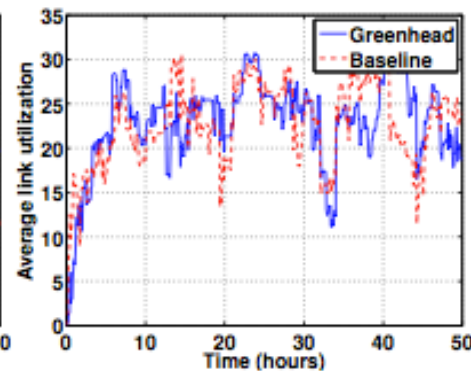
Results: Acceptance, Revenue, Utilization



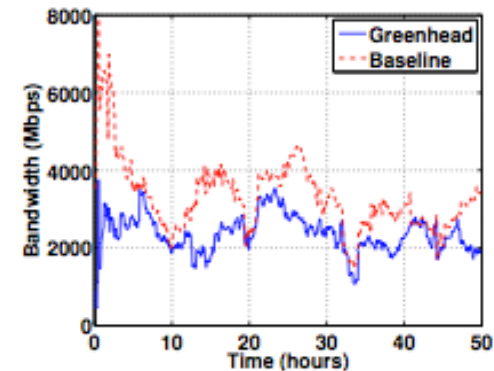
(a) Acceptance ratio



(b) Instantaneous profit



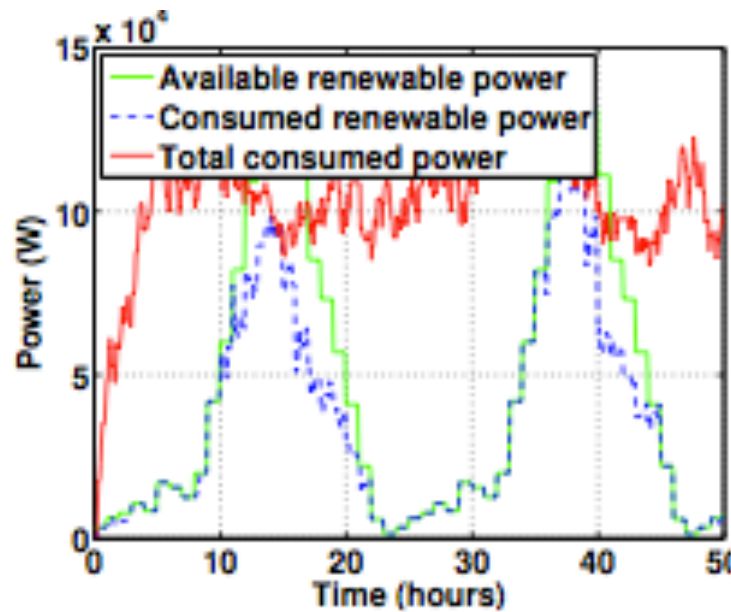
(c) Average link utilization in the backbone network



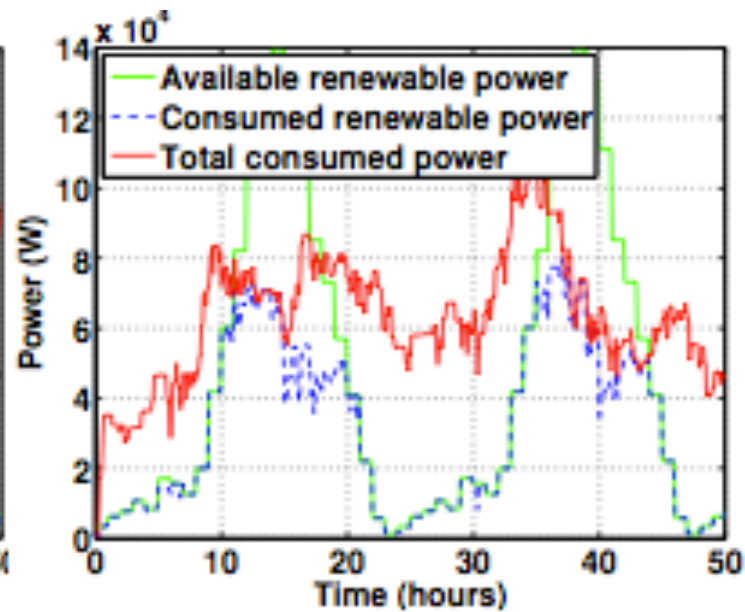
(d) Used bandwidth per request in the backbone network

- Greenhead achieves higher acceptance ratio, higher revenue and better backbone network utilization

Results: Renewable Energy



(a) Greenhead



(b) The baseline

- Greenhead maximizes the utilization of available renewable energy

Outline

- ▣ Future Application Platforms: trends and challenges
- ▣ The SAVI Project
- ▣ SAVI Smart Edge
- ▣ Sample Research Contributions
 - ▣ VDC Planner
 - ▣ Venice
 - ▣ Greenhead
- ▣ Conclusion

Summary

- Current ICT “Revolution” ?
 - Virtualization and Softwarization as enablers of Future Application Platforms
- SAVI Testbed
 - Canadian Future Application Platform, leveraging Multi-tier Clouds and SDN to provide a fully programmable research testbed
- This presentation
 - Shown how some of the challenges underlying the development of the Smart Edge SDI Manager have been addressed, namely resource management, service availability and green operations
 - VDC Planner, Venice and Greenhead operational in the SAVI Testbed

What's Next ?

- Smart Edge & SAVI Testbed 2.0:
 - Expand capacity and number of edges
 - Deploy virtualized FPGA resources
 - Spectrum + RF Frontend Virtualization
 - Develop Wireless Access Manager in Smart Edge
 - Extend Software Defined Radio and Radio over Fiber access
 - Orchestration module in SDI Manager
 - Extend Monitoring with stream processing and data analysis
 - Cloud-RAN to provide LTE using the Smart Edge
- Kaleidoscope: SAVI Demonstrator App

Kaleidoscope



SAVI Kaleidoscope Scenario



- The Blue Jays are in the final game of the 2020 World Series and they are mounting a comeback in the last inning
- Reporters/pundits report through live audio and multi-angle video streaming feeds followed by the public on their 10G iPads /Greenberries
- Fans in stadium and the city share their reactions and commentary using social networking and contribute own audio/video streams from personal devices to shared media repositories
- Using speech and video recognition, audio and video streams are tagged in real time with timestamps, location, and by subject
- Viewers anywhere create one or more customized views using latest release of the Kaleidoscope app
- As the Jays win the game, data traffic peaks. Smart Edge activates maximum number of pico-cells to provide capacity to carry and backhaul massive data streaming content wirelessly
- Local virtual circuits route data to storage in the edge and beyond

SAVI Kaleidoscope Scenario



- Smart Edge ensure that local security personnel continue to have their own secure/dedicated channels for the purpose of law enforcement
- As the crowd moves to local restaurants and bars to celebrate, monitoring in the smart edge and core platforms detect and predict decreasing volumes of traffic, and the power of pico-cells is dynamically and progressively turned off
- Stadium traffic is reduced to fit in one wavelength and power is turned down on lasers and receivers on the local backhaul
- In contrast, dormant pico-cells in restaurants and bars in the area are powered up and backhaul capacity to core datacenters is increased to accommodate the demand as fans turn to Kaleidoscope on giant screens to relive their favourite moments.

The Smart Edge provides flexible, versatile and evolvable infrastructure that can readily deploy, maintain, and retire the large-scale, possibly short-lived, distributed applications



Acknowledgments

- SAVI Partners and information @ <http://www.savinetwork.ca/>
- Students at Waterloo
 - Faten Zhani (VDC Planner)
 - Qi Zhang (Venice)
 - A. Amokrane (Greenhead)