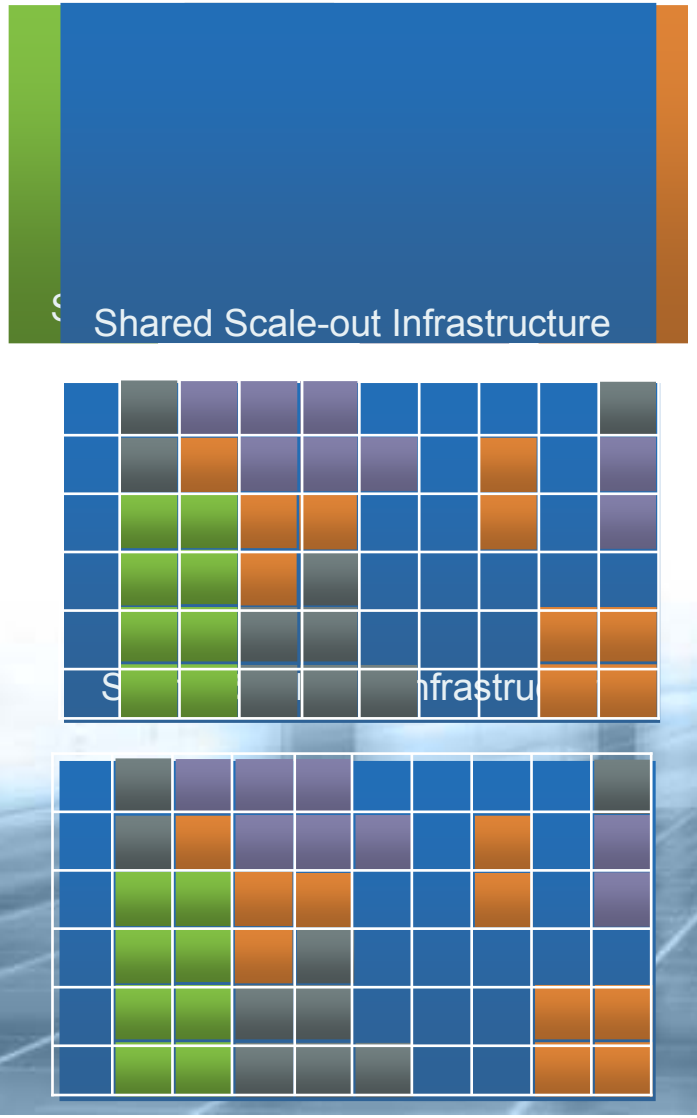
Building Scalable Ethernet Solutions

Yaron Haviv, CTO, Voltaire

September 14, 2009

Data Centers Adopt Scale-Out Models



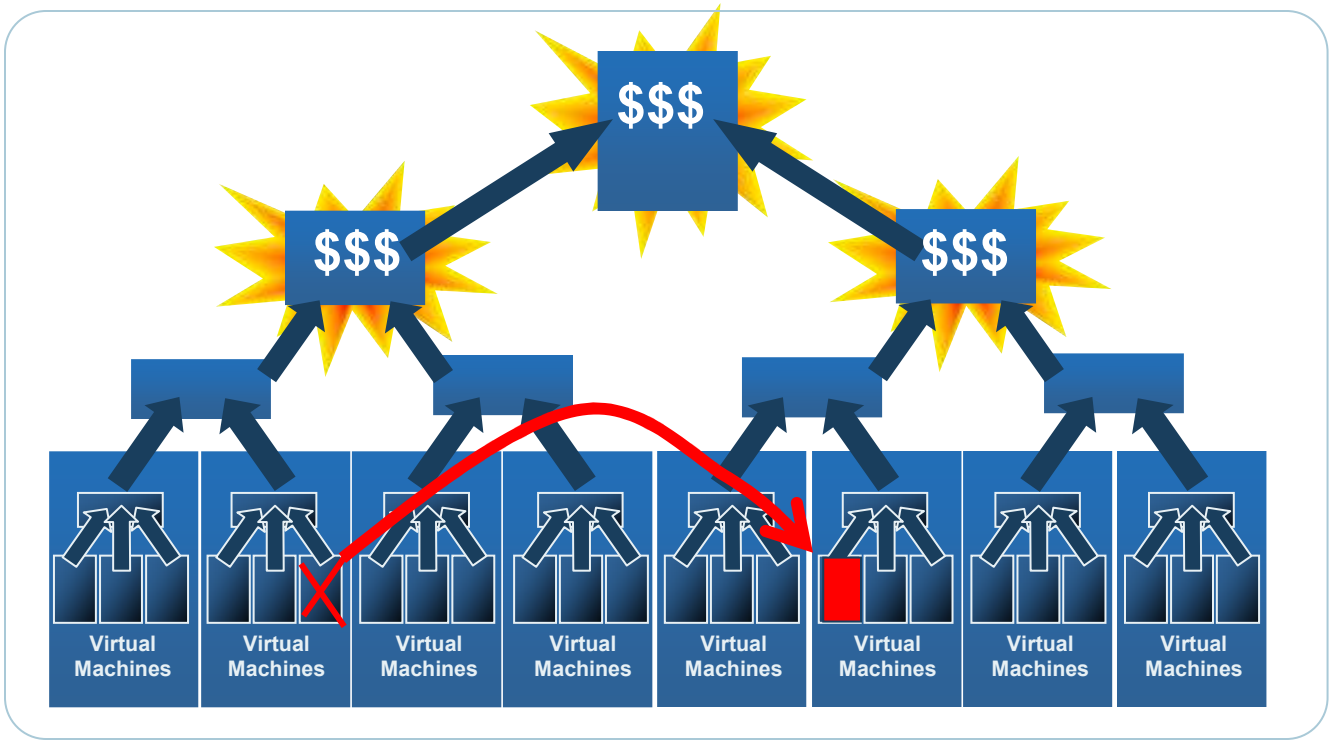
Consolidating servers to large scale fabrics

Partitioning and virtualization

Automated resource management

The Problem

Today's Network Fabrics Don't Scale



- Expensive, Power hungry core/aggregation switches
- Exploding number of switches to manage
- Bottlenecks due to increased density and inter-rack traffic
- Not addressing Virtual Machine mobility

Key Requirements From Data Center Fabric

- **High Density, Low Cost, Low Power**
  - Support many more nodes in smaller foot print and reduced budget
- **Scale-out topology and Improved fabric utilization**
  - Allow large scale L2 networks with high bisectional bandwidth (flat)
- **Multi-class (Lossless/Lossy), L2 Congestion Mng**
  - Enable lossless storage (FCoE) and IPC with traditional LAN (Lossy)
  - Mitigate congestion spreading in lossless networks (isolate, control, monitor)
- **Virtualization and Application aware**
  - Apply policies to VMs and applications vs. to physical nodes/ports
- **Low Latency**
  - Application messaging depend on lower and predictable latency
- **Central Fabric Resource Management**
  - Enable multi tenant and multi-element fabrics to be monitored and controlled

Gaining Scale, Servers and Storage Example

- **When more capacity and lower costs are required Scale-out Architectures are deployed**
  - PC clusters based on commodity elements with 100K CPUs
  - Storage clusters, Web clusters, Database clusters, ..
- **Same Overall architecture:**
  - Many simple elements interconnected
  - Automated/dynamic load distribution
  - Central resource and policy management
  - Virtualization, partitioning, and abstraction
  - Interface virtualization (look like a single system to consumers)
  - Fault management, Master/Standby, ..

Why not follow the server and storage trends, Build scale-out networking solutions !

Comparing Switch Scalability & Density: Traditional Ethernet Vs. Other Fabrics

Platform	Power/Port	Price/Port	Latency	Max wire speed ports
10GbE Edge	7-10W	$400-900	0.3 – 4 us	24-48 in 1U
10GbE Core	35-100W	$2,600-$5,000	> 10 us	140 in >20U
Difference	~5-10X	~5-10X	> 5-10X	7-14X less dense

InfiniBand 40Gb Edge	**< 5W**	**$350**	**0.1 us**	**36 x 40G in 1U**
InfiniBand 40Gb/s IB core	**7W**	**~$1,000**	**0.3 us**	**324-648 40G in 19U**

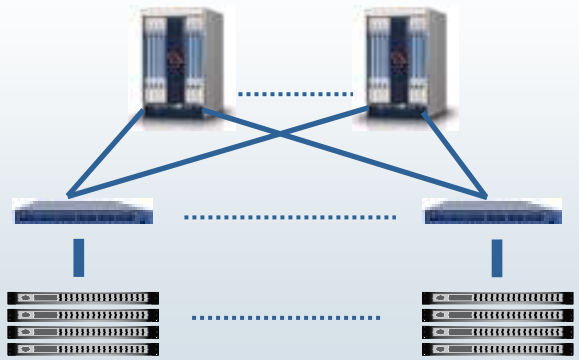
Other Fabric oriented technologies are faster, more scalable and efficient due to simpler L2 architectures

* Based on estimated list prices

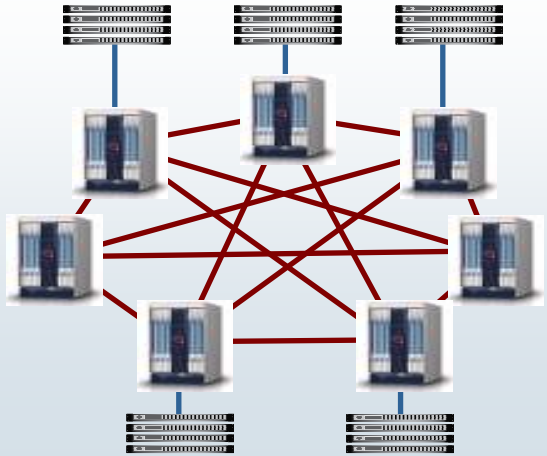
Fabrics Can Scale Linearly Beyond a Single Device

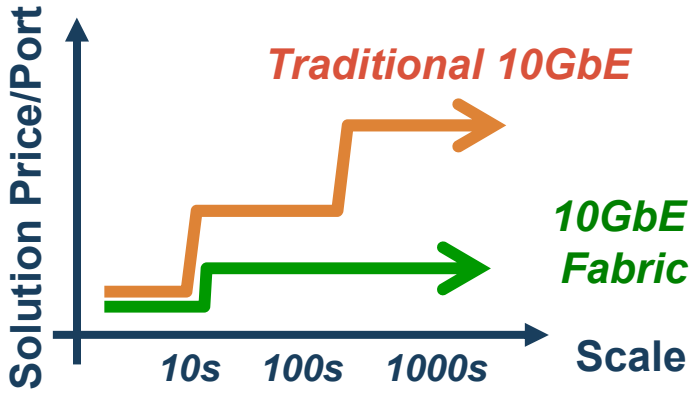
(if we bypass Spanning-tree limitations)

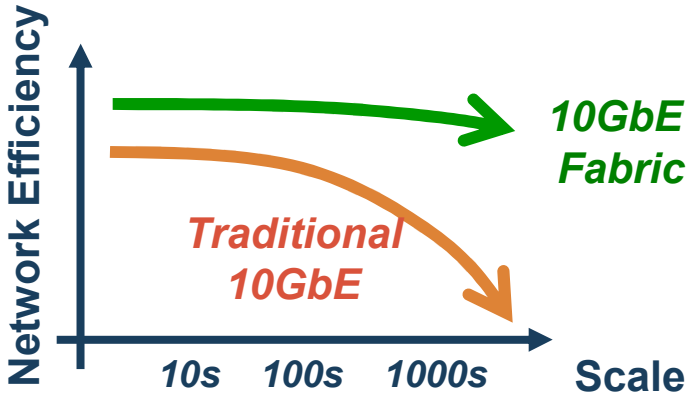
Fat-Tree Topology



Mesh/Cube Topology

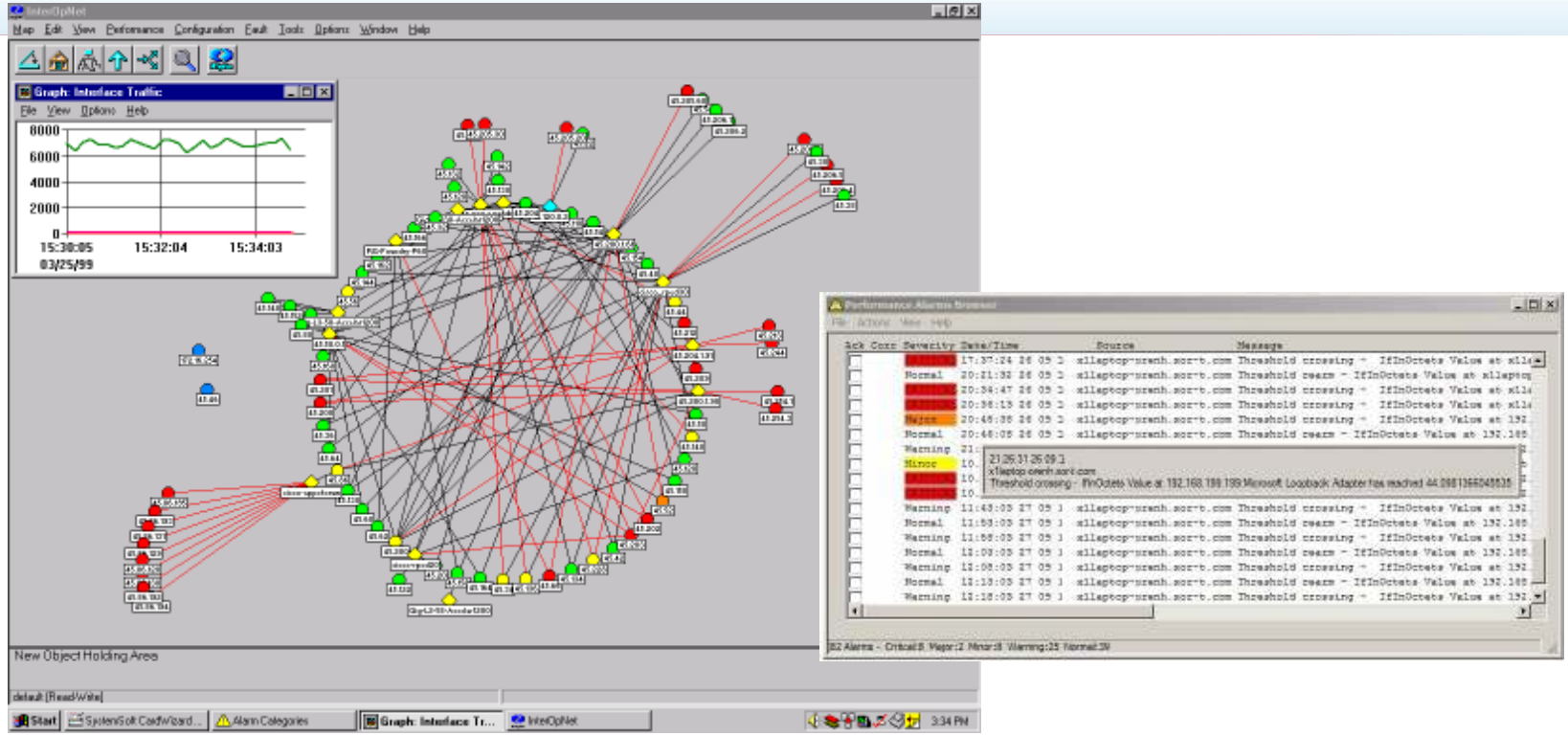






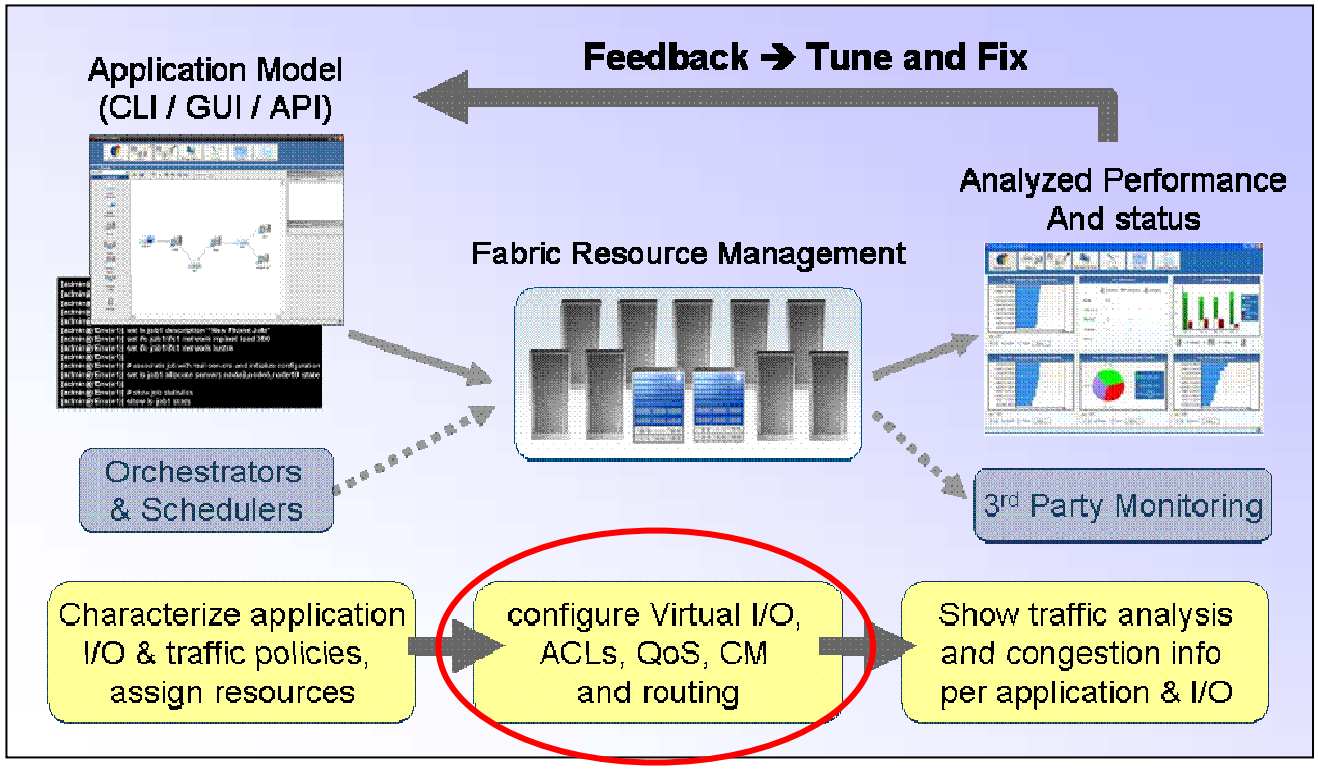
But the key in gaining scalability and efficiency is how we manage the fabric resources ….

Traditional Network Management Paradigms



- Focus on element management, or junctions (routers, ..)
- No aggregation of information to meaningful (service) data
- No notion of virtual objects, services, or resource management

The Future: Fabric as a Service (FaaS™)

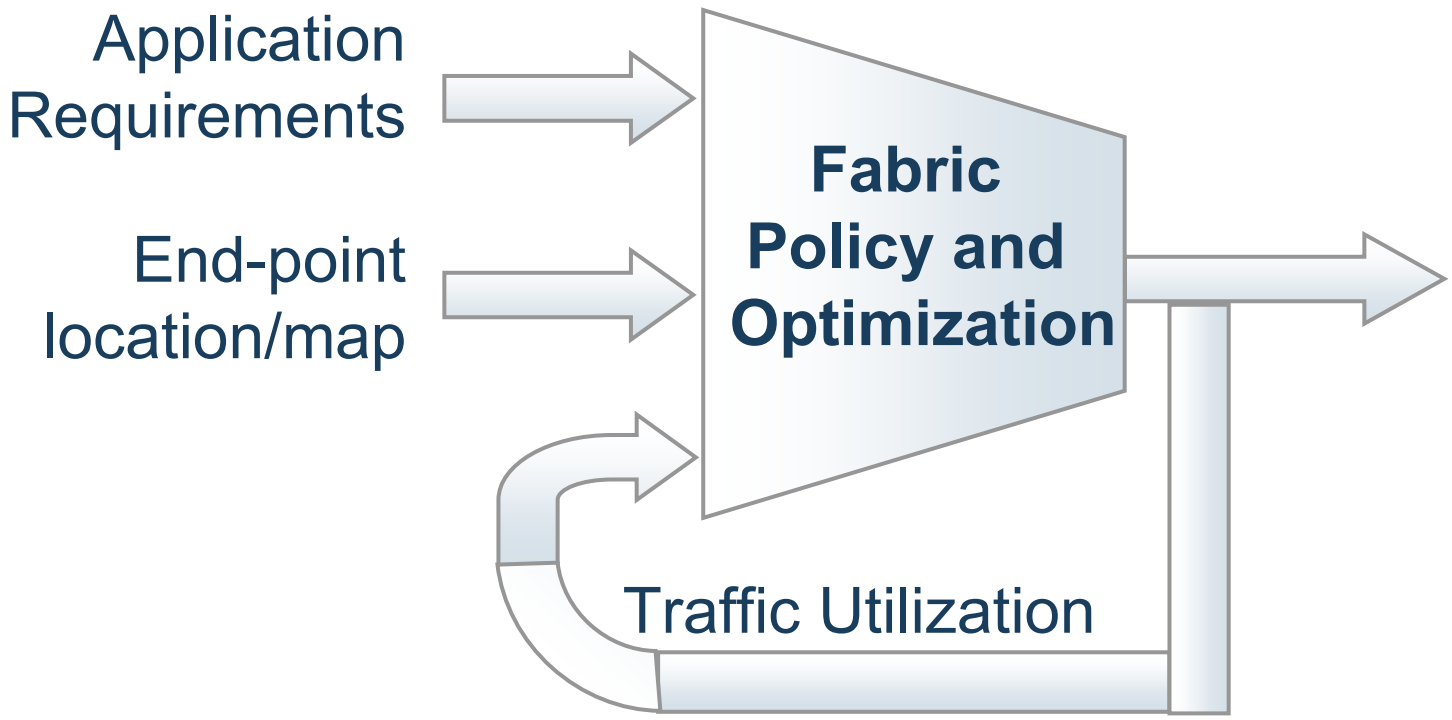


- **Application/Service driven fabric resource management**
- **Significantly Improve application performance, increase utilization, manage fabric SLA, and require less hardware**
- **Address Multi-tenancy, Virtualization, and Elasticity**

Connectivity abstraction in Fabrics

- **Define 3 key elements**
  - Virtual L2 end point (vPort/vNIC)
  - vPort groups/profiles, may even be nested (groups and sub-groups)
  - vPort/group relations
- **vPort Group examples**
  - Network (all the end-points connected to the same network/vlan)
  - Cluster/Tier (e.g. all the NICs of the web-server cluster)
- **vPort/Group relation examples**
  - Node-network (attachment point between node and network, a.k.a NIC)
  - L2 Network (all the vPorts have uniform relation)
  - Client-server (e.g. web nodes communicate with DB nodes)
  - Client-storage (e.g. servers communicate with some storage nodes)

Fabric Optimization Process
Delivering Better Application SLA With Fewer Resources

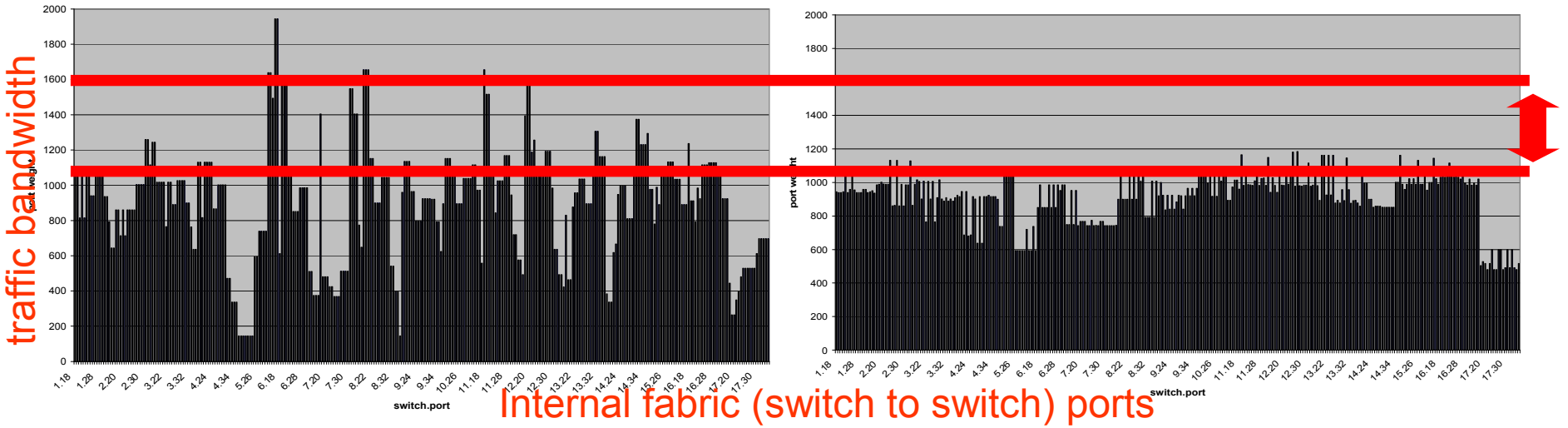


Dynamic Fabric Policy and "Workload" Management Process can provide better scalability and utilization

Fabric Optimization – Benchmark Example

Traditional Solutions
Standard Hashing

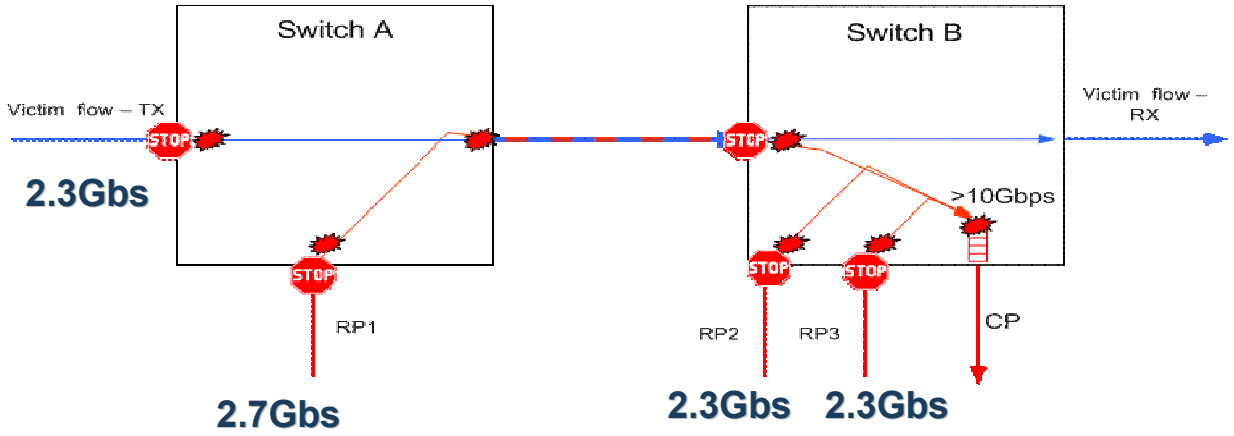
Using Fabric Optimization
Policy based Routing



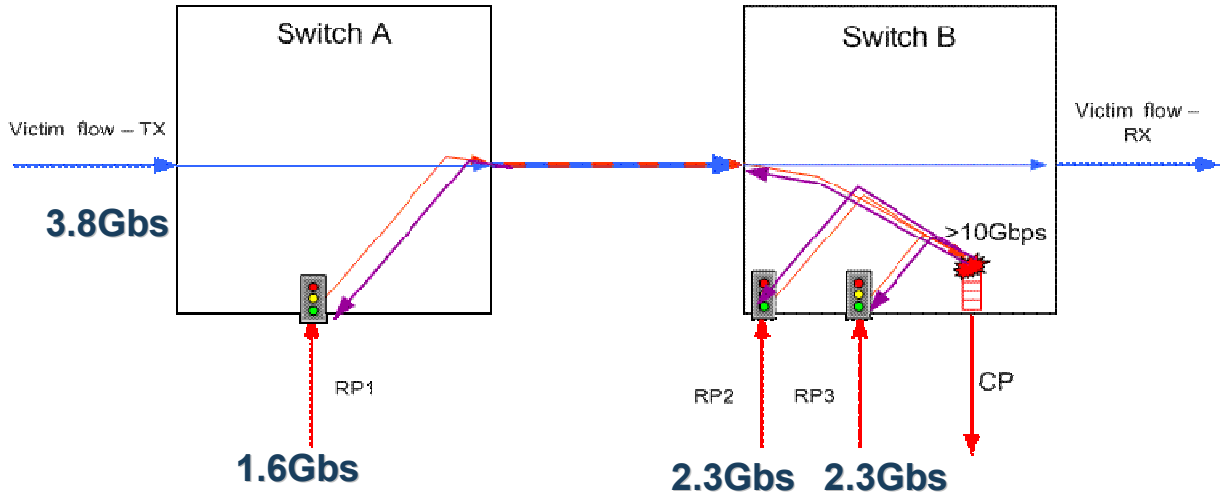
- ***Avoid link oversubscription, Require less aggregation bandwidth (less switches/cost)***
- ***Reduce delays and congestion***
- ***Application SLA (bandwidth allocated to the right apps)***

Using Dynamic Congestion Control (QCN)

Without QCN



With QCN

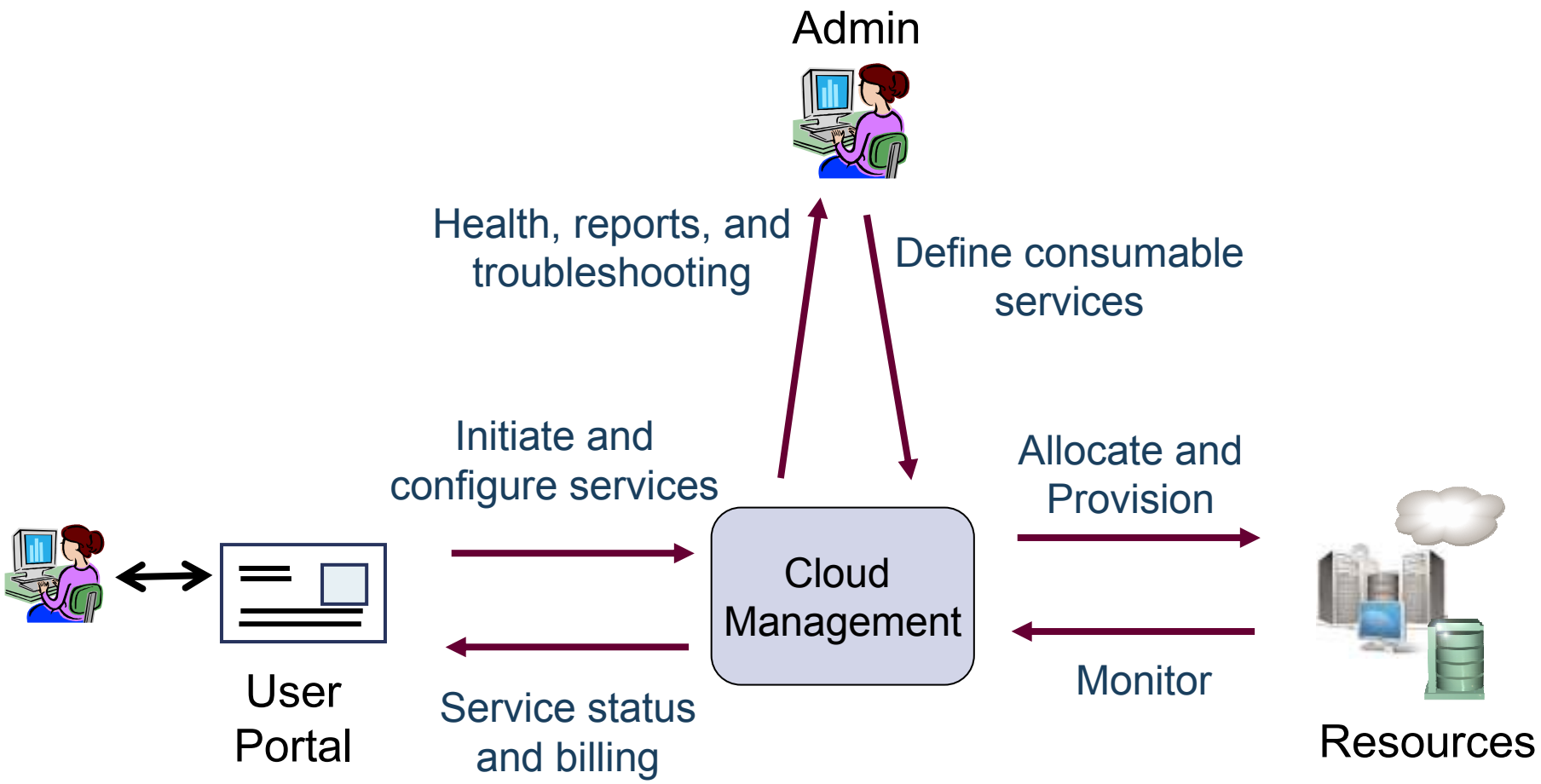


Key Elements in Cloud Solutions

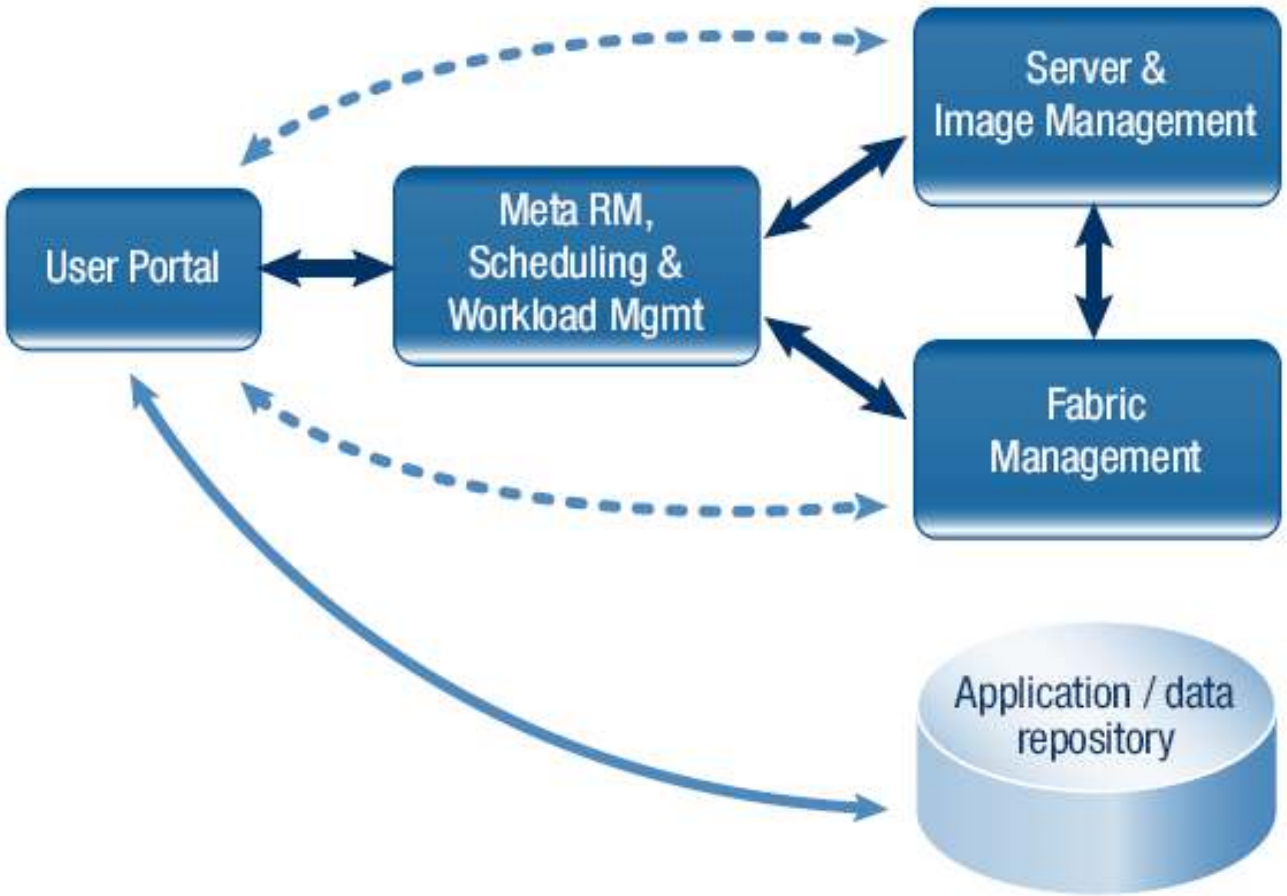
- ***Dynamic Computing Infrastructure***
  - virtualized server, storage, and network infrastructure
- ***Self-Service Based Usage Model***
  - Users must be able to autonomously deploy or use cloud services
- ***Simple to manage with minimal admin/IT intervention***
  - Automate resource and service provisioning
  - Policy driven resource management, scheduling, and fault handeling
- ***Business Service Focus***
  - *manage business services, hardware and software are resources*
- ***Consumption Based Charging or Billing***
  - Consumers pay only for the resources they use

Network Resource Management is Critical in Clouds

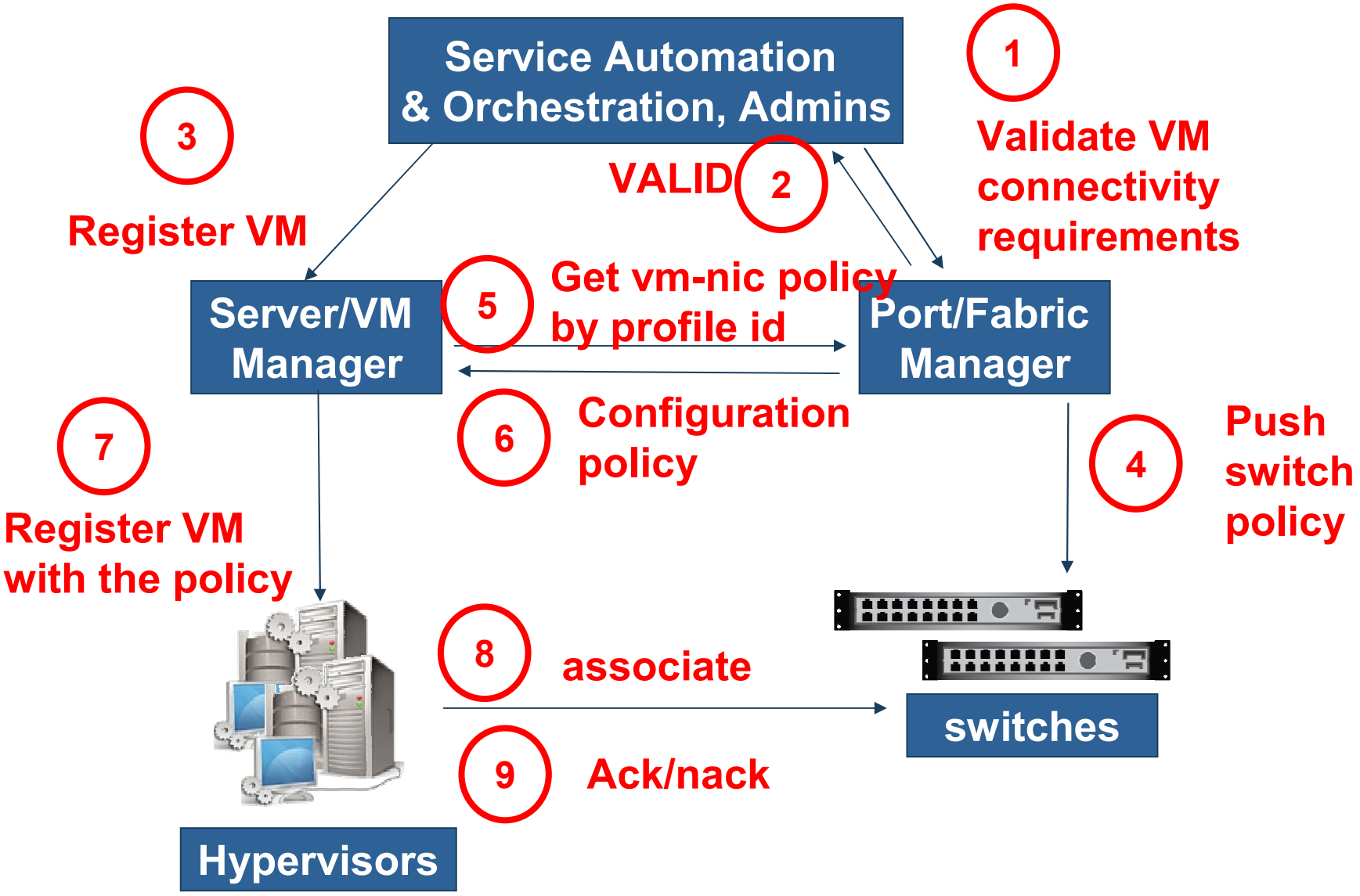
Cloud Management at a High Level



Fabric Management as Part of Cloud Management



Example: VM Initiation Process



Summary

- **New scale-out Data Center architecture require new networking infrastructure**
  - Cheaper: Based on many simple interconnected elements
  - Scalable: Eliminate spanning-tree limitations
  - View and manage fabric as a service
  - Dynamic resource management
  - Focus on virtual-end points rather than physical ports
  - Integrated with overall cloud management

Thank You

yaronh@voltaire.com

September 14, 2009