What We Talk About When We Talk About Cloud Network Performance

Best CCR of SIGCOMM 2013

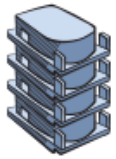
Authors: Jeffrey Mogul, Lucian Popa

Presenter: Jian Zhao

Context: Cloud Computing

- Focusing on Infrastructure-as-a-Service clouds
- Resources in IaaS clouds

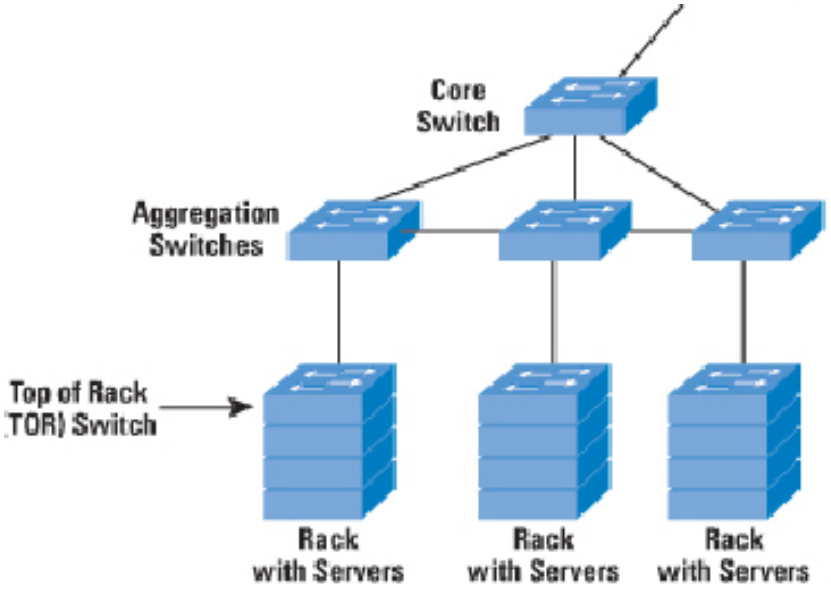
Compute
Amazon EC2
750 hours/month*



Storage
Amazon S3
5 GB*

Guaranteed CPU/RAM/DISK
in size, service level, price

No guaranteed network performance!
Best-effort globally sharing



What are the problems

- Studies have shown huge variations in application performance
  - Which are often caused by variable network performance
  - See “Towards Predictable Datacenter Networks” Ballani et.al., SIGCOMM2011
- Cloud customers want network performance guarantees
  - No network performance guarantees mean no application predictability
- The guarantee is harder than that for CPU/RAM/Disk
  - Best effort globally sharing
  - Hardware trends are unlikely to save us

Difficulties in cloud network performance guarantees

"Just give me enough bandwidth at a good price"

But:

- Where, when, and how do we measure bandwidth?
- Is bandwidth the only important metric?
- How do we set the price?
- How do we actually make this work in practice?

There are lots of ways to approach these questions:

- So not much agreement on how to structure guarantees
- And it's hard to compare research results
- Or to guide research towards useful designs

Out of scope for this paper:

- Performance to/from external (Internet) endpoints
- Performance between VMs of different tenants
- Performance between “availability zones” (AZs) or “regions”

All of which are important and challenging problems

Outline

- What kinds of properties do we want to guarantee?
- The interaction between guarantees and pricing
- Implementation issues

- What properties do we want to guarantee?

Properties for cloud network performance guarantees

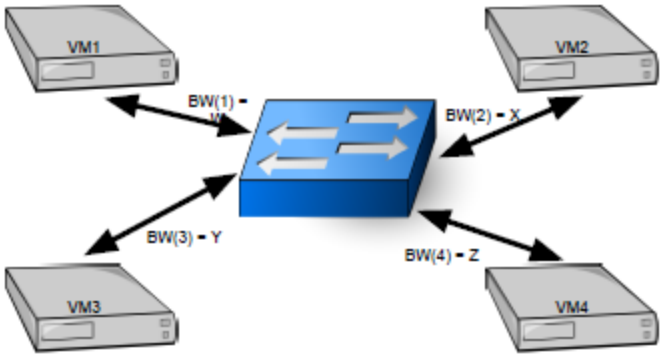
- Customer's point of view:
  - Predictable, high bandwidth
  - Predictable, low latency
  - Predictable, low loss
  - Predictable, low cost
  - Simple, flexible interface
- Provider's point of view:
  - Happy customers
  - Scalable to lots of VMs
  - Efficient implementation
  - High utilization of resources
  - Predicable profit margins
  - Simple/automated management

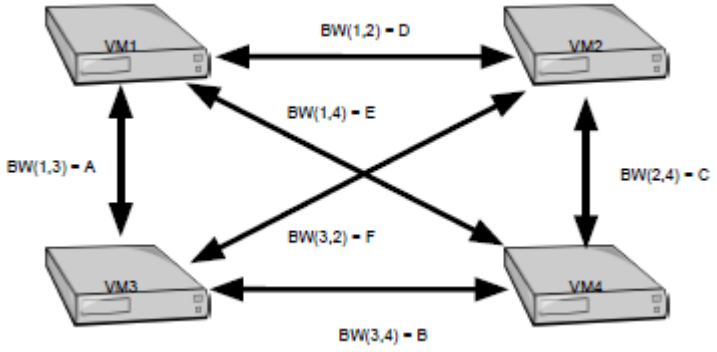
What does “guaranteed bandwidth” mean

- Between what endpoints do we measure bandwidth?
- Over what period do we measure it?
- When is the guarantee violated?

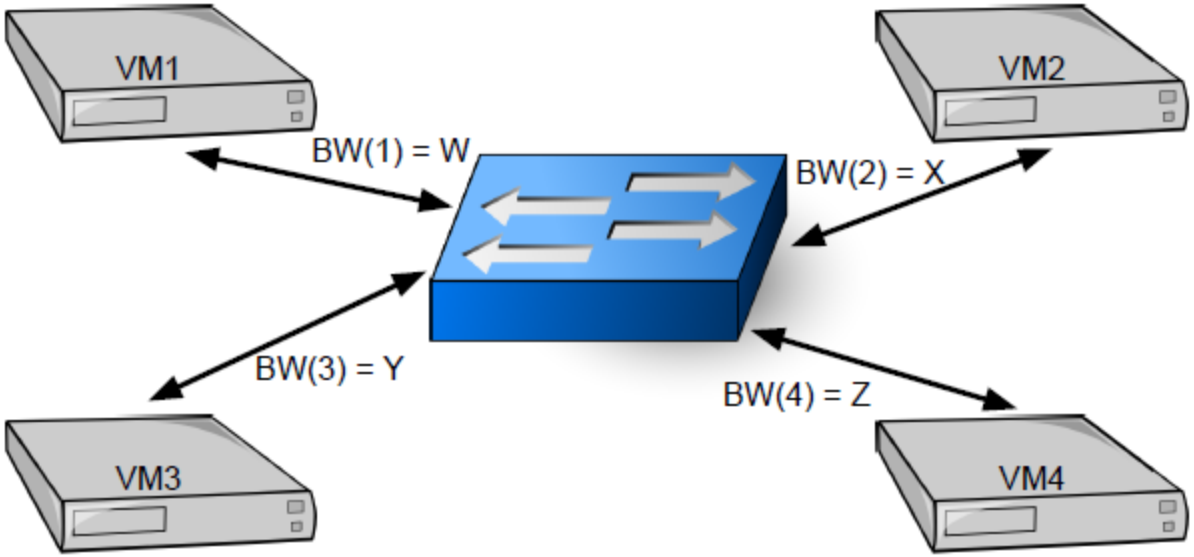
Between which endpoints: Two popular models

- Hose model
  - VMs all connected via one abstract “big switch”
  - Bandwidth guaranteed between switch and VMs
- Pipe model
  - Bandwidth guarantees between pairs of VMs



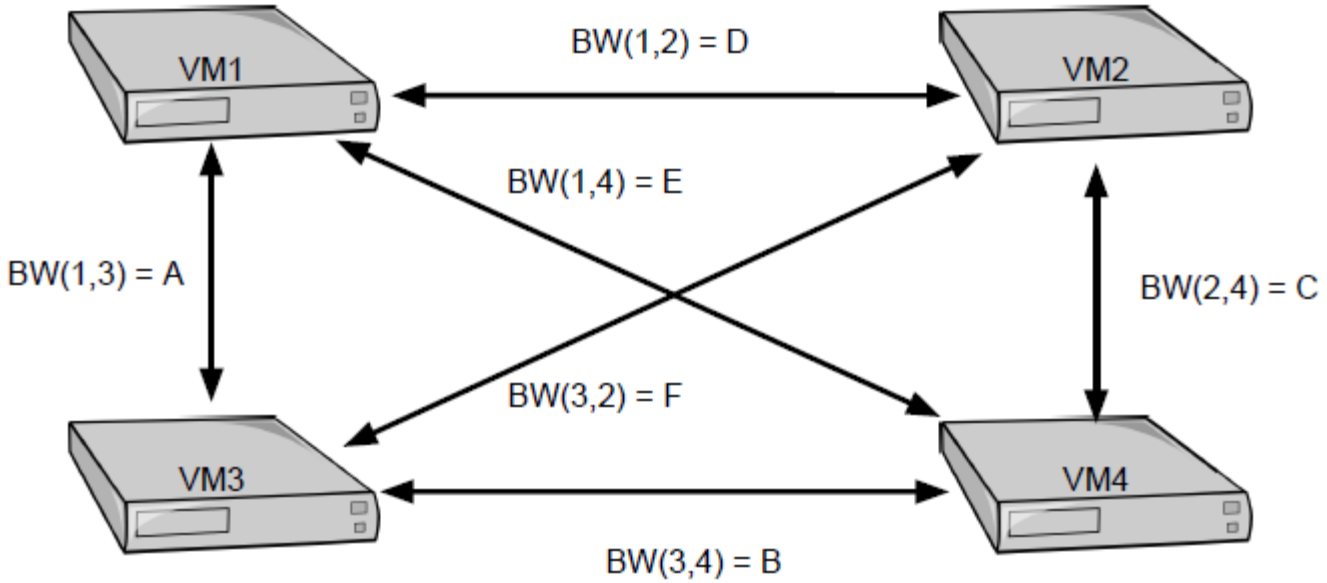


Hose model



- Pros & cons:
  + Simple abstraction, matches “real world” provisioning
  + Easy to specify: one value/VM (or 2, for bidirectional)
  - May force over-provisioning of underlying real resources

Pipe model



- Pros & cons

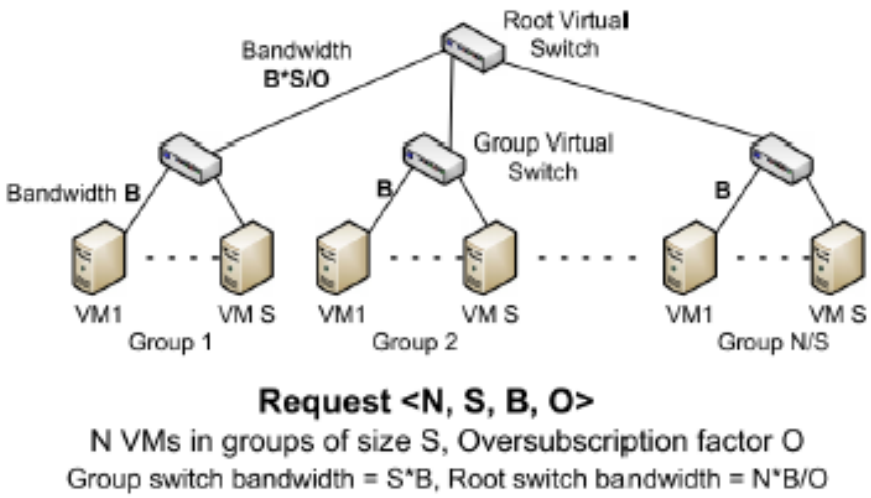
  +Captures actual inter-VM requirements

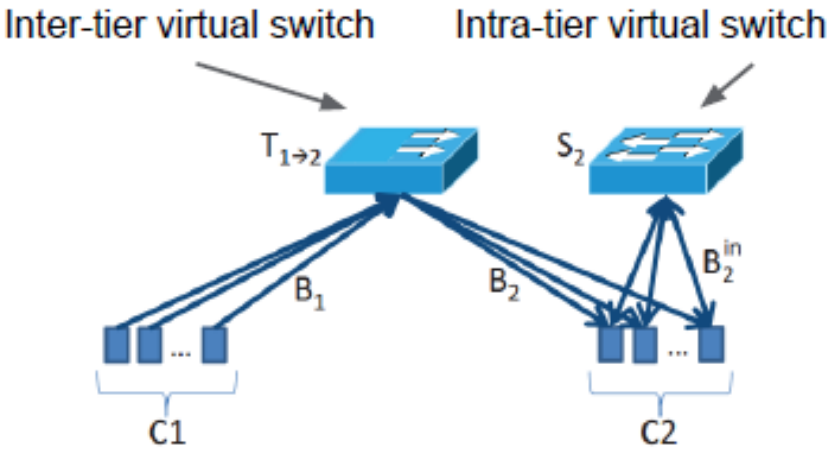
  Effectively, the inter-VM traffic matrix

  -Requires O(N^2) parameters (vs. O(N) for hose model)

Variations on the hose model

- Hierarchical hose model
  - E.g., “Virtual Oversubscribed Cluster” Oktopus,SIGCOMM2011
- Tenant Application Graph
  - CloudMirror,HotCloud13





How to handle dynamic needs

- Bandwidth demands aren't static
  - Workloads vary over time
    - Predictably, over long periods, e.g., daily/weekly cycles
    - Predictably, over short periods, e.g., phases of MapReduce jobs
    - Unpredictably, e.g., flash crowds
  - Cloud customers add/remove VMs
- Some possible approaches
  - Proteus(SIGCOMM'12) suggests scheduling MapReduce jobs so as to interleave their high-bandwidth phases
  - CloudMirror(HotCloud'13) adapts to changes in #of VMs at each tier

What are we measuring

- We could measure/guarantee
  - Mean bandwidth over a given period P
  - Peak bandwidth: worst case result over period P is bounded
  - “Tail latency” (99.99%ile latency)
  - Loss rate
- Different applications will require different measure
  - Batch jobs: mean bandwidth is probably OK
  - Interactive applications: need bounds on tail latency or flow completion time

What properties do we want to guarantee?

- Summing up
  - Mean bandwidth vs. peak bandwidth vs. latency
  - 100% guarantees vs. probabilistic guarantees
  - Models of communication patterns (pipe/hose/VOC/etc.)
    - The pipe model is the most expressive, and also complex
  - How to handle time-varying needs
  - How to support all of these without wasting resources

Cloud providers and customers need to agree on what matters

- How do guarantees interact with pricing?

Differences with Internet

- Money flows from customers (tenants) to providers
  - Accounting for network use within a cloud data-center is quite different from the accounting arrangements on the Internet
  - Allocation of resources can follow the money more closely
- Be careful about transferring ideas about desirable properties of a network from our experience with the public Internet, especially:
  - Fairness
  - Work-conservation

Does fairness matters

- Fairness is one way to allocate resources among competing needs
  - In the absence of other allocation mechanisms, it's not bad
  - Between the flows of one application, fairness might be important
- Fairness between cloud tenants does not matter
- FairCloud(SIGCOMM'12) showed you can't simultaneously have both fairness and minimum bandwidth guarantees

What about spare bandwidth

- We typically like work-conserving systems
  - Otherwise, some capacity goes to waste
  - But these are not fully predictable: what you get tomorrow might not be what you got today
- Cloud providers do not want to give away spare BW for free
  - Otherwise they risk training their customers to expect this
- A possible solution:
  - Build a work-conserving system
  - Bill customers for spare BW, at a reduced rate

Pricing of multiple resources

- Ballani et.al. (HotNets'11) point out problems with simplistic pricing
  - VM time can be wasted waiting for slow network paths
  - Simply adding a bandwidth cost to the VM cost is a bad idea
    - Sneaky provider could stall network to increase VM hours billed per job
- They propose "Dominant Resource Pricing":
  - You pay only for the resource (CPU or net) you are using more of
    - If network is the bottleneck, don't pay for idle CPU time
    - If CPU is the bottleneck, don't pay for idle network bandwidth

Implementation issues

- Hard to implement
  - Hose model guarantees in an oversubscribed network
    - The “one big virtual switch” isn’t non-blocking
    - Virtual Oversubscribed Cluster does this with 2-level hose model + careful VM placement
  - Scalable pipe model guarantees
  - Hose model guarantees + fair work conserving allocation

Where are we headed

- The evidence suggests that we have not entirely converged
  - Few cloud providers offer bandwidth guarantees
  - Customers may not really understand how to request/use them
  - Efficient, scalable implementation is still an open problem
  - Technical solutions cannot be independent of pricing model

Promising areas for future research

- Can clouds support guarantees for network latency?
  - Especially for (say) 99.9%ile latency
- How do we deal with changing workloads
  - Including “flexing” of the number of active VMs
- Some challenges
  - Cloud providers are unlikely to release packet traces
  - True infrastructure costs (HW/SW/power/people) are secret

Thanks very much!