Dependability in Hybrid Clouds: Practitioner Insights

IFIP 10.4 Work Group Meeting – Winter 2018, Goa, India

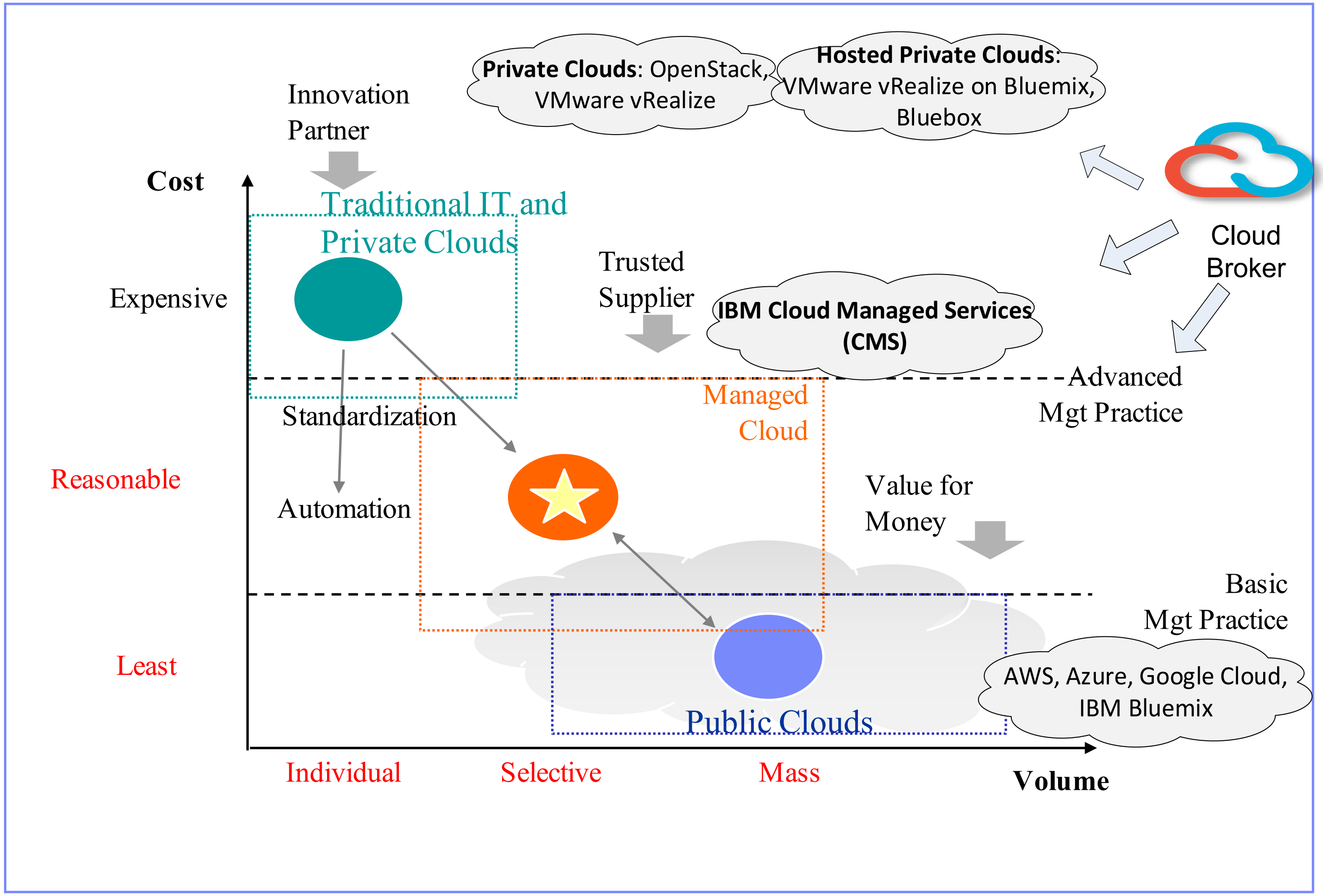
Sreekrishnan Venkateswaran,
Distinguished Engineer, IBM Cloud Center of Excellence

Contents

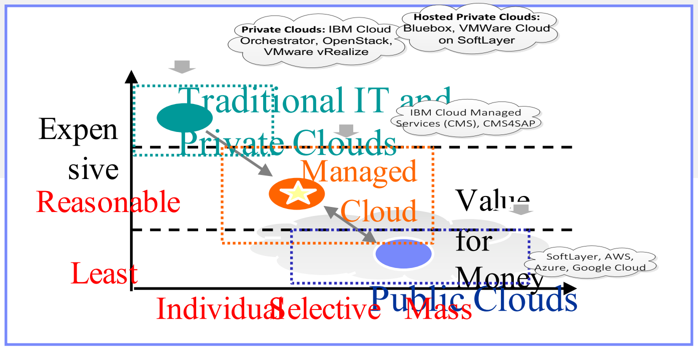
- Today's Hybrid Cloud Landscape & Dependability NFRs
- Availability Fulfilment Approach: Statistics from 50 client deals
- Case Studies from the field:
  - **I:** Engineering "Just Enough" HA on a Private Cloud
  - **II:** Hosting a Clustered Appliance on a Public Cloud
  - **III:** Application/MW HA on a Shared Private Cloud
- Concluding Thoughts: Trends on Dependability on Hybrid Clouds

Cloud Landscape: Building Blocks for Hybrid Hosting in Today's Client Deals

- IT Hosting philosophies plotted across 2 axes – cost & volume
- One end of the spectrum lie high-volume low-cost public clouds
- Other end of the spectrum is the low-volume high-cost single-tenant environments, cloud or legacy
- For enterprise clients, there is a sweet spot in this landscape in terms of price and services via a managed enterprise-grade multi-tenant cloud
- Value close to traditional/private IT by providing management above the hypervisor, enhanced isolation & production SLAs
- Price points close to that of public clouds via standardization, virtualization and automation



Availability Capabilities Across Cloud Categories



Cloud Category	Availability Philosophy
○ On-premise Private ○ Hosted Private ○ Traditional IT	○ Custom Design → **Example: Case Study 1**
Public Clouds	○ Provider offers VM-level availability SLAs ○ Provider offers IaaS-level HA on Bare metal → **Example: Case Study 2**
Managed Multi-tenant (or "shared private") Clouds	In addition to OS-level availability, introduces clustering to provide a more highly available environment → **Example: Case Study 3** ○ HA clusters by allowing customers to specify anti-collocation of the virtual workload onto separate servers for fault containment ○ Connects clusters to shared storage for shared-disk HA topologies.

Non-Functional Requirements in Cloud Deals: A Recent Example

NFR#	Category	Requirement	Deliverables
NFR01	Availability	Cloud management software should be highly available	OpenStack and Virtualization component will have active-active configuration. Handle Server overload v/s Server going down.
NFR02	Availability	Hardware Management	Workload running on a failed host will be restarted on another host in the resource pool for both AIX and VMware if the host fails
NFR03	Business Continuity	Backup & Restore	Support backup policies pertaining to NetBackup
NFR04	Monitoring & Event Mgt	Host monitoring required Guest monitoring required (Managed) Dashboards - utilization monitoring	Monitor both hosts and guests, OS agents to be initially deployed manually during post-provisioning.
NFR05	Image Management	Standard Images and patterns to be maintained	Standard image catalogue will be maintained Application patterns to be created Manage a standardized catalogue of patterns
NFR06	Security	Follow client's security guidelines	TBD
NFR07	Disaster Recovery	RPO of 30 minutes, RTO of 4 hours	Support failover on DR-sensitive workloads within RTO/RPO
NFR08	Security	Network Isolation	Segregation using VLANs (for Lpars) and VxLANs (for x-86 optional)

SLA & SLO Requirements in Cloud Deals: Recent Example

- Availability SLA: 99.5 to 99.9
- Provisioning Request Fulfilment SLO: 15 mins to 24 hours
- DR SLA: RPO/RTO: 15m/4h
- Incident resolution SLO: See table
- On-boarding time SLO: 1 day
- Time to build Pod: 8 months

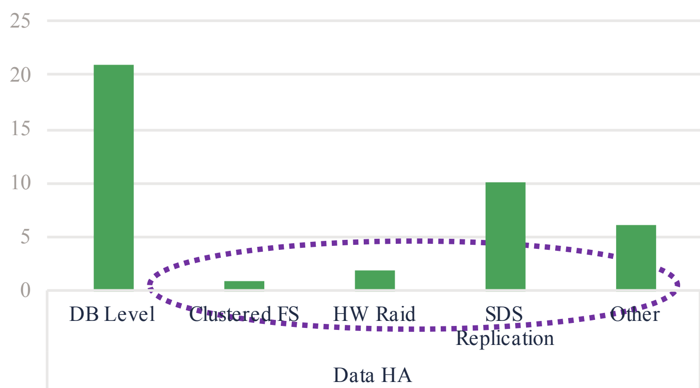
Instance Type	Provisioning Time SLOs
Bronze	15 minutes
Silver	30 Minutes
Gold	60 minutes
Platinum	24 hours

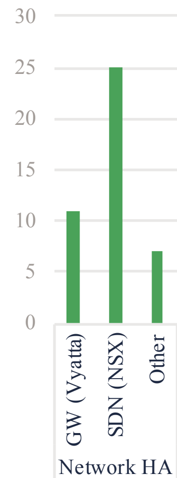
Incident Resolution SLAs

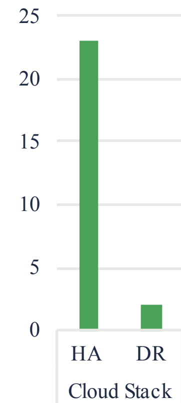
• **Severity**
• Severity 1 – Resolution time 90% within 4 hours
• Severity 2- Resolution time 90% within 24 hours
• Severity 3 – **Response** time 7 calendar days
• Severity 4 – **Response** time 30 calendar days •

Availability Solution Approach Data from Cloud Deals: Recent Statistics (1/2)

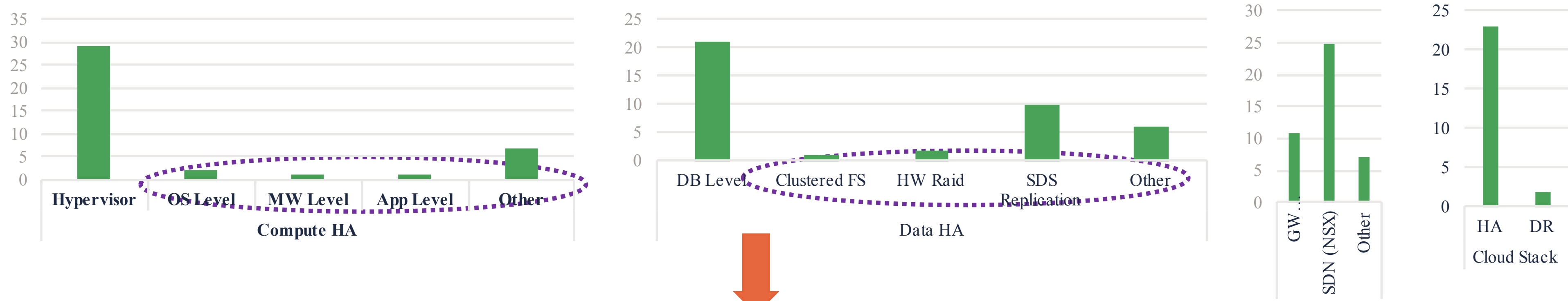
	No HA	Compute HA				Data HA				Network HA		Cloud Stack	
												HA	DR
# Deals (out of 50)	10	Hypervisor-level	OS Cluster	MW (e.g. DB Server)	App-level	DB-level	Clustered File systems	HW RAID	SDS Replicas	GW	SDN	23	2
		29	2	2	1	25	1	2	10	11	25		







Availability Solution Approach Data from Cloud Deals: Recent Statistics (2/2)

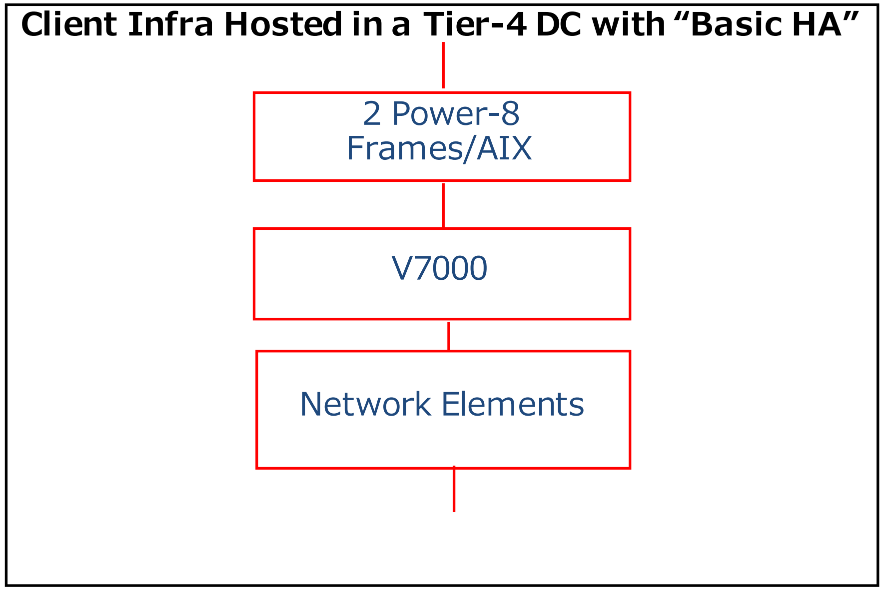


#	Conclusions	Improvement Strategy
1	Most solutions do not map SLA requirements to the level of HA needed across constituent components. They merely follow rules of thumb such as "triple replicate storage since this airline client needs 99.95"	Simple uptime modeling. Focus of Field Case Study 1.
2	For managed cloud deals, the managed service provider usually offers only 2 system availability options, one corresponding to HA & another to non-HA. How to convince managed service provider teams to offer HA required by the client?	

Cloud

Problem Statement

1. A global client required a 99.90 uptime SLA (9h/y downtime) at OS level
2. But the managed Service Provider (MSP) offered only an uptime SLA of 99.0 (3d/y downtime) by default
3. However, the MSP allowed 99.5 (2d/y, 7m/d down) SLA on client IaaS enabled with "Basic HA":
   - Tier-4 Data Center (99.995 at site-level)
   - Storage V7K with Redundant HVACs
   - System-P/AIX
   - 24x7 Hands & Feet in DC

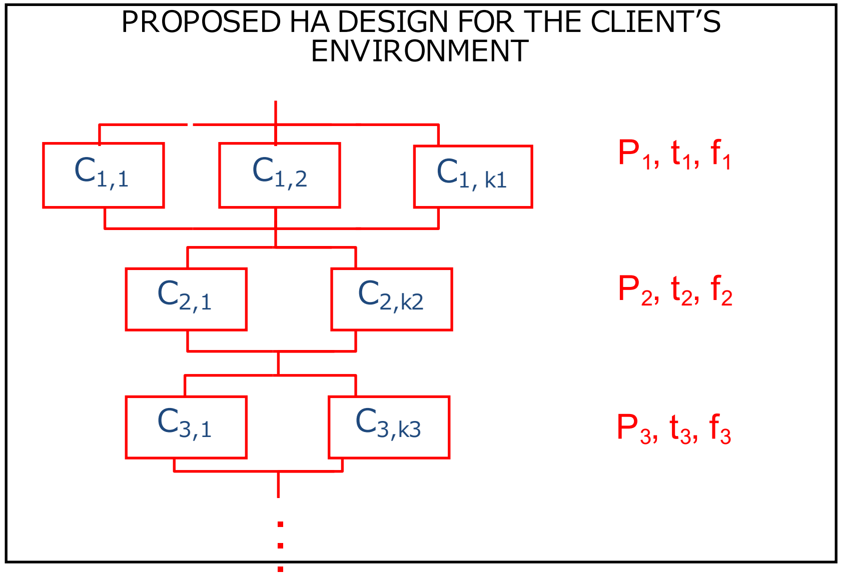


Solution Approach that was Followed

1. Assume that the managed service provider offers only 99.0 with "Basic HA"
2. Engineer additional HA on the IaaS
3. Model the ensuing redundancy and establish to the MSP team that the additional HA can increase the SLA from 99.0 to 99.95 (4h/y) without additional risk to the MSP

Additional HA Engineered in the Proposed System

The following is the proposed design for the client (in the primary data center):



$C_{1,*}$ = Compute
$C_{1,1}$ = System-P Frame-1
$C_{1,2}$ = System-P Frame-2
$C_{1,3}$ = System-P Frame-3 (Redundant node: Power-HA)

$C_{2,*}$ = Storage
$C_{2,1}$ = V7000 Storage Volumes
$C_{2,2}$ = Redundant RAID-10

$C_{3,*}$ = Network
$C_{3,1}$ = Network Elements
$C_{3,2}$ = Switches/Routers/FWs/CPEs in dual redundant mode

We model a cloud-hosted system S as a serial combination of n clusters. Let there be 'n' clusters that constitute the system. Let each cluster C_i be composed of K_i nodes, each denoted as $C_{i,ki}$.

Overall down-time probability of S can be expressed as

$D_s = B_s + F_s$ where ------ [1]

B_s = System downtime due to non-recoverable failures (breakdown of one or more clusters) and

F_s = System downtime due to recoverable failures (outage when clusters recover from node failures)

B_s and F_s are mutually exclusive if we disregard the possibility of an unrecoverable failure during cluster failover

Uptime Modeling [1/3]

P_i = Probability that a node in cluster Ci is down (= 1% from MSP's assumption that 99% can be offered without HA)

f_i = Average yearly failures for component Ci (from cloud broker data lakes)

t_i = Failover latency with the chosen HA algorithm (from empirical observations)

$ḱ_i < K_i$ = maximum number of failed nodes that can be tolerated by the clustering algorithm of C_i

If the level of redundancy in a cluster is N+ṅ, then $ḱ_i$ is ṅ.

Probability that Cluster C_i is UP = $\sum_{j=K_i-ḱ_i}^{K_i} \binom{K_i}{j} (1-Pi)^j {P_i}^{K_i-j}$

Probability that all Clusters in the system are up = $\prod_{(i=1 \text{ to } n)} [\sum_{j=K_i-ḱ_i}^{K_i} \binom{K_i}{j} (1-Pi)^j {P_i}^{K_i-j}]$

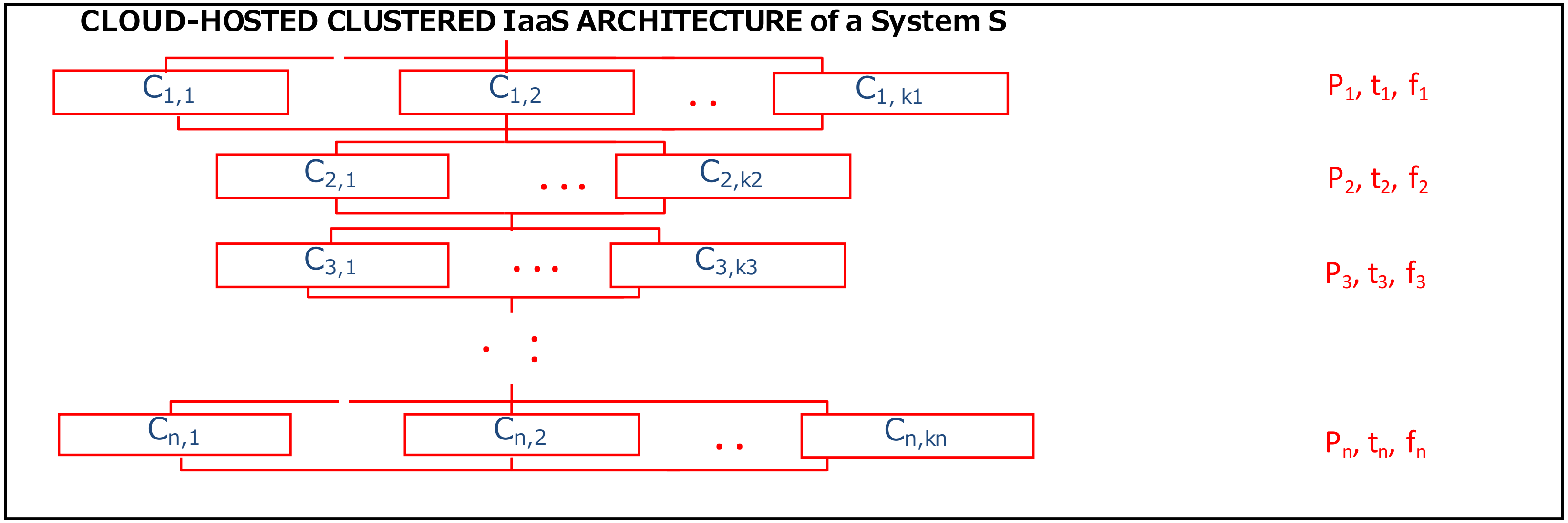
Downtime probability of System S, B_s = 1 - $\prod_{(i=1 \text{ to } n)} [\sum_{j=K_i-ḱ_i}^{K_i} \binom{K_i}{j} (1-Pi)^j {P_i}^{K_i-j}]$ ------- [2]

Uptime Modeling: Calculating D_s [2/3]

Let t_i be the time (in minutes) to failover if a node in cluster C_i goes down. Let f_i be the average number of failures experienced by a node in cluster C_i in a year.

Failover time t_i is a sum of **(i)** Time to detect that the currently active node in cluster Ci is down; this is the time before which a heartbeat miss is detected **(ii)** Time to bring up the failover node if it is on standby and **(iii)** Time for the failover node to take over from the primary node

Since P_i is the probability of a node in Cluster C_i is down, it is also the probability that the currently active node in Cluster C_i is down.
Downtime due to failover transactions in Cluster $C_i = f_i * t_i$



Uptime Modeling: Calculating D_s [3/3]

However, multiple clusters might be simultaneously experiencing failover transactions, so that time cannot be double counted.

Downtime due to failover transactions in cluster C_i, when no other clusters are experiencing failover transactions = $f_i * t_i * (K_i - \acute{k}_i) * P_i\,(X1)$, where X1 is the event that only cluster C_i in the system is experiencing a failover event and $P_i(X1) = \prod_{(j=1 \text{ to } n,\ j <> i)} [(1-P_j)^{(Kj - \acute{k}j)}]$

Note: We ignore the error of counting intra-cluster node failover times when more than $\acute{k}_i$ nodes in C_i fail simultaneously. We also disregard the possibility of an unrecoverable error during cluster failover.

Thus, Downtime due to failover transaction in Cluster C_i when there are no other simultaneous failovers in any other cluster= $f_i * t_i * (K_i - \acute{k}_i) \prod_{(j=1 \text{ to } n,\ j <> i)} [(1-P_j)^{(Kj - \acute{k}j)}]$ minutes

Downtime due to failover transactions across all Clusters = $\sum_{(i=1 \text{ to } n)} f_i * t_i * (K_i - \acute{k}_i) * P_i(X1)$ minutes

Downtime probability due to all failover transactions across clusters

$F_s = (\sum_{(i=1 \text{ to } n)} ((f_i * t_i * (K_i - \acute{k}_i))/525600) * \prod_{(j=1 \text{ to } n,\ j <> i)} [(1-P_j)^{(Kj - \acute{k}j)}])$ ----- [3]

Applying Equation [2] and Equation [3] to Equation [1], we get

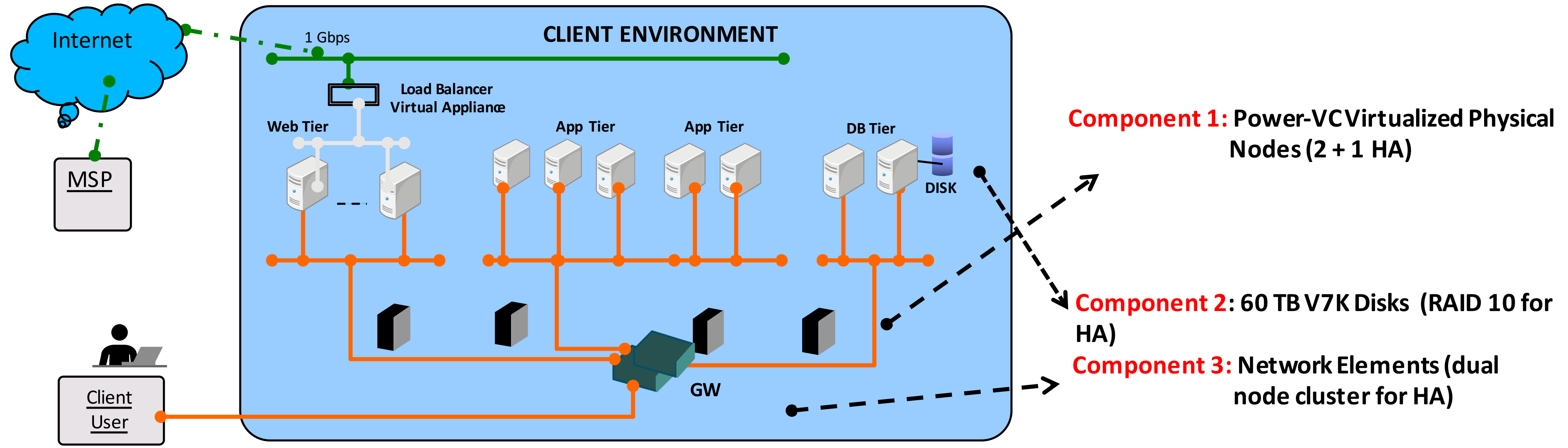
Total down time probability

$$\mathbf{D_s = (1 - \prod_{(i=1 \text{ to } n)} [\sum_{j=K_i-\acute{k}_i}^{K_i} \binom{K_i}{j} (1 - Pi)^j {P_i}^{K_i - j}]) +}$$

$$\mathbf{(\sum_{(i=1 \text{ to } n)} ((f_i * t_i * (K_i - \acute{k}_i))/525600) * \prod_{(j=1 \text{ to } n,\ j <> i)} [(1-P_j)^{(Kj - \acute{k}j)}])} \quad \text{----- [4]}$$

Total Uptime Probability $\mathbf{U_s = 1 - D_s}$ ----- [5]

Estimating New Uptime with the Additionally Engineered HA



Component #	Uptime without HA (P_i) Estimates, given MSP's posture	Average yearly failures (f_i) (Estimates from a cloud broker)	Failover latency in HA mode (t_i) (Empirical, from experience)	Proposed HA method	System Uptime with this architecture (U_s) (Applying Equation #5)
1	99% (3d/y downtime)	1	30 minutes	Power HA (2+1)	99.945% = 9m/y downtime
2	99%	2	10 seconds	RAID 10	
3	99%	1	10 seconds	Dual Node Cluster	

Cloud

Problem Statement

1. A global e-commerce retailer who wished to adopt a "Cloud-First" strategy ran specialized HA appliances (Oracle RAC in this case) in the backend for HA ("below the shopping cart"). Migrating the client to public cloud was ruled out without RAC support
2. RAC Clustering of Oracle database servers is not cloud-friendly and hence is not supported on main stream public clouds because
   - Dedicated network links between cluster servers is needed to carry heart beats. Missed heart beats can generate false negatives and can be disastrous
   - Layer-2 adjacency required between cluster servers
   - Redundant storage needed

Solution Approach that was Followed

1. Un-bond the 2 public cloud interfaces on physical bare metal hosts. Provision heartbeat VLANs on this interface
2. Provisioning bare metal servers on the same physical rack implies they are L2 adjacent since switches in a rack are trunked (link aggregation). If cluster servers fall on different racks open a ticket to get the corresponding switches trunked
3. Use redundant SDS storage (VSAN or Ceph)
4. Made this a "pseudo standard" building pattern in certain public cloud data centers

Problem Statement

1. A research facility in Mexico offers “classroom virtualization”, which is e-delivery of courses over the public Internet
2. A highly scalable web service is connected to a scalable application tier and a database tier (IBM DB2 in this case), all 3 tiers hosted on a “shared private” managed cloud. The DB tier needs to operate in HA mode
3. In general, DB HA will be of limited value at a VM-level since both database server instances could end up on the same physical host

Solution Approach that was Followed

1. Database servers (IBM DB2) were deployed in HA mode in anti-collocated VMs.
2. The database itself was hosted on a shared disk
3. The clustered database could thus tolerate single physical host failures

Concluding Thoughts: Trends on Dependability in Hybrid Clouds

- In the "Cloud First" hosting model that enterprises are increasingly adopting, workload dependability fulfilment is more about solution composition than about engineering effort if target hosting is on modern-day public and "shared private" clouds.
- On private clouds, dependability translates to redundancy engineering, but math modeling is almost always needed for a "just enough" design that maps as close as possible to SLAs
- There are also "non technical" aspects that influence. dependability, such as the standard SLA catalogues of managed service providers and application support teams.

Summary

We discussed:

- Today's hybrid cloud landscape & dependability NFRs
- Availability fulfilment approach statistics from 50 client deals
- Three client case studies where required dependability was improvised/engineered in the face of constraints on the cloud
- Trends on dependability in hybrid clouds


Questions ?!?
MICHEL