Season III, Challenge III

Project : let's survive even if [almost] everything collapses on our new planet

Focus Area : can we plan disasters properly ?

Created by : Lubomir Zvolensky (lubomir@zvolensky.sk)

Contents

1. Executive summary

What an achievement - we already live on Mars !! Unbelievable success of mankind. After thousands of issues we needed to swim through, there is just a small nimble one remaining : what are we going to do when everything goes havoc in our new home ? We need to set up some shield that will keep us alive even when everything goes dark in our computer tent and under the table in kitchen. Those are places we usually have critical servers, right ?

We found rich daddy keen enough to finance some laser communication device. He probably bought used Lightpointes at EBay or Alibaba, but who cares when the link achieves under 10ms latencies from Moon to Mars at any conditions !

The major advantage of FSO (free space optics) devices is their transparent Layer2 bridging - this is like true 50 million miles cable.

Our business needs and survivability dictates to protect following workloads :

- Microsoft Exchange (five VMs consisting of three mailbox servers and two front-ends)
- Web servers, fifteen of them. Local MS SQL is feeding every single one.
- MariaDB cluster, consisting of three nodes. MySQL was probably politically incorrect to choose.
- Five file servers, 15TB in size each. That's lot of movies, did I mention cinema in Challenge II ?
- Fifteen VMs to serve docker containers. Lot of apps to run!
- Two left-overs from the dinosaur era, some say it should go to different space ship but here we are. Cargo and barcode issues are always bad.

As crucial part of design, each scenario needs to have RTO and RPO defined.

RPO - Recovery Point Objective: A definition of the amount of data loss that's deemed acceptable, defined by application, in the event of a disaster-failover scenario. This can be from zero to minutes or hours depending on the criticality of the data. It's a definition that can be quantified by an examination of a given I/O profile, the available bandwidth linking the primary and recovery sites, and the applications tolerance to latency.

RTO - Recovery Time Objective: A definition of the amount of time it takes, from initial disaster declaration, to having critical business processes available to users. Although automation greatly contributes to enabling low RTO, this metric can't be quantified mathematically. It's very dependent on recovery processes that are greatly enhanced with automated technologies.

1.1 Business needs and objectives

As with any design, business needs need to be fully understood in order to provide solution that matches and surpasses demands of customer. Unfortunately in this case, **too many major decision driving factors are unknown, so there can be hefty amount of assumptions and constraints leading to surprisingly wide amount of infrastructure choices.**

Because I expect majority of competitors to use Site Recovery Manager or similar disaster-recovery technology (Zert0, DoubleTake, Veeam Availability…), to make things more complicated and interesting, I chose to perform stretched network/stretched cluster scenario which probably is the most technically challenging one.

Stretched cluster with replicated storage brings up following advantages :

- No need to have and maintain separated Windows Active Directory domains
- No need to have two or more vCenters with underlying infrastructure (patching, updates…)
- Simpler TCP/IP addressing, easily spanning one TCP/IP subnet across both planets
- No need for any additional software on top (no Zert0, Double-Take, SRM…)
- Less room for human error, improper configuration, possibly less unplanned downtimes/outages
- Less maintenance, less effort to keep the infrastructure running
- no necessity for Enterprise Plus licenses for ESXi when appropriate networking infrastructure is in place. In fact, ESX might not know it is running over huge distances at all. Enterprise Plus licensing still might be necessary due to other requested functionality and factors, though.

Of the major disadvantages, following deserve honorary mention :

- more demands on networking infrastructure. Layer2 transparent bridge needs to be created.

Requirements to achieve stretched network and cluster :

- recommended maximum of 10ms latencies between sites
- at least 250Mbps bandwidth for vMotion recommended by VMware
- sufficient bandwidth to perform storage replication between planets (this is not specific of stretched cluster and is valid for all possible configurations and designs)
- stretched network spanning across both remote sites (Cisco OTV, EoMPLS, EoIP, VXLAN…)

My choice was driven by applications that could use their own built-in data replication functionality to achieve business continuity and disaster recovery - with the right technological choices. Microsoft

Exchange with its Database Availability Groups, Microsoft SQL Server with AlwaysOn functionality, MariaDB with built-in replication, easy-to-achieve file replication for file servers, Docker with easily and quickly restartable containers and DB2 with built-in replication mechanisms are pretty in favor of:

a) relying less on VMware infrastructure top-to-bottom for application recovery

b) relying less on operating system clustering techniques and dependencies

c) little necessity of storage awareness and replication based disaster recovery (freedom in datacenters!)

d) long-distance <10ms replication targets

e) using application-native data replication and failovers, which is by far the best from consistency point

When we let applications themselves perform their INTERNAL magic to have data consistent in both datacenters, this is orders of magnitude better than any external storage, network or virtualization based replication. It prevents great amount of unforeseen risks and issues any "application-external" replication can introduce, such as:

- no hard database shutdowns

- no corrupted indexes

- no need to reply transactions plus no transactions lost

- no corrupted jobs, batches or processing (in-the-flight plus caching issues)

- no need to perform long or complicated FSCK/CHKDSK procedures as in the case with virtual-machines-restarted-the-hard-way affecting RTO, RPO objectives

Compared to Site Recovery Manager solution, there are some drawbacks, quirks and complications. Both solutions, chosen stretched cluster and SRM, provide automated failover and automated failback, but SRM also provides :

planned migration with graceful shutdown of production VMs and zero data loss ensured by data sync
non-disruptive testing : production VMs running with isolated environment with recovery VMs
runbooks
audits

Looking at the list of our applications, almost all of them can achieve planned migration with different methods (SQL with AlwaysOn ? Exchange with DAGs ? IIS ? Easy to migrate them from one planet to second without visible downtime when proper planning has been taken). Non-disruptive testing is big thing for sure, but it is not performed every week – usually when infrastructure is set up, and it can be achieved with little manual work which might not represent any serious issue once a year or so.

Our stretched cluster solution doesn't provide nice SRM-style runbooks and automated audits. On the other hand, the major advantages are :

better RTO and RPO (faster recovery)
no additional configuration of site recovery and no additional planning, eliminating human error
faster recovery when disaster strike (as fast as restarting VM in the most complicated case)
minor advantage being simpler VMware infrastructure.

1.2 Requirements, assumptions and constraints definition

CON1: available network bandwidth between planets in unknown. Below 10ms latency is guaranteed.

CON2: no performance statistics are provided. Capacity planning and future growth predictions are impossible, only qualified guess can be performed.

CON3: no information is available about storage in terms of capacity and performance. The only storage-related information we know is "five servers use 15TB of data each".

CON4: storage features unknown. Vendor, model, resiliency, replication possibilities, synchro/asynchro…

CON5: no information available about [theoretically] possible active-active datacenters set up. Some loads could split perfectly, offloading storage/networking/compute resources at primary Mars premises.

CON6: business needs, priorities and goals unknown, importance of applications unknown. Impossible to prioritize RTO/RPO for particular applications, impossible to assess impacts arising from downtime, or create specific start-up order to minimize such impact.

CON7: there seems to be only single link between Mars and Moon. [if it goes down, split brain occurs !]

CON8: exact versions of applications and workloads are not listed (Exchange 2013 ? Exchange 2013 SP1 requiring Windows2012 R2 ? Exchange 2007 ? Windows 2012 ? Win2012 R2 ? IBM DB2 which version ? SQL 2014 ?]

2. Moon infrastructure : disaster recovery datacenter

Choice of using any existing public cloud infrastructure or building our own is presented.

Because we are IT specialists, we natively choose to build our own **mainly due to better optimization** ("fit for particular purpose") and following possible drawbacks or limitations assigned with foreign cloud infrastructures :

Platform choice - no control about infrastructure. We can't influence network, storage, compute resources and their aggregation, utilization, choice of particular servers (model, redundancy) and underlying operating system etc. Can I have ESX v6 in Microsoft Azure today with lives of people depending on it (=is it supported) ? Can I run RedHat7 in Azure ? Is Microsoft already running servers with redundant power supplies ? How often will my virtual machines crash, causing me pain to replicate missed data and generally manage to survive those crashes (file system corruption, database issues etc) ?
Resources – some clouds sell virtual machines, some sell "aggregated bunch of resources". What is the least compute "unit" or "building block" I can buy in terms of CPU/RAM, is it aligned with my needs ? Or will I pay for something I really don't need and/or can't use properly ? What are those CPUs, there is huge performance difference between some cheap 2.0GHz variants and 3.3GHz monsters, so "6 vCPUs are not the same as some other 6 vCPUs" (I can easily influence this when I build up my own servers).
Service offerings and isolation – I really don't want my workloads to be shared, aggregated with or exposed to any other party. I really want my virtual machines to run on my dedicated non-shared hardware (at least servers). My life and life of many others might depend on it. Loss of control is very important for some.
Service level agreements - and then there is the reality. I don't have lawyers on Mars to deal with this. What are the infrastructure fault domains ? We have heard horror stories about some cloud vendors.
Storage services, data/storage integrity and their bindings to other infrastructure – what are storage capacity and speed building blocks? What are possible guaranties in terms of IOPS (read, write) and bandwidth ?
Guest OS support - What operating systems can I run in someone's cloud premises ? Is clustering possible ? Oracle RAC ? Microsoft FailOver clusters ?
Price - Usually clouds are priced pretty high, for what I pay in 12 months or even less to cloud vendor, I can OWN the same infrastructure ! Some cloud vendors have endless list of options and variants to choose from, it is very complicated to get to the most effective solution not only in technical terms. Quite usually potential savings are achieved in large scale only, which is not going to be our case. Moreover we, IT guys, are very efficient in installing, operating and maintaining our own servers.
Time to cloud – how fast can I have my workloads running ? Does it take days to deal with some cloud partner or sales rep ?
Connectivity – how will I connect to my infrastructure hosted in clouds ? What kinds of VPN is possible ? What kind of additional features, such as transparent Layer2 bridging, is available ? Any guarantees for latencies and throughput ?
Backup / recovery options – some clouds require applications to be more "tolerant" and "resistant" to failures due their internal technical limitations. These are quite often forgotten to mention to customer and everybody lears the hard way. We have no room for learning the hard way !
Data security – do we really have control over our data when it's in some cloud we don't control ?

Additional complexity – when dealing with my own infrastructure, I don't need anyone else. No tickets to open. No helpdesk to call. No strange people to talk to. No administrative overhead to deal with. No listening to blatant excuses and "you must be kidding" stuff.
Limited control, limited choice, vendor lock-in - it is reasonable to expect to find little non-HP gear in HP datacenters and HP clouds. Similarly, if Azure secures super-deal with Dell, there will be less IBM servers, IBM storage and Brocade networking, and for some reason we might want to use particular hardware. If we use HP 3Par storage on Mars, it would be convenient to have 3Par on Moon for replication and management purposes, for example. The same applies to Juniper switches and VPN, etc.

Flexibility and SPECIALIZED decision choices for particular use-case are among other factors. Usually there is little possibility to choose "something special" with existing cloud providers and we simply have to use what is readily available from cloud vendor.

2.1 Resource usage calculation

With strong preference and design choice to create our own hosted infrastructure on Moon, let's calculate resources we need to protect :

Tab.1

Usage	Count	CPU	RAM	Total CPU	Total RAM
Exchange	5	4	12	20	60
IIS/SQL	15	2	8	30	120
MariaDB	3	1	4	3	12
File servers	5	2	12	10	60
CoreOS	15	4	12	60	180
Win NT4/DB2	2	1	4	2	8
				125	440
				vCPUs	GB RAM

Grand total, our infrastructure uses 125 vCPUs and 440GB of RAM.

CON2 shows we know nothing about CPU utilization, no performance statistics or capacity trends are mentioned at all. This leaves us with qualified guess only : we can estimate these CPUs are not going to be 100.0% utilized all the time.

For example, file servers tend to load CPUs only slightly even with antivirus scanning or similar background activities.

Based on allocated RAM, all 15 web servers running SQL servers (30 vCPUs total) and all Exchange servers (20 vCPUs total) are NOT expected to be able to drive CPUs extremely high, mainly because performance of database servers will be limited by amount of RAM causing little to no possibility for caching big portion of database in memory and performing quick operations on top of it. Storage performance probably will be keeping CPU utilization relatively low. **While database size is unknown, it is not expected to be super-huge** (ie. 100TB database with 1.000.000 IOPS running on server with 8GB RAM).

Usage	Count	CPU	RAM	Total CPU	Total RAM
Exchange	5	4	12	20	60
IIS/SQL	15	2	8	30	120
File servers	5	2	12	10	60

These three parts of infrastructure account to 60 vCPUs out of total 125 vCPUs, which is 48% of all vCPUs. At the same time, they represent 54% of RAM resources.

Performance estimates can be more problematic for Docker systems :

Usage	Count	CPU	RAM	Total CPU	Total RAM
CoreOS	15	4	12	60	180

While these VMs only have 12GB of RAM, **they possibly can fit significant number of containers with applications which in turn can cause relatively high CPU usage** (it doesn't take "big" application to exhaust CPU). Moreover, docker infrastructure accounts for 60 virtual cores total, representing almost 50% of all virtual cores.

We can easily expect to run some science workloads here, considering we started to colonize Mars. There will be plethora of new information, experiments, measurements and data to process (New Horizons Pluto, July2015 – if we had the transport capacity, we would receive tons of data that would need processing !).

Physical infrastructure influencing factor : **considering all above, I would expect 2:1 to 3:1 vCores-to-pCores aggregation ratio overall, meaning we should comfortably fit all our 125 virtual CPUs into 41 to 62 physical cores**. **Bear in mind these are disaster recovery premises on Moon**, we easily can run our infrastructure even with little performance hit because Moon datacenter is not expected to be active for prolonged periods of time.

Priority of virtual machines in the form of

- **CPU shares**
- **resource pools**
- **CPU limits**
- **CPU reservations**
- **Network I/O control**
- **Storage I/O control**

should be set in accordance with business needs (which is unknown at this time), effectively managing almost no visible performance hit to the most important applications even if physical CPU contention occurs which is very unlikely.

2.2 Recovery datacenter infrastructure

Based on calculation of 125 vCores and 440GB RAM, we **could** continue to use identical SuperMicro servers from Challenge1 which was SuperMicro SuperServer 2048U-RTR4 model http://www.supermicro.nl/products/system/2U/2048/SYS-2048U-RTR4.cfm each configured with :

- four E5-4669 v3 CPUs, 18 physical cores each
- 3TB DDR4 low-voltage load-reduced 1.2V DDR4 modules for energy efficiency
- 24x Samsung 1.6TB SSD for VSAN all-flash, model MZIES1T6HMJH, 12Gbit SAS, 10 DWPD
- 16GB SuperMicro SATA disk-on-module SSD-DM016-PHI for ESXi v6.0 installation
- three additional dual-port Mellanox ConnectX-3 adapters, 56Gbit/40Gbit/10Gbit Ethernet

In order to spread out technological risk, I decided to use totally different servers this time :

Dell PowerEdge hyper-converged systems were evaluated, in particular upper-range models R820, R920 and R930 :

server	CPU Sockets	Cores per CPU	max. RAM (TB)	Disk Bays	Rack Space (U)	Per U capacity (TB)

R820	4	12	3	16	2	16	http://www.dell.c r820/pd?~ck=anav
R920	4	15	6	24	4	12	http://www.dell.c r920/pd?~ck=anav
R930	4	18	6	24	4	12	http://www.dell.c r930/pd?~ck=anav

Primary driving factor to choose Dell R820 model is better storage density – number of internal drive bays to rack space ratio. R820 servers are more "storage dense" for occupied rack space than their R920 and R930 counterparts because on average they provide 16 2.5" drives in 2U rack space compared to 12 slots of R920/R930 models. R920/R930 are created for massive RAM and throughput demands, so their internal infrastructure is not as storage optimized as lower R820 is.

All servers provide sufficient amount of physical processor cores and RAM. Number of expansion PCI-Express slots surpasses any foreseeable demands we might have for the infrastructure (10Gbit or 40Gbit Ethernet adapters Mellanox ConnectX-3), so it is not even mentioned in table above. No significant limitations have been found with R820 at all.

2.2.1 Advantages of different compute infrastructure :

Using different compute nodes in secondary datacenter will protect us against vulnerabilities or failures possibly caused by:

- Part series failures (CPU voltage regulators of low quality, fans, motherboard capacitors, PSUs…)
- Failures in firmware ie. BIOS, RAID adapter, NIC firmware, iSCSI firmware of HBA…
- Failures in VMware drivers, ie. particular RAID adapter, NICs etc ("VSAN low queue depth" style)

While this introduces totally different platform to take care about with different drivers, overall system behavior and possible different set of its own issues, the additional complexity is outweighed by risk mitigation. If one part of our infrastructure is affected by some common bug, the different part of infrastructure can and will take over in case of mass catastrophy in one site (think "can't fork" VMware HP AMS memory leak issue or Brocade switches rebooting after 497 days of uptime leaving datacenter in pitch black).

2.3 Disaster Recovery Topology

Disaster recovery topologies are :

- Active/passive datacenters and one-way failovers
- Active/active datacenters with one-way failovers
- Bi-directional failover
- Shared recovery sites (many to one failovers which is not our case here)

Information about active-active datacenter usage possibility is not available. In specific situation, we could use Moon compute resources as active site, effectively offloading primary Mars premises. I personally do not prefer active-passive configurations, because it leads to unnecessary waste of resources.

My primary goal is always to have resources active, even in recovery/secondary premises. After they have been bought, installed and continuously maintained, the worst "usage" is NO USAGE AT ALL, just passively sitting and waiting for their time to come. This is one of the most crucial objectives and advantages of virtualization : to better utilize existing resources. All of them, all the time. If nothing else, at least some kind of development or testing could be run in passive datacenter on Moon.

Unfortunately, **given the scarce information in this challenge, it is impossible to create respective active/active design** (for example, I could easily run Exchange and IIS/SQL workloads on Moon because there only is <10ms latency and honestly, users do not notice if their emails are delivered or if webpages are displayed 0.1 to 0.5 second later "than they possibly could be"). As displayed in constraint CON6, we don't know any business priorities for particular workloads so it is impossible to run something to Moon because it might be needed locally.

Active/active setup would free up resources in primary Mars premises that could provide performance benefits for example to Docker applications or file serving as it seems to be huge, 5x 15TB of data. Whatever the most sensitive and most critical application is, it has to remain on Mars to provide best response, best performance and lowest latencies for end-users. The rest could be offloaded to Moon, carefully evaluating benefits and drawbacks of such solution. **With <10ms latencies, we can migrate workloads online between Mars and Moon**.

Assumption is we have enough resources on Mars to run all necessary workloads. This must include power, cooling, space to host appropriate amount of compute, storage and networking resources. No shortage is mentioned in challenge so I'm not getting deep into this. 2U SuperMicro servers built up for Mars infrastructure are easily capable of hosting ~120 vCPUs and 440GB RAM workloads.

I would like to elaborate more on active/active or active/passive topic, but this brings up many possibilities and basically no data to establish qualified design.

2.4 Storage replication

One of the main requirements is to have data replicated between datacenters. Because we decided not to use Site Recovery Manager or similar product with host-based replication, we need to have storage that will manage to replicate data between both datacenters and keep them synchronized according to business needs.

If we had to choose storage, it would be iX Systems' TrueNAS Z30 storage solution for its comprehensive features and functionality. It has integrated snapshots and replication, synchronous and asynchronous with online compression. Detailed list of features is out of scope for this project and is readily available on manufacturers site (http://ixsystems.com/)

Data volumes will be separated from operating system volumes. With expansion shelves, up to 888TB of raw capacity is available for this system which is sufficient for our needs. Multiple 10GbE and 40GbE connections are available via iSCSI protocol. We use no fiber-channel on Mars.

2.5 Naming conventions

With stretched networks, it is extremely important to keep perfect overview about locality. Recommendations for naming conventions are :

- Establish extremely concise naming conventions
- Name datacenters and clusters to easily identify them as stretched environments
- Recommendation is to use planet names, datacenter names, ie. MARSDC01ESX06
- Consistency is the key. Everything must be perfectly consistent to be instantly identifiable.

2.6 Split brain scenario

The major technological risk associated with proposed solution is split-brain situation, where the communication link between datacenters goes down and each local site can't decide if it needs to start up the workloads protected from the other side. In usual conditions, split brain scenario is arbitrated by:

- Additional isolation addresses availability
- Storage heart-beating
- SAN quorum witness mechanism
- File-share witness

Unfortunately, nothing like this is possible in our Mars-Moon situation. There is no third site available nor it has been mentioned. While this represents major technological obstacle, stretched network/stretched cluster configuration I chose is no different to standard SRM solutions, effectively crippling them in possibilities to perform disaster recovery. This means :

1) communication link Mars – Moon is the most critical point in infrastructure.

2) at the same time, it is the only single point of failure in the whole infrastructure.

This is clearly a business / project risk that needs to be pointed out and fully understood. In the challenge, nothing was mentioned about RELIABILITY of our laser link. Because we don't have duplicated links or any other form of communication, this is clearly technological obstacle to provide reliably working solution. No matter what technology will be used, storage based replication, host based replication, VMware's SRM, Zert0, Double-Take or anything else, with no third point in Space used as a referee to arbiter network outages, we are doomed.

In such case, virtual machines and workloads will be blindingly started on Moon. When the communication link is re-established :

- IP conflict on network would immediately occur
- Storage would not be able to replicate data correctly as it can't judge which data are to be replicated which way.

Remedy : in case of laser link outage, Moon as secondary datacenter is set shut down running virtual machines and storage in order to protect the workloads. *If there is no link, nobody on Mars can use any workloads running on Moon, so they are "unnecessary".* **After link is re-established, MANUAL INTERVENTION is necessary to forcibly replicate appropriate storage LUNs from Mars to Moon in order to have most actual data on Moon.** Immediately after link outage, Mars side should be reconfigured to disable bridge, in order to prevent potential IP address conflict with workloads if they are started on Moon.

If there is infrastructure outage on Mars during laser link outage, this will represent total downtime as there is no possibility to start and run virtual machines anywhere else. We have three datacenters on Mars that should protect us "locally", mitigating the reliance on interplanetary link.

This situation is valid for all scenarios, not only for particular solution discussed here, so it DOES NOT REPRESENT COMPARATIVE DISADVANTAGE to other solutions.

2.7 DRS groups and rules

If active/active datacenter configuration would be used, to separate resources on Mars and Moon, specific DRS rules must be configured. Virtual machine SHOULD run on host rules would be configured to bind virtual machines to preferred owners, effectively separating chosen VMs to run on Moon during normal situation and fail over to Mars in case of emergency when particular hosts defined in rules are not available.

3. Application Disaster Recovery Plan

Following workloads need to be protected :

- Microsoft Exchange (five VMs consisting of three mailbox servers and two front-ends)
- Web servers, fifteen of them. Local MS SQL is feeding every single one.
- MariaDB cluster, consisting of three nodes.
- Five file servers, 15TB in size each.
- Fifteen VMs to serve docker containers, running CoreOS
- Two Win NT4 virtual machines with IBM DB2 databases.

3.1 Microsoft Exchange disaster recovery protection plan

Unknown version of Microsoft Exchange is being run on Mars. I expect it to be Exchange2013 due to architectural advances compared to older versions. Exchange2007 is already out of support and Exchange2010 doesn't seem to be reasonable choice for such important project.

For overview of architecture, mail flow and roles in Exchange 2013, please consult following articles :

http://blogs.technet.com/b/rischwen/archive/2013/03/13/exchange-2013-mail-flow-demystified-hopefully.aspx

https://technet.microsoft.com/en-us/library/aa998825%28v=exchg.150%29.aspx

https://technet.microsoft.com/en-us/library/bb125012%28v=exchg.150%29.aspx

https://technet.microsoft.com/en-us/library/aa996349%28v=exchg.150%29.aspx

https://technet.microsoft.com/en-us/library/Dd979799%28v=EXCHG.150%29.aspx

Microsoft released several documents, specifically targeting Exchange high availability, site resilience, design planning. Following two are highly recommended :

https://technet.microsoft.com/en-us/library/dd638104%28v=exchg.150%29.aspx

https://technet.microsoft.com/en-us/library/dd638137%28v=exchg.150%29.aspx

Database availability group should be used to provide resilient and redundant storage for mailboxes. Specific requirements related to design are available in linked documents.

We will be using single domain with proper DNS services. Name of DAG servers will be shorter than 15 characters. No special hardware or storage requirements - DAGs don't require or use cluster-managed shared storage. Cluster-managed shared storage is supported for use in a DAG only when the DAG is configured to use a solution that leverages third party replication API.

Because we are already running three mailbox servers and two front-ends, recommendation is to :

a) create additional two virtual machines at Moon premises and include them in DAG group to have all databases replicated to them

b) create additional two virtual machines at Moon premises and let them serve as Front End Transport service, Client Access server.

Virtual machine configuration :

Original MARS Exchange VMs	Replicaton MOON Exchange VMs
4x vCPUs	2x vCPUs
12GB RAM	24GB RAM
Unknown storage	Unknown storage
Total Count : 5	Total Count : 4

By creating these additional virtual machines, we will protect Mars citizens against outages created in their local datacenters the most simple and straight-forward way. When disaster strikes on Mars, email will be delivered from Moon with little delay.

Design justification : configuration with less CPU cores but more memory was chosen intentionally. 100% CPU Load is not exactly expected on these systems, but definitely amount of RAM would be too small when consolidation is requested. 2 CPU cores are able to deliver thousands of messages each second. If this configuration would be insufficient, additional CPU cores could be added online via Hot-Add functionality. More RAM will allow systems to perform better caching, saving precious storage performance, contributing to better overall behavior.

When configuring new VMs, each new DAG member must have the same number of networks as existing members have. Moreover, each DAG member must have no more than one MAPI network which must provide connectivity to all other Exchange servers and other services, such as Active Directory and DNS. If separate MAPI and Replication networks are used, they must be on different TCP/IP subnets. Latency below 500ms must be achieved between DAG members, maximum 16 members can be in a DAG group.

Assumptions :

1) we don't know if single Active Directory domain is used and if all these servers are member of it

2) number of physical paths between DAG members is unknown, MAPI network and Replication network configuration is unknown.

3) bandwidth requirements are unknown for Exchange plus bandwidth of laser link is unknown, too

4) single VM is able to withstand the load, created by email clients.

5) email communication is critical. Five virtual machines devoted to it show very high importance. **This was the driving force to create 2+2 additional VMs on Moon** (we could get around with single one for Client Access and single one for Mailbox DAG, but that would represent no redundancy at all when disaster strikes on Mars).

Design considerations : because we have storage replicated between Mars and Moon, ESX High Availability functionality was considered : when original virtual machines would collapse on Mars, they would be automatically restarted in configured order on Moon. This would represent temporary outage for email clients on Mars, expected to be about 10 to 20 minutes depending on speed of storage (ability to start Exchange systems fast).

While this would be functional solution, importance of email communication and pretty heavy allocation of existing resources to it signalizes we demand more than just simple VM restart. Because this application failover functionality is built in Exchange software, there are not too much reasons to leave it unused.

3.1.1 Exchange Email RTO, RPO definition

RTO : 0 minutes. Instant failover is provided on application level.

RPO : 0 minutes. Instant failover is provided on application level.

These targets are not possible with HA restarting virtual machines.

3.1.2 Exchange Email Procedures and Processes

No manual intervention is necessary with proposed configuration. Administrator needs to make sure original VMs are restarted when situation allows. No special checks are necessary.

3.2 IIS Web Servers running local instances of MS SQL

Fifteen IIS web servers with local MS SQL instances need to be protected. Assumption is these servers provide different functionality, different sets of applications with no redundancy at this moment.

Current configuration :

15x 2 vCPU / 8GB RAM

Design suggestion is to aggregate these workloads and create two huge VMs on Mars :

2x 4 vCPU / 48GB RAM

One possibility would be again to blindingly restart crashed VMs in remote premises, given we have stretched network with Layer2 bridging and replicated storage. Clever placement of VMs data on appropriate LUNs would allow us to have the same data synchronously available at remote datacenter. No SQL server replication or advanced AlwaysOn functionality is necessary.

Design choice : SQL Databases replicated and protected by AlwaysOn shared-nothing architecture. Virtual IP address is used to access SQL Data, if original VMs fail on Mars, these data will be seamlessly supplied by Moon datacenter. Load balancing to be used for IIS HTTP/HTTPS.

If IIS load balancing would prove unacceptable due to performance/latency reasons, IIS services would be switched off and left sitting idle on Moon. Simple monitoring script can be created to start them up automatically if one or more IIS servers are not reachable on Mars. We have latency promised below 10ms, so that should represent no real problem for users.

3.2.1 IIS Web Servers running local instances of MS SQL RTO/RPO

RTO : 0 minutes. SQL AlwaysOn plus IIS load balancing provide “no outage” situation.

RPO : 0 minutes. SQL AlwaysOn plus IIS load balancing provide “no outage” situation.

3.2.2 IIS Web Servers running local instances Procedures

No manual intervention is necessary with proposed configuration. Administrator needs to make sure original VMs are restarted when situation allows. No special checks are necessary in disaster situations.

Proper planning and careful configuration is needed in order to provide services as necessary. This is application layer.

3.3 MariaDB cluster disaster recovery protection plan

We need to protect three MariaDB nodes. Default master-slave and master-master replications are limited to two nodes, so for our particular case, we will use Galera Cluster functionality.

MariaDB Galera Cluster is a synchronous active-active multi-master cluster. Currently it is limited to Linux platform only – no platform was mentioned in challenge data, but given these virtual machines use 4GB RAM, we might assume there is Linux (Windows has bigger overhead and this really is least amount of RAM someone should run any database server on). Advantages of Galera are :

- Synchronous replication
- Active-active multi-master topology
- Read and write operations on any cluster node
- Automatic membership control, failed nodes drop from the cluster
- Automatic node joining
- True parallel replication, on row level
- Direct client connections, native MySQL look & feel

The above features yield significant benefits for clustering solution, such as:

- No slave lag
- No lost or corrupted transactions when database node fails
- Both read and write scalability
- Smaller client latencies

MariaDB Galera Cluster uses Galera library for the replication implementation. MariaDB supports replication API via wsrep API project. Because particular version of MariaDB was not mentioned in challenge, following table lists WSREP Library to MariaDB version bindings :

Galera wsrep provider version	MariaDB Galera Cluster version
25.3.9	10.0.17, 5.5.4
25.3.5, 25.2.9	10.0.10, 5.5.37
25.3.2, 25.2.8	10.0.7, 5.5.36
23.2.7	5.5.35

Detailed description of MariaDB and Galera configuration is outside of scope of this document, following links should be used to consult implementation, configuration and operational details :

https://mariadb.com/kb/en/mariadb/what-is-mariadb-galera-cluster/

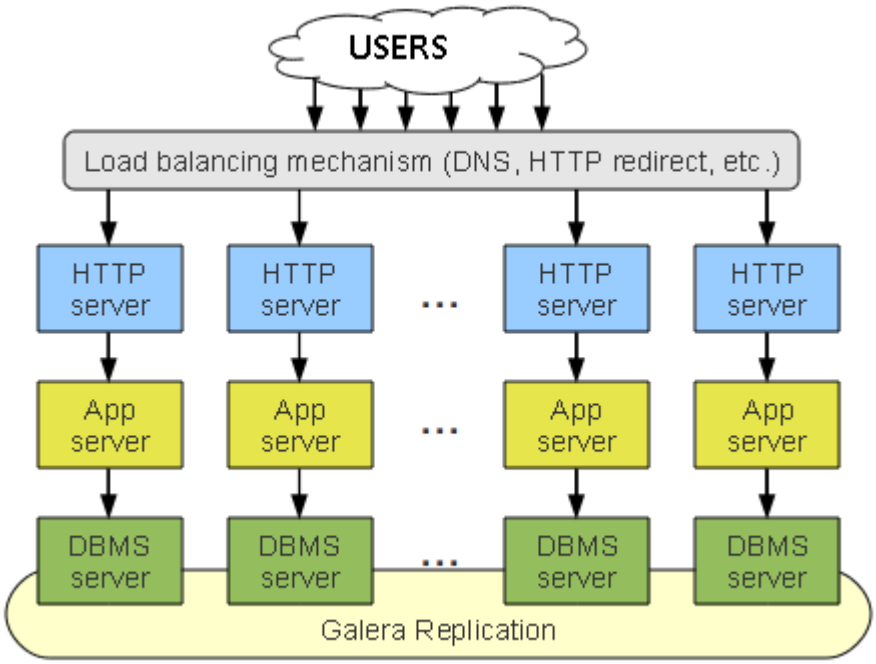
https://mariadb.com/kb/en/mariadb/mariadb-galera-cluster/

https://mariadb.com/kb/en/about-galera-replication/

http://galeracluster.com/documentation-webpages/gettingstarted.html

http://jmoses.co/2014/03/18/setting-up-a-mysql-cluster-with-mariadb-galera.html

No specifics as of how exactly MariaDB nodes are accessed and how they are providing data to network or applications were provided. Default architecture looks like :



3.3.1 MariaDB RTO, RPO definition

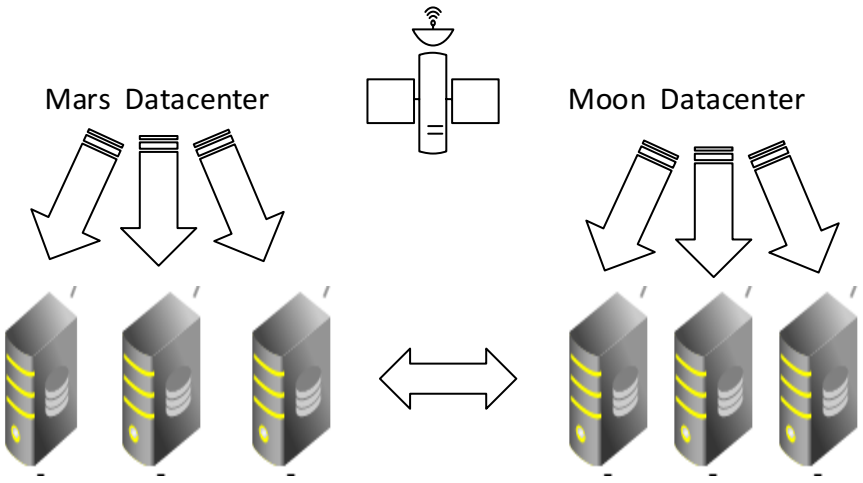
Synchronous replication with active-active cluster nodes spread across Mars and Moon enable us to define :

a) RTO of 0 minutes

b) RPO of 0 minutes

In case of any host failure on Mars, surviving active hosts on Moon will seamlessly take over. There is no visible outage for clients at all. With proper database management and laser link uptime (discussed above), this solution will achieve 100.0% availability.

For sake of simplicity, this solution requires to have identical three virtual machines running on Moon, providing one to one copy of protected MariaDB nodes from Mars :



Our architecture is stretched network so all nodes will use IP addresses from the same TCP/IP subnet, simplifying management, access and replication. IP Addressing is not part of this design, as it is unknown for original configuration.

When time and particular priorities allow, I would recommend to investigate possibility to consolidate databases to single VM on Mars as it could be suitable for recovery/redundancy purposes and it is less effort to manage one virtual machine instead of three. Of course, having only single VM puts more strain on any planned actions requiring downtime and failure domain is wider (three services at the same time), but we have to note this is recovery facility and recovery instances only.

3.3.2 MariaDB processes and procedures used in disaster situation

MariaDB internal replication will automatically manage availability of data in disaster situations.

No manual intervention is necessary.

In case of host failure on Mars or Moon, failed virtual machine = MariaDB cluster node will be automatically restarted by VMware HA functionality on surviving hosts without manual intervention.

In case of total disaster on Mars, laser link bridge and round-robin mechanisms will manage seamless and transparent redirection to Moon resources without visible outage for users on Mars.

Depending on failure, appropriate manual action need to be taken in order to raise cluster resiliency against consequent failures. It could be either host failure on Mars (requiring restart, hardware intervention), infrastructure failure (no power available at Mars premises, dust storm) or laser link communication failure requiring different set of actions.

3.4 File Servers disaster recovery protection plan

Five file servers, each with 15TB of data, are present in infrastructure. Their operating system is unknown, could be Windows or Linux or anything else. This factor severely limits possibilities of disaster recovery design.

For example, if we knew Windows 2012 R2 is used, it would give us SMB3.0 protocol with its Continuously Available File Shares, SMB Scale-Out, SMB Direct, SMB Multichannel and mainly SMB Transparent Failover features. Also Windows clustering could be used to provide redundancy against failures and automated recovery when disaster situation strikes. DFS replication would take care of data replication, if storage-based or host-based replications are not possible to use or if additional benefits would be provided. The list of possibilities goes on and on.

Right now, these file servers are "black boxes". The only thing I realistically can provide is high-availability mechanism from ESX hosts and rely on replicated storage to have data available at backup premises on Moon. When virtual machine fails, it will be automatically restarted via ESX HA.

Major assumption is the link between Mars and Moon has sufficient bandwidth to replicate data either synchronously or asynchronously in specified schedules. Both solutions have their advantages, synchronous replication having 0 minutes RPO.

3.4.1 File Servers RTO and RPO definition

RPO : 0 minutes for real-time replication

RTO : 15 minutes. Virtual machines will be automatically restarted via ESX HA, it just takes some time until they come back online.

Risks :

a) if real-time storage replication is not possible, RPO of 0 minutes is not achievable. In this situation, conditions of how often replication is possible dictate RPO. For example, if data can be replicated in 30 minute intervals only, then RPO of 60 minutes is realistic (one replication fails, the former took place another 30 minutes ago).

b) RTO, time to recover, might be significantly prolonged by FSCK, CHKDSK or similar tools running automatically over filesystem. Because virtual machine is restarted after hard crash in primary datacenter, filesystems will be left in dirty status with consequent check after restart. As we have 15TB of data, this might take tremendous time to run through depending on number of files, folders and discovered issues. Consequent reboot or two and additional run of FSCK/CHKDSK might be necessary, prolonging recovery time even further into realm of HOURS. Windows2012 ReFS and Solaris/FreeBSD/Linux ZFS filesystem implementations do not perform these checks at all, so they do not

represent this risk. There are some other drawbacks associated with these filesystems, but details are outside of scope of this design.

3.4.2 File Servers Processes and Procedures

Restart of failed virtual machine will be done automatically with no manual intervention being necessary. Stretched network will make sure no additional changes are required and clients will have files available the same way as before.

The only recommended manual action is to perform standard checks on filesystem level to make sure there is no silent unnoticed corruption. In case of problems, restore from backup is necessary for affected files and directories.

Additional information available about Windows2012 SMB3.0 protocol advantages :

http://blogs.technet.com/b/clausjor/archive/2012/06/07/smb-transparent-failover-making-file-shares-continuously-available.aspx

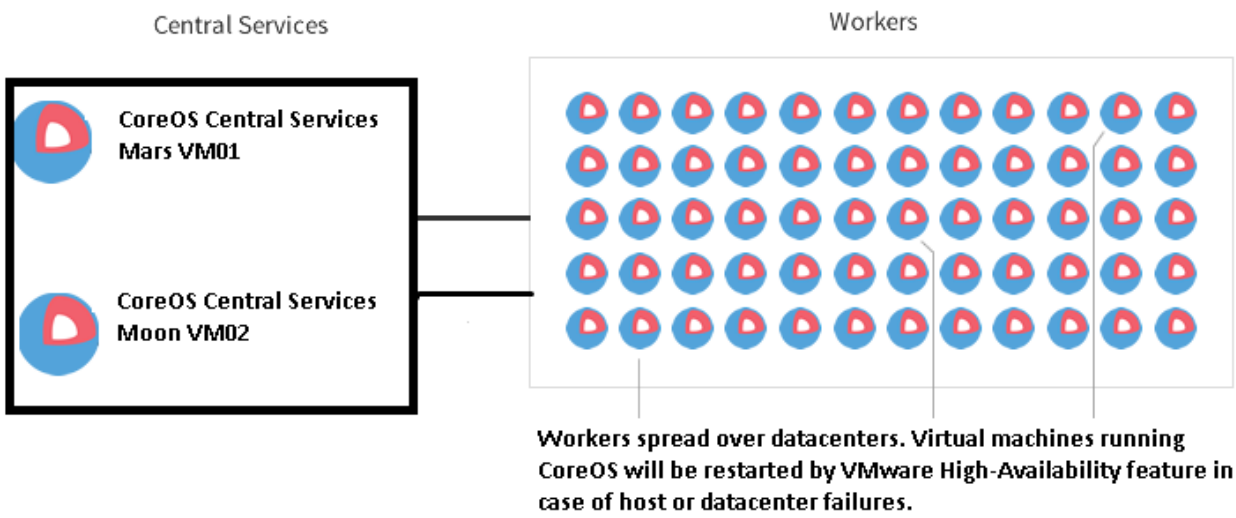
https://technet.microsoft.com/en-us/library/jj127250.aspx

https://technet.microsoft.com/en-us/library/dn281957.aspx

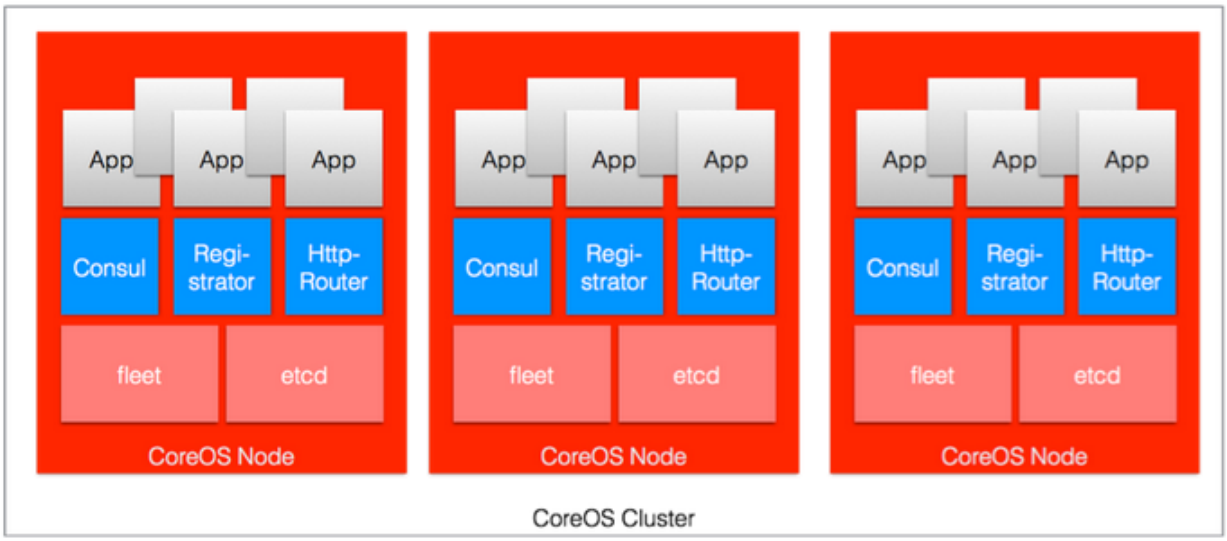
3.5 CoreOS with Docker disaster recovery plan

Fifteen virtual machines are running CoreOS with docker containers. For such number of virtual machines, a few of them should be dedicated to run central cluster services such as etcd and distributed controllers for applications.

I recommend to spawn one central services VM on Mars and one on Moon. Because they will be placed in single TCP/IP LAN segment, they will easily provide supplemental roles in disaster scenarios :



Separating these services out into a few known machines makes sure they are distributed across planets, datacenters (and potentially server cabinets within each datacenter) to create and span across "availability zones" in order to provide maximal redundancy. The role of discovery service (such as Consul) is minimized :



Components and roles :

- ETCD provides a reliable mechanism to distribute data through the cluster. It's the CoreOS distributed key value store.
- Fleet : cluster wide init system of CoreOS. Allows to schedule applications to run inside the Cluster.

- Consul: example application which eases service discovery and configuration. Allows services to be discovered via DNS or HTTP and provides the ability to respond to changes in the service registration.
- Registrator : registers and deregisters Docker containers as a service in Consul. Registrator runs on each Docker Host.
- HttpRouter : dynamically routes HTTP traffic to any application providing a HTTP services, running anywhere in the cluster. Listens on port 80.
- Apps : actual applications that may advertise HTTP services to be discovered and accessed.

Typical application example could be NGINX or multi-tier applications with connection to database back-ends.

Replicated storage will make sure virtual machine data is available on Mars and Moon. VMware High Availability mechanism will make sure any of fifteen virtual machines is restarted in case of host failure or datacenter failure. As long as there is no laser link disruption, this will make docker containers available at all times, protected against Central Services component outage, too.

3.5.1 CoreOS with Docker RPO and RTO definition

RPO : 0 minutes. We have storage replicated real-time between datacenters.

RTO : 15 minutes. In case of crash, affected virtual machines need to be automatically restarted.

Depending on infrastructure, particular application and its internal redundancy mechanisms, no visible outage might be experienced by end users.

3.5.2 CoreOS with Docker procedures and processes :

Automatic restart of virtual machine is performed in case of host or datacenter outage. No manual intervention is necessary. Depending on applications used with docker, no further manual intervention might be necessary at all.

3.6 Windows NT4/IBM DB2 disaster recovery plan

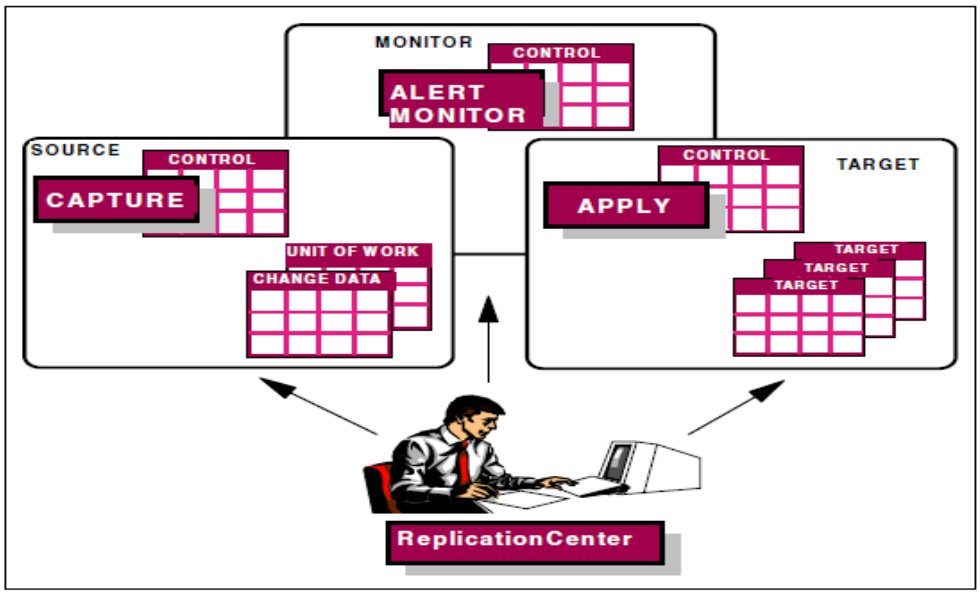
Windows NT4 platform is very old and barely suitable for virtualization. It is already out-of-support by Microsoft, no patches, no maintenance fixes are released for long time. While I would recommend to migrate data away from these servers as soon as possible, we have to deal with current situation.

Disaster recovery would be based on DB2 internal mechanisms, similarly to MariaDB situation. Once again, particular version of software is missing, so it is complicated to assess features as they were developing during time. As basis for following design, IBM DB2 Data replication V8 (RedBook) dated December 2002 was used.

DB2 supports bi-directional replication with multiple active-active sources and targets. It uses four components :

- Administration
- Capture
- Apply
- Alert Monitor

These components communicate via control tables, created and populated via Replication Center GUI, which defines replication sources and maps sources to targets :



DB2 synchronization requires to :

- create and configure the primary and standby databases.
- configure communications between the primary and standby databases.
- choose a synchronization strategy (log shipping, log mirroring, suspended I/O and disk mirroring, HADR High Availability Disaster Recovery)

DB2 replication can be synchronous, real-time : two identical virtual machines need to be spawned in Moon datacenter, providing replication copies for main Mars instances. **HADR mode of DB2 database with SYNC, synchronous configuration, needs to be used.**

SYNC mode provides the greatest protection against transaction loss. In this mode, log writes are considered successful only when logs have been written to log files on the primary database and when the primary database has received acknowledgement from the standby database that the logs have also been written to log files on the standby database. The log data is guaranteed to be stored at both sites with this configuration.

Our architecture is stretched network so all nodes will use IP addresses from the same TCP/IP subnet, simplifying management, access and replication. IP Addressing is not part of this design, as it is unknown for original configuration.

Detailed installation, configuration and operation of DB2 is outside of scope of this document. Original vendor documentation plus https://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.ha.doc/doc/t0051385.html is recommended to follow.

3.6.1. IBM DB2 RTO, RPO definition

RPO : 0 minutes RPO will be achieved via HADR mode and synchronous replication

RTO : 15 to 30 minutes. With these old databases, manual intervention is necessary to promote standby database to primary status. Depending on existing notification methods, automation level (scripts etc), availability of administrators and similar factors, it will take 15 to 30 minutes. Anything below 15 minutes might be too optimistic.

Note : too many unknown factors are present in this scenario. Given some conditions are met, such as extreme little latency for laser link (we've been told "latency is below 10ms" in the worst case so easily the latency might be 1ms) and sufficient bandwidth/throughput, we could also incorporate Fault

Tolerance ESX mechanism : IBM DB2 virtual machines only contain single vCPU and 4GB RAM, so they are definitely small.

FT would protect them online, real-time, seamlessly, transparently and… effortlessly. That would be the best protection, disaster recovery and business continuity by far with zero minutes RPO and RTO targets. In our situation, usage of FT is questionable, so I opted for “standard” disaster recovery design.

3.6.2 IBM DB2 processes and procedures used in disaster situation

DB2 internal replication will automatically manage availability of data in disaster situations.

Manual intervention or automated script is necessary to promote standby databases to primary role.

In case of host failure on Mars or Moon, failed virtual machine = DB2 node will be automatically restarted by VMware HA functionality on surviving hosts without manual intervention.

In case of total disaster on Mars, laser link bridge and human intervention will manage promotion of Moon resources to production status, albeit with visible outage for users on Mars.

Depending on failure, appropriate manual action need to be taken in order to raise cluster resiliency against consequent failures. It could be either host failure on Mars (requiring restart, hardware intervention), infrastructure failure (no power available at Mars premises, dust storm) or laser link communication failure requiring different set of actions.

Links

1. http://www.ixsystems.com/truenas/

2. http://www.dell.com/us/business/p/poweredge-r820/pd?~ck=anav

3. http://www.dell.com/us/business/p/poweredge-r920/pd?~ck=anav

4. http://www.dell.com/us/business/p/poweredge-r930/pd?~ck=anav

Assorted notes : to use in text appropriately

VCENTER PROTECTION - run out of time, would protect it by HA to be automatically restarted on Moon if it crashed on Mars.