Deployment of High Availability Computing Clusters on Research Vessels

Jon C. Meyer

Information Systems Manager, Shipboard Technical Support

Scripps Institution of Oceanography, UC San Diego

To Be Discussed

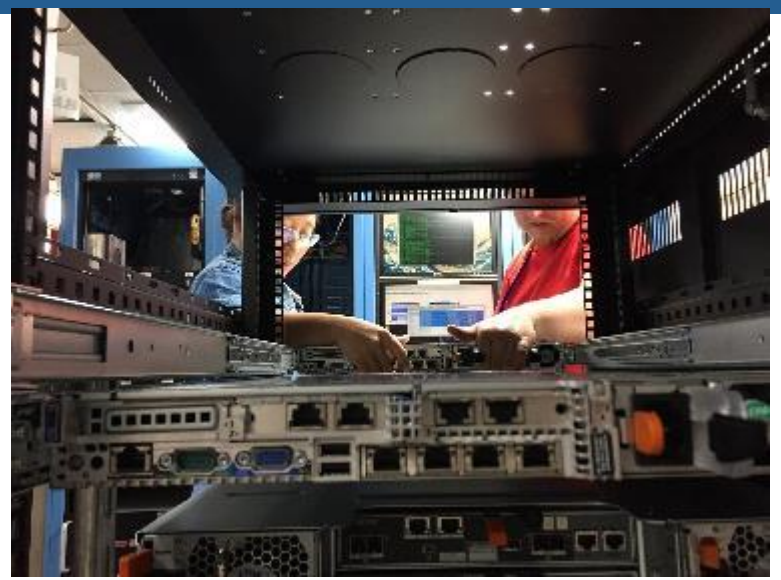
1. What is a cluster?
2. Motivations for buying clusters for ships
3. Styles of clusters evaluated
4. Overview of our cluster design
5. Five clusters purchased
6. Lessons Learned
7. Risks and Rewards

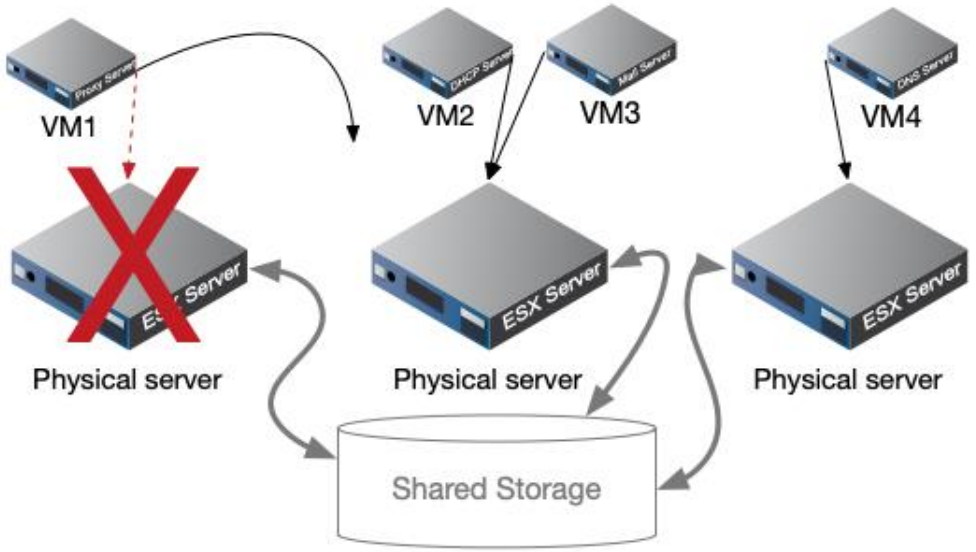
What is a cluster?

en.wikipedia.org/wiki/Computer_cluster

“A computer cluster is a set of loosely or tightly connected computers that work together so that, in many respects, they can be viewed as a single system.”

“Clusters are usually deployed to improve performance and availability over that of a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.”





UC San Diego

Motivations for buying clusters for ships

- It’s a ship. Failures happen.
  - Rust happens
  - Power blips happen
  - We need redundancy of some sort

- There is no counting on cloud resources on ship.
- There are some services aboard that we cannot tolerate downtime on (E.G. DNS)
- (continued...)

Motivations for buying clusters for ships

- Enterprise-grade hardware means longer lifespan and vendor-defined lifecycle maintenance plan.
- Overall, this means less average change year-to-year for the installations and stability of our ship's cyberinfrastructure is increased.
- High Availability – no single point of failure, by design
- Commercial Off The Shelf (COTS) solution means better outside vendor support
- Worldwide manufacturer(s) mean support in many ports
- (continued...)

UC San Diego

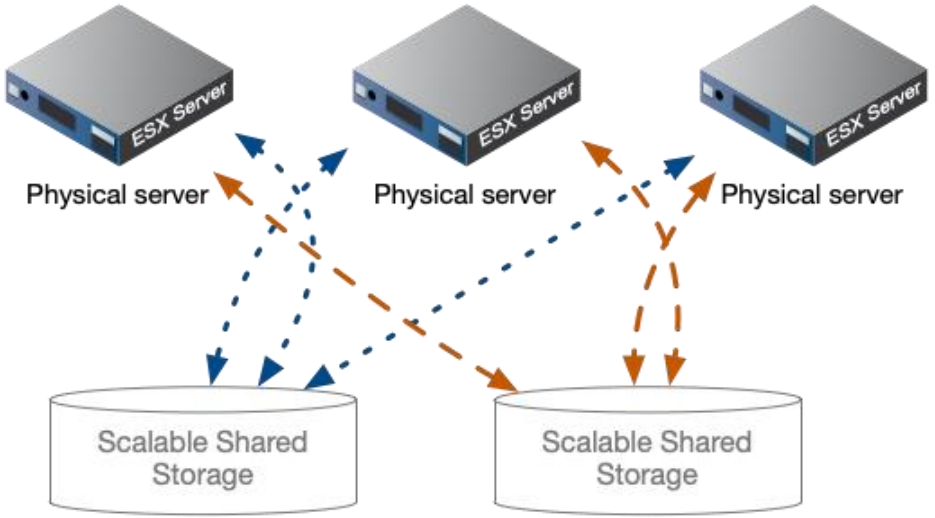
Motivations for buying clusters for ships

- Leverage expensive computing power across more OS installations – better resource utilization.
- This means VMs that are not doing much can have leanly allocated proc/RAM/etc
- Less bare metal OS installs means less computers in racks – more space-efficient.
- Decouples OS installs from specific hardware – upgrading your cluster does not have to mean upgrading your service architecture and vice versa.
- Ability to regularly backup OS installs in a uniform manner without the OS being impacted.
- Not graphics-intensive, to start. Focused on services.

UC San Diego

Styles of clusters evaluated

Traditional (disaggregated) cluster	Hyperconverged cluster
Comprised of two logic sets of components: 1. 2 or more computers with processor, RAM and network available to the cluster 2. 1 or more shared storage devices via network (E.G. Nimble, Dell PowerVault)	Ideally, just multiples of the same computer: All processing, RAM, network **and** storage in 3+ of the same computer



Styles of clusters: Traditional (disaggregated)

- Pros
  - Has been an industry standard for some time
  - Flexible integration model: can use various storage arrays for various purposes.
  - Large, modern file systems use a lot of RAM and proc. Shared Storage has unique RAM and proc, so your cluster's RAM/proc design does not necessarily need to account for it.
- Cons
  - Storage management devices have their own administrative interface, so more to learn, especially if you have different vendors
  - Also more to configure for monitoring and notification of failures
  - Storage management for cheaper arrays can have an OS-specific UI that are not fantastic

Styles of clusters: Hyperconverged

- Pros
  - Gaining popularity as a standard
  - Simpler to administer and understand – all resources managed from the same UI
  - Potentially cheaper hardware
  - Potentially more efficient use of space and computing resources
- Cons
  - Your nodes must generally all be the same, meaning you have to budget to replace all nodes around the same time … or undermine administrative simplicity to your cluster when you partially upgrade
  - Costs savings (compared to traditional) can be trivial (or worse) when buying minimal clusters – may be better to consider for bigger deployments

Styles of clusters evaluated

Choice: Traditional (disaggregated) cluster

This is chiefly so that we can splay our lifecycle replacement costs across years
(E.G. upgrade shared storage one year, processing servers the next)

Overview of our cluster design

Purpose	Model	Software	Count	Notes
Processing nodes	Dell R630/R640	• Vmware ESXi • Vmware vSphere • Vmware vCenter	2 or more	• Dual PSU per chassis
Primary storage array	HP Nimble or Dell PowerVault	• Vmware vCenter	1 or more	• Used for all VMs, • Dual controllers • Nimble is active/passive • PowerVault is active/active
Backup target	Dell PowerVault	• Windows server • Veeam (direct-attached LUN)	1 or more	• Dual controllers • Dual PSUs • Can sub in for primary, in a pinch
Network	Dell S5000 or direct connect	• NetBSD under-the-hood (S5000) • Vmware vSwitch	S5000: 2 or more	• S5000 switches stack to become one logical unit • Redundant stacking cables • Dual PSU per chassis

UC San Diego

Five clusters purchased, installed/installing

Location	Purchased	Install Date	Production
SIO Colocation Facility (Munk Lab, shore)	2016 Dec	2017 Feb *with vendor-supplied staff training	2017 Mar
R/V Roger Revelle	2016 Dec	2017 May	2017 Sep
R/V Sally Ride	2016 Dec	2017 Jul	2017 Oct
R/V Robert Gordon Sproul	2018 Jul	Pending (2018 Dec planned)	2019 Feb (planned)
USCGC Healy	2018 Jul	Pending (2018 Nov planned)	2019 Mar (planned)

UC San Diego

Lessons Learned

- Single-vendor solution is probably best
  - We opted for more performance from and up-and-coming vendor (Nimble) who was bought by HP
  - We use Dell servers, switches and storage for everything else. Despite the performance the Nimble provides, given it to do over, having a single vendor to contact would be better.
- Hyperconverged clusters probably should be a minimum of 3 nodes
- Adding graphics performance adds notable expense
  - There are a couple options, one is more complex, one is more expensive
- If you have a smaller install, you don't need 10Gb switches
  - A 2-node cluster is a way to get redundancy over two computers, keeping costs notably lower – but with no expansion capability

Risks and Rewards

- Risks
  - Added complexity
  - Steep learning curve
  - More abstract problem domain to troubleshoot
  - Stronger need for cybersecurity to prevent disruptive failures
- Rewards
  - Increased stability
  - Hardware failures are not disruptive to users
    - Science mission, ship operations can continue, ignorant of failures, as desired
    - Feasible to attend to failures when convenient, rather than react immediately
  - This makes is feasible to consider admin from off-ship, only (with regular maintenance visits) to combat the complexity problem

Questions? Thanks!

UC San Diego