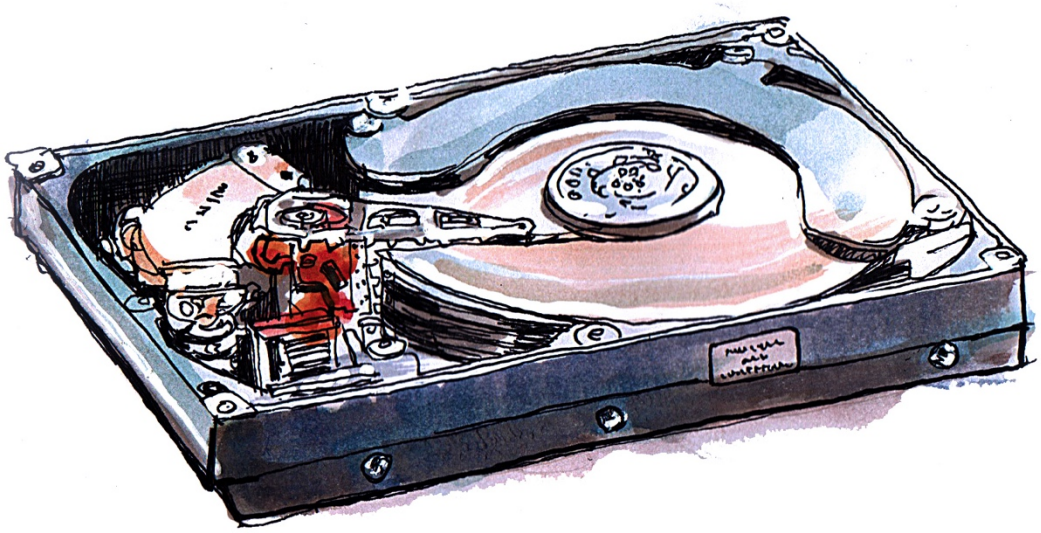
Langton Blue

Author: Chris M. Evans

CHALLENGES OF STORAGE APPLIANCE SCALING



Commissioned by Scality and independently authored by Langton Blue Ltd

First Published: 4 April 2016
Latest Update: 4 April 2016
Document ID – LB1WP0013
Release 1

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

Table of Contents

Table of Figures

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

1- What do You Want to Scale?

The need to scale in storage appliances comes from two basic requirements; increasing storage capacity and/or storage performance. Capacity defines the ability to add more storage space to the system and in effect store more data. Performance defines the ability to service more I/O requests in a timely fashion and can be divided into both throughput (IOPS, I/O's per second or MB/s) and latency (response time of an I/O request, usually measured in milliseconds).

The need to scale across both axes is determined by how application demands are changing. Some applications will increase in capacity as, for example more records are added to a system, more documents are available for searching, or more users are taken on board. From a performance perspective, an application may start processing more I/O requests as a web server receives more traffic, or as the films within a content delivery system become more popular.

In most cases, the performance and capacity demands of applications will scale independently. The scaling ratios aren't guaranteed to be fixed and this is especially the case in mixed workload environments where the requirements are unpredictable and based on the status of many applications. Over time, there may be more demand for capacity, followed by an increase in performance as new data is processed by the application. There has been a steady increase of mixed workload environments, with the rise of cloud service models, and use cases like IoT (Internet of Things) and "data lakes."

1.1- Why Scale?

When it comes to storage infrastructure design, there are two main approaches that can be taken. Systems can either be scaled, with increased capacity and performance, or additional separate systems can be added to the infrastructure. Taking the separate systems approach creates "islands" of storage, so described because in general those pools of storage are distinct from each other and without additional capabilities, the resources can't be combined together into one logical pool of storage. Obviously there are some reasons why separate storage appliances have to be deployed, for instance where physical data centre space or power is constrained, or when data has to be geographically dispersed for resilience. Outside of these issues, the decision on which route to take is based on a number of factors:

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

- **Risk** – having a single environment storing all data that continually grows, increases risk for the organisation. There is the risk of system failure and outage, which by implication would affect all applications. IT departments have to decide what defines their acceptable level of risk in terms of appliance size. Making that decision isn't straightforward. It involves considering a number of aspects, including the actual capabilities of the hardware, the reliability of the vendor's products and the amount of maintenance and updates that occur over time.
- **Operational Complexity** – a single system can be both easier and more complex to manage than multiple systems, depending on scale. Having all data in a single system means any changes that impact the storage (upgrades, system maintenance, code fixes/patches) can affect the business and so getting maintenance slots can prove a problem. Having many systems means those problems aren't as much of an issue; the impact of maintenance is reduced to the users on that system. However, as multi-system environments grow, the management aspects increase significantly. Free space on systems may become fragmented and adding capacity to one application may require migrations or data movements. There are simply more systems to upgrade, so any maintenance work has to be planned and executed multiple times. Maintenance slots typically get reserved for out of hours (such as the weekend) to minimise the impact if a failure does occur, meaning there are relatively few slots throughout the year that can be used, especially if other maintenance work (on servers, the network or application) are also taking place.
- **Application Restrictions** – applications accessing data through a namespace (like a file system, or HTTP/REST-based API) may require additional programming or virtualisation software to make multiple physical appliances appear as part of a single logical infrastructure. Some vendors mitigate this problem through the use of federation (allowing systems to act as a single group) and multi-pathing, whereas for other platforms (particularly object stores, the ability to logically span multiple infrastructure components is inherent in the design.
- **Data Migration** – in a static architectural model (such as scale-up), hardware eventually needs to be replaced. As the number of smaller, discrete storage appliances increases, then so does the overhead on replacing those systems over time, with a need to manually migrate data from older to newer systems becoming part of the operational cost overhead. These tasks can become excessive and costly if hardware is replaced on a regular basis.
- **Architectural Restrictions** – some storage systems will place architectural restrictions on growth, based on physical factors (numbers of disks supported,

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

performance of the controllers, performance of internal networks, power/cooling/weight requirements). In other scenarios, the environmental aspects of physical expansion may be a problem, where for instance scaling is based on having to place new racks of storage adjacent to the existing controllers and disks.

There's a decision point that has to be made in deciding how large single systems should be allowed to grow before another discrete deployment is created, based on weighing up the above factors that needs significant architectural and operational experience to get right.

2- Scaling Models

There are two main approaches to scaling out storage infrastructure: scale-up and scale-out.

2.1- Scale-Up

The scale-up model typically expands the physical storage capacity of an appliance, but can also be used to add additional performance capacity. This design is most recognized through implementations that use a dual-controller architecture and one or more "disk shelves" that connect to both controllers.

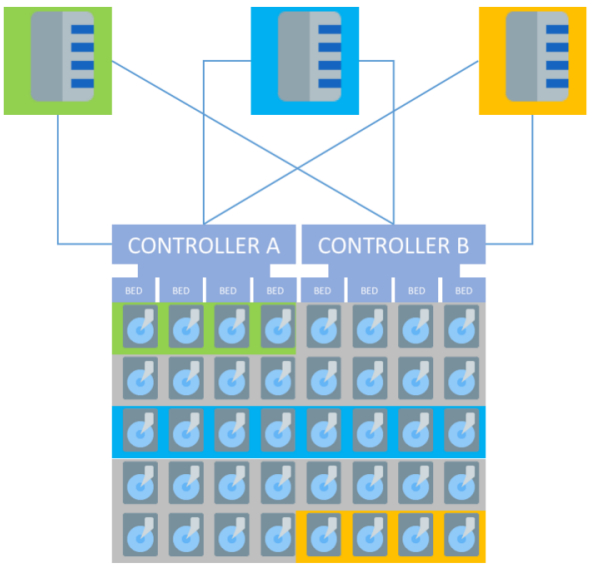


Figure 1 - Scale Up Architecture

The throughput of scale-up systems is limited by two factors: the capability of the controller and the number of backend storage devices (whether HDD, SSD or both). The capability of the controller gets improved over time by adding faster (and more) processors, more memory and faster front-end connectivity such as 8/16Gb Fibre Channel and 10GbE Ethernet. The capacity of scale-up designs is increased through adding more media, either HDD or SSD. Both increase capacity and back-end

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

throughput, but the system as a whole is limited by the maximum performance of either the controllers or the disks.

One other significant performance issue with scale-up designs is the impact on expansion to existing workloads. On newly deployed systems, the first applications will see the best level of performance as there is little contention for controller resources. Data may well be spread across the entire media and so the applications see the benefit of having access to the entire appliance. Over time, more applications are added, back-end disk capacity could be expanded and the capability of the controllers is spread ever more thinly across more data. The experience for the first applications added to the system degrades over time, with increased latency, potentially lower throughput and increased likelihood of spikes in performance due to "noisy neighbors". The solution to this problem in many instances has been to overprovision resources and have much more controller bandwidth available than is required.

2.1.1- Scale-up Strengths & Weaknesses

We can summarize the strengths and weaknesses of scale-up systems as follows.

- **Strength: Easy to implement** – From a vendor perspective, scale-up systems (especially those build around a dual-controller model) are relatively easy to implement. The architecture needs to cater for controller failure, which is usually implemented through a mirrored cache in both controllers. Older designs cater for controller failure but the lack of mirrored cache means performance can degrade significantly if a controller is lost.
- **Strength: High granularity on scalability** – scale-up systems typically allow storage capacity to be increased by adding more disks to an existing system. The minimum expansion capacity will be based on factors like data protection (e.g. adding a RAID group), but can be more flexible than some scale-out solutions.
- **Strength: Simple Fault Diagnosis** – data is stored on a fixed number of disk shelves, accessible by only two controllers, so diagnosing faults or correcting corrupted data is a relatively easy task. There's no risk of "split-brain" type scenarios that could happen with scale-out systems.
- **Strength: Easy rebuild from component failure** – the rebuild of protection (such as replacing a disk within a RAID group) is relatively light on overhead and in many systems is achieved using predictive sparing, where the failing status of a device is detected while the device is still accessible. This allows data to be moved rather than rebuilt, which is much more I/O intensive.

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

- **Strength: Relatively low media overhead for resiliency** – scale-up systems maintain one or more disks to recover from failure. As systems scale in capacity, the number of spare drives can be kept relatively small compared to overall capacity.
- **Weakness: Limited Failure Tolerance** – The tolerance of failure within scale-up systems can be limited by the RAID implementation. For example, systems implementing RAID-5 cannot tolerate more than one failure in a RAID group; systems implementing RAID-6 cannot tolerate more than two failures. Architectures are therefore designed around multiple RAID sets that are grouped together as a pool in order to get the best performance. Failure of any single RAID set however, brings down the entire pool.
- **Weakness: LUN performance decline** – over time as more capacity is added to a scale-up system, the performance of the controllers is spread over more storage capacity. The result is that the performance levels experienced by the first users of the system declines over time, unless some quality of service metrics were initially applied to that workload.
- **Weakness: Forklift upgrades** – replacing a scale-up system typically means performing a forklift upgrade - deploying a new system and implementing a project to migrate data from the old to the new hardware. The forklift upgrade carries risks because data has to be copied from one system to another and may involve application outages to achieve. The process is made more complicated when applications rely on snapshots and replication for data protection as all of those pieces need to continue to be in place during and after the move. Vendors mitigate forklift upgrades with some upgrade-in-place processes that can replace the controllers or through migration tools at the host level (like VMware Storage vMotion).
- **Weakness: Data rebalancing required** – scale-up systems can suffer from the need to rebalance data across existing media, especially as additional capacity is added to a system. In many cases this re-distribution is a manual task. Vendors have mitigated this through the use of wide striping (spreading data for a LUN or volume across many disks) or automated tiering algorithms.
- **Weakness: Systems are rarely scaled down** – a scale-up system is rarely modified to remove capacity, as the data distribution design makes it difficult to unbind disks and disconnect them from the system.

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

2.2- Scale-out (Tightly Coupled)

There are two models for scale-out storage, one of which is the tightly coupled configuration. This type of storage architecture is so named because of the close relationship between the multiple nodes that define it. Typically, a node is one or more servers grouped together, with nodes combined into a cluster. Data is either distributed or replicated across nodes.

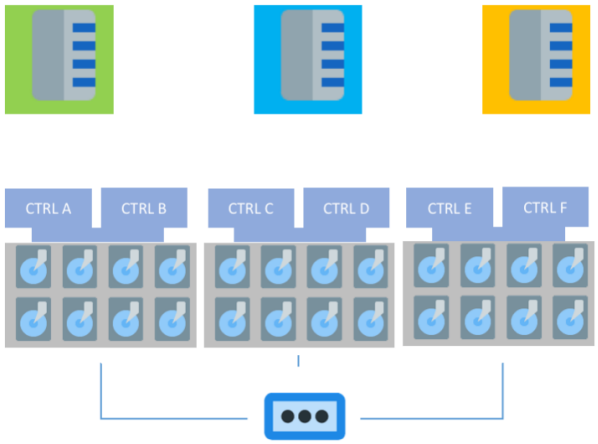


Figure 2 - Scale-out (Tightly Coupled)

System expansion within the tightly coupled model is based on the addition of nodes, which provide additional performance and storage capacity. The clustering technology for the nodes is based on a fast (sometimes proprietary) backplane, such as PCIe or Infiniband. This technology allows requests that are received from one node to be serviced by another, with I/O traffic passing across the backplane where needed.

As tightly coupled systems scale, all nodes still need to be able to communicate with each other and that creates a volume of backplane traffic that eventually places a limit on the scalability of the solution.

Data protection in tightly coupled solutions is implemented in multiple ways. Some vendors protect data within a node, making the node a highly available unit with dual controllers and battery backup/UPS capability. Data within the node is protected using standard protection mechanisms like RAID (Redundant Array of Inexpensive Disks). An alternative is to implement replication of data between nodes, creating multiple mirrors to protect against media or node failure.

2.2.1- Tightly Coupled Strengths and Weaknesses

We can summarize the strengths and weaknesses of tightly coupled systems as follows:

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

Strength: More scalable than Scale-up – tightly coupled systems are naturally more scalable than scale-up configurations as they are usually built of multiple scale-up systems put together. However, the limits of scalability are based on the two factors of intra-node and multiple-node scaling; either more capacity can be added to a single node, or more nodes can be added, both with restrictions.

- **Strength: Increased resiliency** – having more nodes provides a higher level of resiliency in some designs where the volumes can transition between node pairs.
- **Weakness: Symmetric design requirements** – some vendor implementations only support symmetric node configurations, that is a design where each node must have the same performance and capacity capabilities. In this instance, expansion of a system over time requires purchasing additional nodes, which may or may not be available from the vendor. In addition, this design puts a constraint on the amount of capacity (performance or storage) that can be added to the system and may result in over-purchasing of hardware.
- **Weakness: Unbalanced configurations** – with symmetric implementations, the scaling factor for either storage performance or capacity is fixed, meaning a configuration can't be extended to match the demand requirements of the application. Either system performance or system capacity is wasted.
- **Weakness: Low granularity on scalability** – Increasing system capacity is either based on adding extra nodes or shelves on nodes, depending on the configuration. In some cases, nodes need to be added in sets (Isilon requires sets of three, Cleversafe sets of twelve). This can be expensive depending on the architecture, if the configuration has to be kept symmetric.
- **Weakness: Nodes uptime becomes critical** – in implementations where data isn't replicated between nodes, the loss of a node becomes a critical failure and can result in data being inaccessible to all applications. This is why some vendors have over-engineered the node design, making it highly resilient.
- **Weakness: Upgrades affect the entire cluster** – upgrades typically require all nodes to be updated, making the upgrade process equally as complex (or in some cases more complex) than scale-up solutions. Vendors have implemented changes to their code that have required both disruptive and destructive outages to deploy.
- **Weakness: Forklift upgrades may still be needed** – where the limits of scalability are small (e.g. supporting only 4 to 8 nodes) the limits of capacity may be reached quite quickly and a migration to new hardware may still be needed. This is especially the case for symmetric configurations where existing nodes can't be replaced by higher capacity ones in a staged migration process.

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

2.3- Scale-out (Loosely Coupled)

The second scale-out architecture is based on loosely coupled nodes combined into a cluster. In this architecture, the nodes communicate through a shared network, such as 10GbE Ethernet. The benefit of the loosely coupled design is that nodes aren't bound by an expensive proprietary backplane that limits the level of expansion and so systems can scale to tens or hundreds of individual nodes.

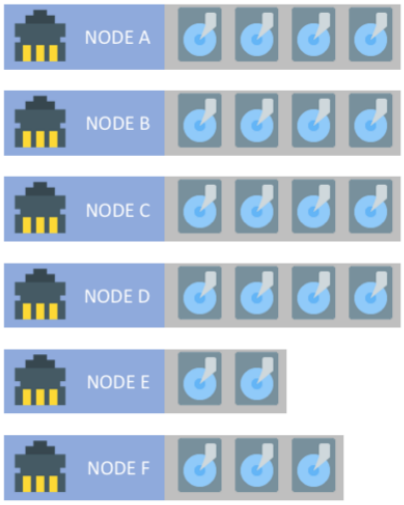


Figure 3 - Scale-out (Loosely Coupled)

One key part to the design of loosely coupled systems is in eliminating any single point of failure (SPOF) in the mechanism by which nodes communicate. This means having no central data index and no central metadata master. If this is implemented correctly, then clusters can scale linearly in both performance and capacity.

Loosely coupled systems manage hardware failure at either the disk or node level. Data is replicated between nodes, either using simple mirroring or more complex algorithms based on RAID (sometimes called network RAID) and erasure coding. If either a disk or a node fails, the data is read from elsewhere in the cluster. Resiliency is restored by recreating the lost data on other nodes.

In some implementations, storage of data at the back end is based on an object store, rather than traditional block or file systems. The format of the data is essentially separated from the protocol used to access it. This means the rebuild of lost data is simply focused on recreating failed objects and their sub-components, rather than ensuring the integrity of any file system overlays. The result is a faster and more focused rebuild process. Assuming that file systems are typically underutilized, object stores can have rebuilds that are five or more times faster.

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

2.3.1- Loosely Coupled Strengths and Weaknesses

We can summarize the strengths and weaknesses of loosely coupled systems as follows:

- **Strength: High Scalability** – loosely coupled systems offer a high degree of scalability, as there is less inter-node traffic than with tightly coupled solutions. Traffic consists mainly of replicating data and metadata because the host traffic is accessed usually through redirection of requests rather than passing data across the backplane.
- **Strength: Hardware vendor independence** – some loosely coupled systems operate entirely in user space, requiring no OS modifications. This promotes independence from particular OS's and particular hardware vendors, which enables greater flexibility and potential cost savings.
- **Strength: Asymmetric node support** – most vendors offer the ability to support multiple node sizes and types, with varied performance and capacity capabilities. This allows expansion of a cluster based on either capacity or performance requirements and to use newer, faster nodes when they become available. Some solutions support multiple node sizes, types, and vendors in a single system, simplifying long-term expansion and upgrade strategies, and enabling greater potential cost savings.
- **Strength: Node-based upgrade strategy** – upgrades can be achieved through updating software on individual nodes in a "rolling upgrade" process (assuming that is built into the design). This kind of maintenance can be achieved much less disruptively than other solutions.
- **Strength: Separate data access from data storage** – fully scale-out solutions can operate separate data and access nodes, separating the scalability of front-end data access from the back-end storage. This also allows protocol specific features to be implemented in the access layer (for instance in object stores that also support block and file).
- **Strength: Simple Migration** – many vendors support the ability to migrate nodes in and out of a loosely coupled configuration. This allows rolling upgrades to replace all nodes over time – there's no need to implement fork-lift upgrades. In addition, nodes can be evacuated of data and removed from a cluster, in order to change the configuration or design, rather than as an upgrade only process.
- **Strength: Geographic dispersion** – scale-out systems where low latency isn't a requirement can benefit from geographic dispersal of data between multiple data centres, without the need to deploy entire replicas of data. Erasure coding

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

allows data to be dispersed and recreated from a subset of the encoded components, tolerating device, system and data centre failures without the traditional overhead of double the storage capacity.

- **Weakness: Protocol support** – loosely coupled implementations typically provide limited or no support for Fibre Channel (FC) protocol, as the way in which FC is implemented with multi-pathing means data could be read & written from every node in the cluster. Solutions typically offer iSCSI (block) and HTTP/HTTPS (object). NFS/SMB (file) is typically offered through third party file system gateways, which typically add cost and an un-integrated control plane. That control plane can be a single point of failure in any type of scaled deployment.
- **Weakness: Not fully capacity optimised** – some solutions don't implement RAID or erasure coding for data protection, but instead rely on mirroring data between nodes. In some instances, this can result in a 3x capacity overhead. Mirroring can be an expensive option with all-flash systems.
- **Weakness: Implementation complexity** – implementing loosely coupled storage systems reliably can be complex and as a result we've seen few deployments in the market place outside of the object storage category. This is slowly changing as flash enables better levels of performance. Object storage vendors are now adapting their products to provide all object access protocols. Some object storage vendors are also adding file access protocols, opening the door to a wider base of applications. Some complexity issues involve managing the ongoing protection of data, rebuilding from data loss, the overhead of performing RAID/erasure coding calculations and catering for partial or intermittent node inaccessibility (e.g. in geographically dispersed deployments).

3- Choosing a Strategy

The decision on which platform works best depends on the requirements in place when making an architectural and specific vendor platform selection. In general, scale-up systems are good for smaller more simple IT environments. Loosely-coupled scale-out systems will meet the requirements of end users with high scalability needs, such as large archives, multi-workload cloud environments, or web-scale application deployments. Scale-out tightly coupled systems fit somewhere in between. Good questions to ask any potential vendor include:

- How efficient is usable capacity compared to raw capacity?
- What data protection methods are used?
- Can systems be upgraded in place without outage/disruption?

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

- What is the upgrade process?
- How does performance and capacity scale? What are the increments?
- What is the support for different form factors in the same system?
- Can capacity be removed as well as added to a cluster/configuration?
- How are data lookups handled? Is the metadata architecture fully distributed?
- How is data recovery handled? How automated is the process?
- How are different storage protocols supported, like file? How does the protocol support handle performance requirements and failure scenarios?
- How much impact do data rebuilds have on host I/O performance?

3.1- Summary

In summary, scale-up and scale-out architectures provide different features that meet the needs of scalable storage. The relative simplicity but limits of scale-up solutions contrast with loosely-coupled systems that provide high scalability, where required. Ultimately the right choice of architecture depends on the needs of the end user, both from a technical and operational perspective.

Copyright © 2009-2016 by Langton Blue Ltd. No reproduction without permission.

4- More Information

For additional technical background or other advice on the use of storage in the enterprise, contact enquiries@langtonblue.com for more information.

Langton Blue Ltd is hardware and software independent, working for the business value to the end customer. Contact us to discuss how we can help you transform your business through effective use of technology.

Website: www.langtonblue.com
Email: enquiries@langtonblue.com
Twitter: @langtonblue
Phone: (0) 330 220 0128

Post:
Langton Blue Ltd
133 Houndsditch
London
EC3A 7BX
United Kingdom

4.1- The Author

Chris M Evans has worked in the technology industry since 1987, starting as a systems programmer on the IBM mainframe platform, while retaining an interest in storage. After working abroad, he co-founded an Internet-based music distribution company during the .com era, returning to consultancy in the new millennium. In 2009 he co-founded Langton Blue Ltd (www.langtonblue.com), a boutique consultancy firm focused on delivering business benefit through efficient technology deployments. Chris writes a popular blog at http://blog.architecting.it, attends many conferences and invitation-only events and can be found providing regular industry contributions through Twitter (@chrismevans) and other social media outlets.

No guarantees or warranties are provided regarding the accuracy, reliability or usability of any information contained within this document and readers are recommended to validate any statements or other representations made for validity.

Copyright© 2009-2016 Langton Blue Ltd. All rights reserved. No portions of this document may be reproduced without the prior written consent of Langton Blue Ltd. Details are subject to change without notice. All brands and trademarks of the respective owners are recognised as such.