Viability of Small-Scale HPC Cloud Infrastructures

Emmanuel Kayode Akinshola Ogunshile
Department of Computer Science, University of the West of England, Bristol, U.K.

Keywords: RDMA Networking, TCP/IP Networking, HPC Infrastructures, Cloud Platforms, Integration, Management and Performance.

Abstract: RDMA networking has historically been linked closely and almost exclusively with HPC infrastructures. However, as demand for RDMA networking increases in fields outside of HPC, such as with Hadoop in the Big Data space, an increasing number of organisations are exploring methods of introducing merged HPC and cloud platforms into their daily operations. This paper explores the benefits of RDMA over traditional TCP/IP networking, and considers the challenges faced in the areas of storage and networking from the perspectives of integration, management and performance. It also explores the overall viability of building such a platform, providing a suitable hardware infrastructure for a fictional case study business.

1 INTRODUCTION

ALIGNING the expansion of IT infrastructure with growing requirements is a challenge for many organisations. Failure to deal with this challenge can result in IT or server sprawl: a situation in which an organisation ends up with a vast number of underutilised IT resources that can wastefully consume power and cooling resources in the data centre and become extremely difficult to support.

The inevitabilities of this are increased utility costs and excessive human and financial resources required to support this bloated infrastructure.

The reasons for IT sprawl to occur are straightforward. A small organisation may:

- Lack the budget to pre-emptively build an IT infrastructure that can cater for planned growth and scale beyond this, or experience unexpected growth, making IT an entirely reactive rather than proactive component of the organisation;
- Lack the flexibility to re-deploy their existing infrastructure to most efficiently accommodate new requirements, e.g. migrating existing infrastructure to a virtualised environment, and thus;
- Be limited to simply adding to their existing infrastructure as requirements grow.

As a practical example of this, a small, siloed organisation has limited initial requirements that may easily be fulfilled by standalone servers. Therefore, a few servers are purchased for each department and separate networks are built for each department. However, the reactive nature of IT in the organisation emphasizes the urgency of new requirements and present computing resources are likely to be mission critical, thus untouchable. All that can be done is add new resources to existing infrastructure; an approach that may be adequate for managing a few systems, but rapidly becomes unmanageable beyond this scale. The resulting infrastructure will likely have a huge number of small failure domains that will be difficult or impossible to mitigate, a wide variety of software and hardware platforms that may even require external support due to internal limitations in expertise, and may be inherently unreliable, both in terms of service availability and data integrity and accessibility. Using virtual machines on existing hosts, as opposed to running workloads on bare metal, is often considered and used as a partial and pre-emptive solution to IT sprawl. However, in itself this only mitigates utilisation issues, the initial investment cost of hardware and excessive utility cost expenditures; aside from hardware deployments, the challenges imposed by managing a vast number of machines still exist, and network management is potentially more difficult, traversing both physical and virtual environments. With regards to enterprise storage, simply running virtual machines alone does nothing to ensure the desirable attributes of security,

Ogunshile, E.
Viability of Small-Scale HPC Cloud Infrastructures.
In *Proceedings of the 8th International Conference on Cloud Computing and Services Science (CLOSER 2018)*, pages 275-286
ISBN: 978-989-758-295-0

integrity, accessibility and scalability. It could be argued that localised storage for virtual machines hampers these attributes; consolidation means fewer components and thus an increased probability for a particular piece of data to be impacted by a failure.

IT sprawl only represents only a subset of the issues faced by growing organisations. Its effects are only accentuated by our rapidly developing capabilities in generating, processing and storing data. It is almost inevitable that organisations looking to take advantage of these capabilities are going to face challenges with scale: network and storage performance, resiliency and expansion; infrastructure manageability; and operational and support costs. Cloud computing is frequently viewed as the solution to all of the previously mentioned challenges. The term *cloud computing* itself is often considered a buzzword, with many descriptions falling under a similar degree of ambiguity; it is common for generic definitions to include the consolidation of compute resources, virtualisation, simplified maintenance, and the use of remote, outsourced infrastructure among others. However, cloud computing is best defined by a core characteristic: service orientation. This is represented by the *as a Service* (aaS) models, the primary three of which are:

Infrastructure as a Service (IaaS) Delivery of typical infrastructure components as virtual resources, such as compute, storage and networking; extensions of these such as load balancers; and on public cloud platforms even virtual private clouds (VPCs) such as (Amazon Web Services, 2017).

Software as a Service (SaaS) Delivery of software applications or packages of any scale, ranging from local applications, such as Microsoft Office and its Office 365 counterpart, to multi-user, organisation-wide applications such as Salesforce.

Platform as a Service (PaaS) Provides a platform with a suite of common components, such as databases, authentication systems and interpreters for various languages, and abstracts virtually all infrastructure components, to allow developers to build web-based applications or service backends.

It's worth noting that these models are entirely independent of each other; IaaS relates only to IT infrastructure while SaaS and PaaS focus on delivery to the end user. However, there is a good degree of interoperability; Cloud Foundry serves as a strong example of a PaaS solution available on numerous IaaS platforms, such as (OpenStack, 2016).

While the issues with IT sprawl are commonly associated with more traditional IT infrastructures—frequently those supporting business processes directly performed by desktop clients—HPC environments in many organisations can be susceptible to the same issues. HPC is commonly perceived as an entirely separate branch of the computing industry, which is reasonable to an extent; modern HPC environments focus on the use of non-commodity InfiniBand (IB) fabrics for low latency, high bandwidth inter-node communication, compared with cloud environments that use commodity Ethernet networks hosting virtualised bridges, routers, virtual LAN (VLAN) and Virtual eXtensible LAN (VXLAN) networks. However, HPC environments can benefit from the flexibility that cloud computing offers; rather than being limited to a single platform and a scheduler, users can build a virtual cluster at whatever scale they deem suitable, with the software platform of their choosing. It is only more recently that technologies such as single root I/O virtualisation (SR-IOV) have made virtualising HPC workloads viable in practice, but there are still issues scaling such environments to hundreds or thousands of nodes, hence why there aren't any HPC-oriented cloud offerings from any major provider, nor any virtualised HPC environments at scale.

This paper aims to explore how a small-medium organisation could implement a hybrid HPC/cloud environment with a fictional case study business, EWU Engineering (EWU). It is arranged into the following sections:

2: Case Study Overview of EWU's current and predicted business in addition to their requirements.

3: Storage Provides a technical overview of the limitations of RAID (redundant array of independent/inexpensive disks) experienced by EWU, and technical justification for the recommended underlying file system, ZFS, and distributed storage solution, GlusterFS.

4: Hardware Recommendations Focuses primarily on the hardware suitable for implementing the solution using the technologies covered in previous sections.

2 CASE STUDY

EWU Engineering are an automotive consultancy and design business based in Birmingham, England. Established in 2011, they provides services including, but not limited to:

- End-to-end computer-aided engineering (CAE) for: safety testing; noise, vibration and harshness, durability and failure mode effects analysis using a variety of physics simulation tools such as LS-DYNA, ANSYS or others to conform with customer requirements
- Computer-aided design/modelling (CAD/CAM) of a variety of components ranging from trim pieces to entire custom mechanisms
- CNC and manual machining: milling, turning and facing for low volume or pre-production components
- Thermoplastic 3D printing for rapid prototyping
- Full project conceptual renders and designs
- Standards compliance testing on concept and existing components

The majority of their work has primarily been in high volume production projects under contracts with the likes of Jaguar Land Rover, but they occasionally do work for low volume or one-off projects, and components for motorsport teams. The business is segregated into four departments, employing forty-seven workers: the accounts and management departments have five employees each; the machining department has seven employees; the CAE/design department has thirty employees. Their current turnover is around £7 million per year; they estimate this to increase to £10 million per year over the next two and a half years as a result of increased demand for their end-to-end, concept to production-ready CAE and CAD/CAM services.

2.1 Current Infrastructure

Having grown rapidly over the past three years in particular, EWU wanted to minimise their dependence on locally hosted services and insourced IT staffing. As a result, they currently contract a local business to manage their domain and Office 365 subscriptions.

Table 1: Accounts and management departments file servers.

	acctmgmt-srv{1,2}
Chassis	Intel P4304XXSHCN
Power	2x 400W, redundant, 80+ Gold
Motherboard	Intel S1200BTLR
CPU	Intel Xeon E3-1275v2
RAM	2x8GB DDR3-1600 ECC
OS storage	2x WD Blue 1TB, RSTe RAID-5
File storage	4x WD Red 3TB, MD RAID-10, LVM+XFS
OS	CentOS 6.8
HA network card	Intel X520-DA2

The accounts and management departments share two file servers, as do the design and machining departments. The specifications of these servers are outlined in Tables 1 and 2 respectively. Such departmental pairings make sense as these departments frequently require access to the same files. Corosync and Pacemaker are used to implement high availability on these file servers and DRDB is used for real-time data synchronisation between each server. The accounts and management file servers run virtualised domain controllers for the entire business; the low load on these servers deemed them suitable for the purpose.

Table 2: Design and machining departments file servers.

	desmech-srv{1,2}
Chassis	Intel P4304XXSHCN
Power	2x 400W, redundant, 80+ Gold
Motherboard	Intel S1200BTLR
CPU	Intel Xeon E3-1275v2
RAM	2x8GB DDR3-1600 ECC
OS storage	2x WD Blue 1TB, RSTe RAID-5
File storage	10x WD Red 4TB, MD RAID-6, LVM+XFS
OS	CentOS 6.8
HA network card	Intel X520-DA2

The accounts and management file servers hold both working and archive data. The designing and machining department file servers hold documentation and data for project milestones and archived projects. Project files in progress are stored on workstations and laptops, and documentation is stored in Office 365 for collaboration within the department and with clients.

EWU's CAE/design department has the only significant computational performance requirement in the business, hence their minimalistic backend IT infrastructure. This work is performed entirely on workstations or laptops, the latter of which are only issued to employees who frequently work away from EWU's premises. In recent months the number of frequent remote workers has increased drastically. The twenty workstations have been purchased and built as and when required over a period of four

years. Examples of lower and higher tier specifications of workstations purchased this year are outlined in Table 3.

Table 3: Workstation specifications.

	Lower	Higher
Chassis	Fractal Design R4	
Power	Corsair RM550x	
Motherboard	Asus Z170-K	Gigabyte X99-UI
CPU	Intel Core i7-6700	Intel Core i7-68(
RAM	2x8GB DDR4-2400	4x8GB DDR4-2
Graphics card	Nvidia GeForce GTX950	
OS storage	Crucial MX300 275GB	
File storage	Western Digital Black 2TB	
OS	Windows 10 Pro	

The client network is wired with CAT-6, being a fairly new building, however they are using the built-in Gigabit Ethernet network interface controller (NIC) on all PCs.

2.2 Current Problems and Requirements

EWU are finding that individual workstations and laptops are no longer sufficient to meet its computational performance requirements, with many workloads taking hours to complete. During this time, employees are finding these workstations unusable for working on other tasks, further impeding productivity. Additionally, the purchase of these laptops is becoming increasingly expensive and there is much debate as to whether they are truly fit for purpose; expensive quad-core machines are required for their performance, yet this amounts to significant deficiencies in weight, size and battery life.

The business currently use their file servers for completed projects and milestones only, with most active project files being downloaded from the client companies' servers through a variety of means (dictated by the client). There is a genuine concern regarding data security, integrity and accessibility of working data following a number of hardware and software workstation failures and the accessibility of workstations and laptops to visitors of the EWU's premises. However, their current file servers do not have the capacity, nor are they sufficiently expandable to store all working data in the company. Furthermore, the client network does not have the bandwidth to adequately support all their workstations for such usage.

EWU's primary concern is that they have experienced occasional data corruption on their file servers. At worst this has required them to re-run some workloads, or retrieve copies elsewhere; they have been unable to determine the cause of the corruption. Additionally, they have found RAID array rebuilds on the design and machining departments' file servers to be slow and unreliable, with significant manual intervention being required when these rebuilds fail. Due to their highly available configurations, these servers have not experienced downtime. They used RAID-6 on these servers in order to get as much usable space as possible (32TB usable), but these servers are nearly full despite being rebuilt six months ago, with few options for expansion. They have used RAID-10 on their Exchange servers due to a lack of software RAID-6 support in Windows Server 2012, and on their accounts and management departments' servers due to the low drive counts.

EWU have realised that they are experiencing sprawl in their IT infrastructure in exactly the same way that a data centre would, the only difference being that their infrastructure is built almost entirely from client machines as opposed to servers. They are aware that their workstations are being underutilised, therefore ruling out the purchase of replacements entirely. However, they would like to maintain the flexibility that workstations offer; users should be able to spin up whatever software environment is best suited to the task at hand, with the additional benefit of scaling their environments appropriately based on the size of the task. Additionally, they would like the performance of an HPC environment, taking tasks that require over an hour to complete down to several minutes.

As mentioned in the opening of this section, EWU have expectations for significant growth over the next two and a half years. They estimate that they will need a storage solution with around 90TB of capacity for hot and warm data (accessed frequently and occasionally respectively), with the capability of expansion as and when needed. For this initial purchase, they have a budget of £100,000. They will be employing three full time staff members to support the platform, with a mix of cloud, HPC and Linux experience.

3 STORAGE

For EWU's cloud solution, storage is likely to be the area with the largest scope for variation. In this

instance, networking hardware choices are limited by the requirements of the workloads run on the cluster and their virtualisation compatibility, and the cloud platform choices are largely dictated by cost and documented capability: the underlying hypervisor may be the same regardless of platform. On the other hand, storage solutions are layered, from a conventional underlying file system up to a distributed storage solution; compatibility and combining functionality need to be considered. As a result, storage forms the bulk of this paper.

3.1 Limitations of RAID

At a single node scale, RAID (redundant array of independent/inexpensive disks) is the de facto solution for pooling disks. Traditional RAID implementations as described here function at a level in between the block devices and the file system, with the possibility of volume management layers such as Logical Volume Manager (LVM) and encryption layers such as dm-crypt being used in between. Hardware RAID implementations present a single block device to the operating system—though many RAID cards allow utilities such as smartctl to see the drives connected to them with appropriate drivers—whilst software implementations such as Linux's Multiple Device (md) RAID subsystem build a virtual block device directly accessible physical block devices. While not representative of all available RAID levels, Table 4 outlines the available common implementations that are used today. Occasionally used are nested levels -10, -50 and -60, which use RAID-0 striping on top of multiple RAID-1, -5 and -6 arrays.

RAID is a cost-effective, widely compatible method of aggregating disk performance and capacity. However, aside from the evident scalability limitations of RAID to a single machine, there are severe limitations that make all levels of RAID in particular parity RAID unsuitable for implementation in any high capacity or distributed/parallel storage solution.

3.1.1 Parity RAID

RAID-5 and -6 use distributed parity—the exclusive OR (XOR) of the blocks containing data—in order to calculate missing data on the fly in the event of disk failures, allowing the failure of any one or two disks in the array respectively. On paper, parity RAID arrays are among the most cost effective and efficient methods of aggregating disks. However, many incorrectly assume that even at low drive counts, this aggregation will compensate for any introduced overheads. A single system write operation requires the following nonconcurrent operations in a parity RAID array:

- Read data from disk
- Read parity blocks (one for RAID-5, two for RAID-6)
- Recalculate parity
- Write data to disk
- Write parity to disk (one for RAID-5, two for RAID-6)

Due to the performance of modern hardware we can consider parity calculations to be entirely inconsequential in practice. However, mechanical disk performance continues to be the single largest bottleneck in any computer system; any amplification of write operations is highly undesirable. A Western Digital Red WD80EFZX has a peak sequential transfer rate of 178MiB/s. Even at a sustained peak sequential transfer rate, theoretically it would take over 13 hours[1] to fill the disk without any additional activity; a real world rebuild will likely be significantly longer. With 4–8TiB drives now becoming common, building, rebuilding or migrating data into a large parity RAID array is infeasible. Performance serves as a significant contributing factor in the deprecation of parity RAID as a recommended solution for high capacity storage. RAID-10 is the commonly recommended alternative; with the low cost-per-gigabyte of modern drives, 50% space efficiency to mitigate the performance drawbacks of parity RAID is usually deemed acceptable. The implication of this is that the failure of a RAID-1 pair will cause the array to fail.

3.1.2 Compatibility

There is no common implementation for RAID with the standard defined by the Storage Networking Industry Association (SNIA) specifying only the functionality required of an implementation (Storage Networking Industry, 2009). This can include the encoding used; for example, while Reed-Solomon encoding is commonly used for RAID-6, some controllers use proprietary encoding schemes (Microsemi Corporation, 2008). The implication of this is that a softwarebased RAID array cannot be migrated to a hardware-based array and vice versa, and beyond controllers sharing the same controller or controller chipset families arrays cannot be migrated between different hardware RAID devices. Hardware RAID card failures will require an

equivalent replacement to be sourced. The most suitable solution in these circumstances is to use a RAID card for its battery backup and possibly caching capabilities only and to use its just a bunch of disks (JBOD) or passthrough mode while implementing software RAID.

3.1.3 Capacity Balancing

Current RAID implementations offer no capacity balancing for arrays built from varying capacity disks, with the space used on each disk matching that of the smallest disk in the array. While it is rare to mix disk sizes in this manner, it does mean that replacing disks as they fail with higher capacity disks with a lower cost per gigabyte is no longer beneficial.

3.1.4 Disk Reliability

Disk reliability is a complex subject; manufacturers provide basic reliability specifications such as mean time before failure (MTBF) and bit-error rate (BER), but these statistics don't take real-world operational conditions into account and may be outright false. While there have been a number of small scale reports on the conditions impacting disk reliability, Google's study of over 100,000 drives back in 2007 concluded that there was no consistent evidence that temperatures and utilisation resulted in increased failure rates (Pinheiro et al., 2007). Cloud storage company Backblaze Inc. provide what are perhaps the most current and regular (at least

1. $8(2^{40}/2^{20})$MiB / 178MiB per second / 60^2 (seconds to hours) = 13.09 hours yearly) reviews of hard drive reliability. Their data suggests that the most important factor in drive reliability is the drive model chosen (Klein, 2016). Looking at their separate 2015 study for the notoriously unreliable Seagate ST3000DM001 3TB disks they deployed in 2012, it can be seen that a huge proportion of these drives failed within the same time period, peaking at 402 failures in Q3 2014 (Backblaze Inc, 2017). While this isn't representative of most drives in production today, such figures suggest that the concurrent failure of multiple drives in an array—failing the entire array and leaving it in an unrecoverable state—is certainly a real danger. In short, the failure domain for a RAID array is extremely large, regardless of the RAID level in use.

3.2 ZFS

ZFS is a file system originally created by Sun Microsystems. Originally open-sourced as part of OpenSolaris in 2005, contributions to the original ZFS project were discontinued following Oracle's acquisition of Sun Microsystems in 2010 (OpenZFS, 2017). The OpenZFS project succeeds the original open-source branch of ZFS, bringing together the ports for illumos, FreeBSD, Linux and OS X (Welcome to OpenZFS, 2017). While OpenZFS and ZFS are distinct projects, the term *ZFS* may refer to either or both of them depending on context. However, there are no guarantees to maintain compatibility between the on-disk format of the two (ZFS on Linux, 2013). In this instance and indeed most instances, ZFS refers to the *ZFS on Linux* (ZOL) port. The OpenZFS project is still in its infancy, however its ZFS ports have already been proven to successfully address a large number of issues with current storage solutions.

While ZFS itself is also not scalable beyond a single node, it is an ideal choice as the underlying file system for a distributed storage solution. It will mitigate the corruption issues that EWU have experienced, offer easy low-level snapshotting and backup for huge amounts of data as they scale their storage infrastructure and can offer excellent performance even at a small scale as EWU will be building.

3.2.1 Overview

Unlike traditional file system, RAID and volume manager layers, ZFS incorporates of these features. Some ZFS primitives relevant to the discussion of the proposed solution include:

Virtual Device (VDEV) Built from one or more block devices, VDEVs can be standalone, mirrored, or configured in a RAID-Z array. Once created a VDEV cannot be expanded aside from adding a mirror to a single disk VDEV.

RAID-Z ZFS has built-in RAID functionality. In a basic configuration it has the same caveats by default. However,
the biggest difference is the capability of triple parity (RAID-Z3), with an additional performance cost still.

zpool Built from one or more VDEVs, a ZFS file system resides on a zpool. To expand a zpool, we can add VDEVs. ZFS will write data proportionately to VDEVs in a zpool based on

capacity; the trade-off is space efficiency versus performance.

Datasets A user-specified portion of a file system. Datasets can have individual settings: block sizes, compression, quotas and many others.

Adaptive Replacement Cache (ARC) In-memory cache of data that has been read from disk, with the primary benefits being for latency and random reads, areas where mechanical disk performance suffers greatly.

Level 2 Adaptive Replacement Cache (L2ARC) SSDbased cache, used where additional RAM for ARC becomes cost-prohibitive. As with ARC, the primary benefit is performance; a single decent SSD will be capable of random read I/O operations per second (IOPS) hundreds to thousands of times higher and latency hundreds to thousands of times lower than a mechanical disk.

ZFS Intent Log (ZIL) and Separate Intent Log (SLOG) ZFS approximate *equivalents* of journals; Other ZFS features include: compression, recommended for most modern systems with hardware-assisted compression usually being of inconsequential CPU performance cost with the benefit of marginally reduced disk activity; dynamic variable block sizing; ZFS send/receive, which creates a stream representation of file system or snapshot, which can be piped to a file or command (such as ssh), allowing for easy and even incremental backups.

3.2.2 Basic Operations

ZFS' on-disk structure is a Merkle tree, where a leaf node is labelled with the hash of the data block it points to, and each branch up the tree is labelled with the concatenation of the hashes of its immediate children (Fig. 1), making it self-validating.

During write operations, the block pointers are updated and the hashes are recalculated up the tree, up to and including the root node, known as the uberblock. Additionally, ZFS is a copy-on-write (CoW) file system—for all write operations, both metadata and data are committed to new blocks. All write operations in ZFS are atomic; they either occur completely or not at all.

As detailed in the following text, these three attributes are directly responsible for many of the benefits in performance and data integrity that ZFS offers.

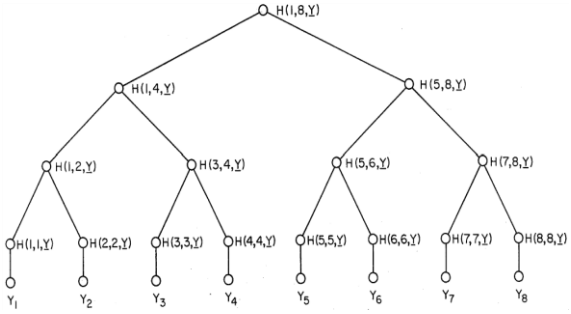

Figure 1: Merkle Tree.

3.2.3 Consistency

On modification, traditional file systems overwrite data in place. This presents an obvious issue: if a failure—most commonly power—occurs during such an operation, the file system is guaranteed to be in an inconsistent state and *not* guaranteed to be repaired, i.e. brought back to a consistent state. When such a failure occurs, non-journalled file systems require an file system check (fsck) to scan the entire disk to ensure metadata and data consistency. However, in this instance, there is no reference point, so it is entirely possible and common for an fsck to fail.

Most of the file systems used today use journaling in order to ensure file system consistency. This involves writing either metadata alone or both metadata and data to a journal prior to making commits to the file system itself. In the occurrence described previously, the journal can be "replayed" in an attempt to either finish committing data to disk, or at least bring the disk back to a previous consistent state, with a higher probability of success.

Such a safety mechanism isn't free, nor does it completely avert risks. Ultimately, the heavier the use of journalling (i.e. for both metadata and data) the lower the risk of unrecoverable inconsistency, at the expense of performance.

As mentioned previously, ZFS is a CoW file system; it doesn't ever overwrite data. Transactions are atomic. As a result, the on-disk format is always consistent, hence the lack of fsck tool for ZFS.

The equivalent feature to journalling that ZFS has is the ZIL. However, they function completely differently; in traditional file systems, data held in RAM is typically flushed to a journal, which is then read when its contents is to be committed to the file system. As a gross oversimplification of the behaviour of ZFS, the ZIL is only ever read to replay transactions following a failure, with data still being read from RAM when committed to disk. It is

possible to store replace the ZIL with a dedicated VDEV, called a SLOG, though there are some important considerations to be made, detailed in later section.

3.2.4 Silent Corruption

Silent corruption refers to the corruption of data undetected by normal operations of a system and in some cases unresolvable with certainty. It is often assumed that servergrade hardware is almost resilient to errors, with errorcorrection code (ECC) system memory on top of common ECC and/or CRC capabilities of various components and buses within the storage subsystem. However, this is far from the case in reality. In 2007, Panzer-Steindel at CERN released a study which revealed the following errors under various occurrences and tests (though the sampled configurations are not mentioned):

Disk Errors Approximately 50 single-bit errors and 50 sector-sized regions of corrupted data, over a period of five weeks of activity across 3000 systems.

RAID-5 Verification Recalculation of parity; approximately 300 block problem fixes across 492 systems over four weeks.

CASTOR Data Pool Checksum Verification Approximately “one bad file in 1500 files" in 8.7TB of data, with an estimated “byte error rate of $3*10^{-7}$".

Conventional RAID and file system combinations have no capabilities in resolving the aforementioned errors. In a RAID-1 mirror, the array would not be able to determine which copy of the data is correct, only that there is a mismatch. A parity array would arguably be even worse in this situation: a consistency check would reveal mismatching parity blocks based on parity recalculations using the corrupt data.

In this instance, CASTOR (**C**ERN **A**dvanced **STOR**age manager) and it’s checksumming capability coupled with data replication is the only method that can counter silent corruption; if the checksum of a file is miscalculated on verification, the file is corrupt and can be rewritten from the replica. There are two disadvantages to this approach: at the time of the report’s publication, this validation process did not run in real-time; and this is a file-level functionality, meaning that the process of reading a large file to calculate checksums and rewriting the file from a replica if an error is discovered, will be expensive in terms of disk activity, as well as CPU time at a large enough scale.

ZFS’s on-disk structure is a Merkle tree, storing checksums of data blocks in parent nodes. Like CASTOR, it is possible to run a scrub operation to verify these checksums. However, ZFS automatically verifies the checksum for a block each time it is read and if a copy exists it will automatically copy that block only, as opposed to an entire file.

All the aforementioned points apply to both metadata and data. A crucial difference between a conventional file system combined with RAID and ZFS is that these copies, known as *ditto blocks*, can exist anywhere within a zpool (allowing for some data-level resiliency even on a single disk), and can have up to three instances. ZFS tries to ensure ditto blocks are placed at least 1/8 of a disk apart as a worst case scenario. Metadata ditto blocks are mandatory, with ZFS increasing the replication count higher up the tree (these blocks have a greater number of children, thus are more critical to consistency).

Another form of silent corruption associated with traditional RAID arrays is the “write hole"; the same type of occurrence as outlined above but on power failure. In production this is rare due to the use of uninterpretable power supplys (UPSs) to prevent system power loss and RAID controllers with battery backup units (BBUs) to fix inconsistencies by restoring cached data on power restoration. However, the problems remain the same as Panzer-Steindel outlined in arrays without power resiliency; there is no way of determining whether the parity or data is correct, or which copy of data is correct. ZFS’ consistent on-disk format and atomic operations mean that data will either be committed from ZIL or won’t be committed at all, with no corruption taking place either way.

There are additional complexities regarding ZFS’ data integrity capabilities; Zhang, Rajimwale, Arpaci-Dusseau *et al.* released a very thorough study in 2010, finding that provided a copy was held in ARC, ZFS could actually resolve even the most extreme metadata corruption as a secondary benefit to performance, as it would restore consistent metadata on commits to disk. However, they also found that ZFS does make assumptions that memory will be free of corruption, which could result in issues for systems with faulty memory or non-ECC memory. This is beyond the scope of this paper, however the general consensus is that single-bit errors are common enough to warrant the use of ECC memory; most servers sold today do.

Table 4: Common RAID levels and theoretical performance.

				Parity		Performance	
Level	**Configuration**	**Failure tolerance**	**Usable storage**	***Blocks***	***Write operations***	***Read***	***Write***
0	block striping, 2+ disks	None	*nd*	None	N/A	*nd*	*nd*
1	block mirroring, 2 disks	1 disk	*d*	None	N/A	*nd*	*d*
5	block striping, 3+ disks	1 disk	$n(d-1)$	1 distributed	R{d,p}, W{d,p}	$n(d-1)$	*nd*/4
6	block striping, 4+ disks	2 disk	$n(d-2)$	2 distributed	R{d,p1,p2}, W{d,p1,p2}	$n(d-2)$	*nd*/6

All of this is of particular importance with the gradually reducing cost of disks and proportional reduction in power consumption as capacities increase causing many organisations to keep “cold" and “warm" data—accessed infrequently and occassionally respectively—on their primary “hot" storage appliances and clusters for longer periods of time.

3.2.5 Snapshots

LVM snapshotting allows any logical volume to have snapshotting capabilities by adding a copy-on-write layer on top of an existing volume. Presuming volume group *vgN* exists containing logical volume *lvN* and snapshot *snpN* is being taken, the following devices are created:

vgN-lvN virtual device mounted to read/write to the volume

vgN-snpN virtual device mounted to read/write to the snapshot This allows snapshots to be taken, modified and deleted rapidly, as opposed to modifying *vgN-lvN* and restoring later

vgN-lvN-real actual LVM volume; without snapshots, this would be named *vgN-lvN*, would be mounted directly and would be the only device to exist

vgN-lvN-cow actual copy-on-write snapshot volume

When a block on volume *vgN-lvN-real* is modified for the first time following the creation of snapshot *vgN-snpN*, a copy of the original block must first be taken and synchronously written in *lvN-cow*. In other words, LVM effectively tracks the original data in the snapshot at modification time, and the first modification of the block guarantees a mandatory synchronous write to disk. This is hugely expensive in terms of write performance; some tests yield a six-time reduction in performance, while others claim to have “witnessed performance degradation between a factor of 20 to 30". Furthermore, the performance degradation introduced by snapshots is cumulative—the aforementioned tasks need to be performed for each snapshot. LVM snapshots should be considered nothing more than a temporary solution allowing backups to be taken from a stable point in time.

For native copy-on-write file systems such as ZFS, snapshots are a zero-cost operation. They simply use block pointers like any other data, therefore there is no impact on performance.

3.3 Distributed Storage

Distributed storage solutions serve as a significant departure from traditional storage architectures such as storage area networks (SANs) and network attached storage (NAS). The single most compelling argument in favour of distributed storage is hyper-convergence—the deployment of storage and compute resources together on the same nodes. There are numerous advantages:

Scalability As requirements grow, storage and compute resources can be grown linearly and concurrently—just add more nodes full of drives as required. Conversely, neither compute nor storage resources may exist in excess.

Utilisation Wastefully idle compute resources, whether in compute or storage nodes, are no longer present; for EWU’s infrastructure running ZFS as an underlying file system this is particularly beneficial as any unused main system memory can be used for ARC caching.

Performance Bottlenecks for cross-protocol gateways, such as Fibre Channel to Ethernet no longer exist, and latency can theoretically be reduced due to data locality, whether from a geographically close node or from data being located on the same node.

Table 5: Initial cost breakdown for EWU's HPC-cloud platform.

Component	Selection	Cost per unit (GBP)	Line cost (GBP)
Server	SuperMicro SuperServer 6028U-TNR4T+	~1500	~1500
CPU	2x Intel Xeon E5-2690 v4 (2.6GHz, 14 core)	2133.98	4267.96
RAM	8x 16GB DDR4-2400 ECC	159.98	1279.84
OS storage	2x Intel DC S3510 120GB	117.98	235.96
L2ARC cache	Intel DC P3600 400GB	509.99	509.99
SLOG storage	Intel DC P3600 400GB	509.99	509.99
GlusterFS storage	6x WD Red WD80EFZX (8TB, 3.5", 5400rpm)	298.49	1790.94
RDMA HCA	Mellanox MCX313A-BCCT (40/56GbE)	344.52	344.52
		Node cost (GBP):	**10453.69**
		8 node cost (GBP):	**83629.52**
RDMA switch	Mellanox MSX1036B-2BRS	9905.52	9905.52
QSFP cabling (RDMA network)	8x Mellanox MC2207128-003	75.98	607.84
Ethernet management switch	Netgear XS716E	1103.99	1103.99
Ethernet management cabling	8x 3M CAT-6A	7.49	59.92
		Total platform cost (GBP):	**95306.79**

The primary argument against hyper-convergence is balancing the infrastructure to ensure that neither compute nor storage performance is negatively impacted under load. There are a huge number of factors that could influence this: CPU performance, the amount of system memory per node, the requirements of the distributed storage platform, and the client demand placed on the cluster.

For EWU's deployment, distributed storage will serve as a reliable, high-performance backing store for both OpenStack Block Storage (Cinder) bootable block devices for virtual instances and for EWU's working data. The latter of these use cases is the most critical as there is a significant performance requirement; the performance limitation for provisioning new instances will likely be the execution time of the setup process.

Ceph is the dominating open-source distributed storage platform for OpenStack deployments. The April 2016 OpenStack user survey revealed that approximately 39% of surveyed production deployments are running Ceph as the underlying storage solution for OpenStack Block Storage, versus 5% for GlusterFS. It is therefore easy and common within the OpenStack community to assume that Ceph is the de facto distributed storage solution for all use cases. However, for EWU's implementation it is largely unsuitable.

Ceph has been primarily focused around object storage. CephFS, its POSIX-compliant file system layer, only reached its first stable release in April 2016 with the *Jewel* release of Ceph. CephFS in all implementations is still extremely limited: only a single CephFS file system is officially supported and there are no snapshotting features or the use of multiple metadata servers enabled as stable features, both of which are potentially crucial as facilitators for maintaining data integrity. A more significant issue is Ceph's performance, particularly at smaller scale deployments. Due to the complexity of tuning distributed storage solutions, particularly Ceph compared to GlusterFS, many of the available performance studies are extremely inconsistent. A comprehensive study by Donvito, Marzulli and Diacono found Ceph's performance to be significantly inferior to that of GlusterFS for all workloads; GlusterFS was well over five times faster than Ceph for synthetic sequential reads and writes, with Ceph yielding 7MB/s and 12MB/s for random writes and reads respectively, compared with

406MB/s and 284MB/s for GlusterFS. While this may (and perhaps could) have improved, the aforementioned limitations ultimately limit Ceph as a viable option.

GlusterFS is the distributed storage solution of choice for EWU. Along with its superior performance to Ceph, it provides a stable release RDMA transport which can be enabled on a per-volume basis. It more closely aligns with traditional file systems than Ceph; administrators directly deal with the following primitives (detailed further in its documentation), which does require more work when creating or rebalancing volumes than Ceph:

Trusted Storage Pool (TSP) A collection of servers configured for a GlusterFS cluster.

Bricks Directory exports that are used as parts of a GlusterFS volume. Analogous to block devices in a traditional file system.

Volume An aggregation of bricks, analogous to volumes in a traditional file system.

Distributed replicated volumes are the recommended GlusterFS volume configuration for EWU; data is striped across bricks and volumes are replicated, a configuration functionally the same as RAID-10 with matching performance characteristics: write performance matches that of a single volume, read performance is an aggregation of all replicas. The most suitable method of doing this would be to have replicas on half the nodes, randomising the servers on which bricks for each volume resides. This minimises the chance of both copies being taken offline during concurrent node failures (e.g. if a common power distribution unit (PDU) fails).
This does mean that space efficiency for EWU's cluster is a very low 4:1, or four copies for every piece of data; two in ZFS and GlusterFS each. However, these copies serve different purposes: the former protects against silent corruption and the latter ensures consistent availability.

4 HARDWARE RECOMMENDATIONS

Table 5 details the node specification and total initial purchase cost of the cluster sans cabling and switching. Note that pricing for the server itself was estimated based on the cost of a barebone server configuration including just the chassis, motherboard and power supplies; actual pricing could not be obtained, and this server is not available in a barebones configuration.

The following sections justify the component selection for EWU's HPC-cloud platform.

4.1 Server

The SuperMicro SuperServer 6028U-TNR4T+ features a 2U chassis capable of holding up to eight 3.5" Serial ATA drives and four Non-Volatile Memory Express (NVMe) devices in its hot-swap bays; EWU's configuration populates all but two NVMe bays. It supports the current Broadwell Intel Xeon-E5-2600 v4 series family of processors, and features two 1000W, 80 Plus Titanium rated power supplies in a redundant configuration. SuperMicro are a top-tier hardwareexclusive vendor, unlike the likes of Dell or HPE who typically sell complete hardware/software solutions. As a result, SuperMicro's pricing is likely to be notably lower.

4.2 CPU

Computational performance of HPC systems is typically measured in gigaflops (GFlops), and benchmarked with High Performance Linpack (HPL). There are three important metrics:

Rmax The theoretical maximum performance of a system: it is the sum of CPU frequency, number of cores, instructions per cycle (16 for the Broadwell architecture), CPUs per node (2 in EWU's case) and the number of nodes.

Rpeak The peak performance achieved by the system under testing.

Efficiency Rpeak divided by Rmax, typically expressed as a percentage.

Intel offer three high core count, two way (dual processor compatible) Xeon CPUs that represent comparatively good value: E5-2680 v4 (14 core, 2.4GHz, 1075.2GFlops per node), E5-2690 v4 (14 core, 2.6GHz, 1164.8GFlops per node) and E5-2697 v4 (18 core, 2.3GHz, 1324.8GFlops per node).

Without real-world benchmarks and a genuine workload to measure against, it is difficult to exactly determine which is the best suited CPU. At best a speculative estimate can be made: under light loads, the E5-2680 v4 is better value for money. For medium workloads, the E5-2690 v4 may perform best due to its higher clock speed. For an extremely heavy number of virtual clusters, the notably higher core count of the E5-2697 v4 is likely to be the best choice. While a middle ground has been chosen, speculative comparisons such as Rmax are extremely primitive: they don't take into account the overheads from virtualisation, network communication,

platform tuning and kernel and driver versions, among others.

4.3 L2ARC and SLOG

EWU's storage solution will rely heavily on L2ARC caching to provide improved read performance, particularly for random reads; increasing main system memory quickly becomes cost prohibitive, and doesn't make sense with limited network bandwidth. Intel's P3600 NVMe SSD is capable of 2100MB/s sequential read: across 8 nodes this is more than adequate to saturate the 56Gbps link provided by the network.

SLOG devices function in place of on-disk ZIL in ZFS, preventing double writes to disk. As a result, a SLOG device will be exposed to extremely heavy write activity exclusively. As the drive is supporting mechanical disks and is only read from during recovery, performance is not a priority; the key is endurance. The P3600's stated endurance is 2.19PB of writes, around one hundred times higher than a consumer solid state drive.

4.4 RDMA Networking

At a small scale pricing between FDR IB and 40/56GbE is comparable: the chosen Mellanox SX1036 switch is around £2000 more expensive than a comparable IB switch from the same vendor. However, versatility is an unavoidable argument; RoCE v2's competitive performance against IB coupled with hardware accelerated VXLAN networking for traditional cloud computing usage makes it a more compelling solution. Furthermore, as more vendors adopt the RoCE v2 standard (currently only offered by Mellanox), HCA/NIC prices will continue to fall. The chosen ConnectX-3 adapter is around 60% of the price of comparable FDR IB adapters.

5 CONCLUSIONS

The biggest limitation for converged HPC and cloud infrastructures has been compatibility with, or the feasibility of using RDMA networking technologies within a cloud computing platform, and managing RDMA networks within the framework of the environment. While the methods described in this paper have only be standardised recently, it is clear to see that they are already reasonably mature. However, at scale new methods may be required. Over the coming years it is likely that we will see developments in multiple root I/O virtualisation (MR-IOV), allowing individual SR-IOV VFs to be shared among virtual guests, and standardised paravirtual software-based RDMA devices attached to physical RDMA interfaces.

REFERENCES

Amazon Web Services, Inc. (2017). Amazon Virtual Private Cloud (VPC), [Online]. Available: https : / / aws . amazon . com / vpc/ (visited on 05/02/2017).

OpenStack Foundation. (22nd Jul. 2016). Cloud Foundry (package), (Online). Available: https : / / apps . openstack . org / #tab = murano - apps & asset = Cloud % 20Foundry (visited on 06/02/2017).

Storage Networking Industry Association, 'Common RAID Disk Data Format Specification', Tech. Rep., version 2.19, 27th Mar. 2009. (Online). Available: https://www.snia.org/tech_activities/ standards/curr_standards/ddf.

Microsemi Corporation. (17th Mar. 2008). What are the differences between the Adaptec RAID 6 and RAID 6 Reed-Solomon?, (Online). Available: http://ask. adaptec.com/app/answers/ detail / a _ id / 15313 / ~ / what - are - the - differences - between the - adaptec - raid - 6 - and - raid - 6 - reed - solomon (visited on 26/02/2017).

E. Pinheiro, W.-D. Weber and L. A. Barroso, 'Failure Trends in a Large Disk Drive Population', in *5th USENIX Conference on File and Storage Technologies (FAST 2007)*, Feb. 2007, pp. 17–29. (Online). Available: https://research.google.com/ archive/disk_ failures.pdf.

A. Klein. (31st Jan. 2017). Backblaze Hard Drive Stats for 2016, Backblaze Inc., (Online). Available: https : / / www. backblaze . com / blog / hard - drive - benchmark - stats - 2016/ (visited on 15/02/2017).

CSI: Backblaze – Dissecting 3TB Drive Failure, Backblaze Inc., (Online). Available: https : / / www . backblaze . com / blog / 3tb - hard - drive - failure/ (visited on 21/02/2017).

OpenZFS. (27th Feb. 2017). History, [Online]. Available: http:// open-zfs.org/wiki/History (visited on 27/ 02/2017).

Welcome to OpenZFS, (Online). Available: http://open-zfs.org/wiki/Main_Page (visited on 27/02/2017).

ZFS on Linux. (20th Jan. 2013). ZFS on Linux issue #1225: Explain "The pool is formatted using a legacy on-disk format." status message, (Online). Available: https://github.com/ zfsonlinux/zfs/issues/1225#issuecomment- 12555909 (visited on 27/02/2017).