

Organically distributed sustainable storage clusters

Paul W. Poteete

Department of Computer Science and Cybersecurity, Geneva College, Beaver Falls, PA, USA

ARTICLE INFO

Keywords:

Distributed systems
Clustering techniques
Storage
Ecological design
Low-power
Organic growth

ABSTRACT

The ability to create low-cost, high-availability, moderate-performance, low-power, sustainable file storage clusters that may be organically distributed throughout an organization would allow organizations to bring data back from cloud-based providers, provide local backup solutions, create local distributed storage pods, and allow remote developing countries to have access to information and other compute resources. The Internet of Things has driven much of the development in low-power ecological systems. The emergence of these devices allowed for the creation of this research project. This research utilized the design science method to create an instantiation of this concept as a demonstrative artifact that could be powered on USB power provided from almost any source. This includes the ability for small solar arrays to provide adequate power to charge the onboard power banks, allowing for continual use over periods of power loss or darkness. This artifact was evaluated using real-time direct download from up to twentyfour workstations. During the course of the research for a period of over approximately 400 days, the artifact performed without interruption. This could be an indication that it may be possible to replace cloud-based storage with organically-distributed sustainable systems for enterprise-level use.

1. Introduction

This research investigates the ability to implement sustainable organically distributed clusters for file access within an organization. This will capitalize on low-power moderate-performance systems as an effective solution for organizational computing while maintaining a sustainable environmental impact. For clarification, the concepts of environmental impact and sustainability operate around several spheres of the product itself, production, and implementation of technical systems [1]. In this research, the environmental impact will be limited to the physical mass required in the usage of small compute systems and storage arrays versus the size and weight of their conventional counterparts in storage area networks or network attached storage, and the energy consumption variance for the respective small versus large systems. The concept of sustainability will be limited to the areas of cost, ease of maintenance and cooling, and again, energy. In this case, sustainable forms of energy are considered to be directly available methods of energy acquisition through solar panels or other renewable means.

The concept of organic distribution is based on the ability for near ad-hoc growth of data storage clusters at any location within an organization as the organization grows or sees necessity in redundancy and availability, as is similar to how organic systems grow according to their need for nutrients or space. In this research, clusters may be placed at almost any location within an organization based on current

or perceived future access needs. This is somewhat similar to self-organized ad-hoc networks based on autonomous system control [2]. Many organizations have experimented with the development of pathways and other forms of organic architecture within their organizations to render the paths of least resistance or the most appropriate design for efficient movement [3]. This form of architectural discovery was not possible due to the large amounts of power, cooling, engineering, and mass of traditional storage area networks [4].

A factor often overlooked in the deployment of production systems is aesthetics. A foundational requirement for the deployment of traditional storage clusters would be power, network access, cooling, space, and security. These conventional requirements forced the placement of storage clusters to data center environments, server rooms, or office locations with sufficient access to network resources and power. When developing an organic cluster, almost any location may be selected. This is especially true if using wireless access and solar power to provide electricity to the array. Also, due to the ability to shape these systems in several different artistic designs, deployment could be decided solely on aesthetics. The phenomenon of new low-power devices in the Internet of Things has allowed for the creation of ecological systems of various sustainable configurations [5]. These systems are often unregulated and undirected [6], but this lack of control and standardization may be the creative factor that allows for the creation

E-mail address: paul.poteete@geneva.edu.

URL: <https://www.linkedin.com/in/PaulWPoteete>.

<https://doi.org/10.1016/j.array.2022.100275>

Received 1 November 2022; Accepted 29 December 2022

Available online 2 January 2023

2590-0056/© 2023 The Author. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

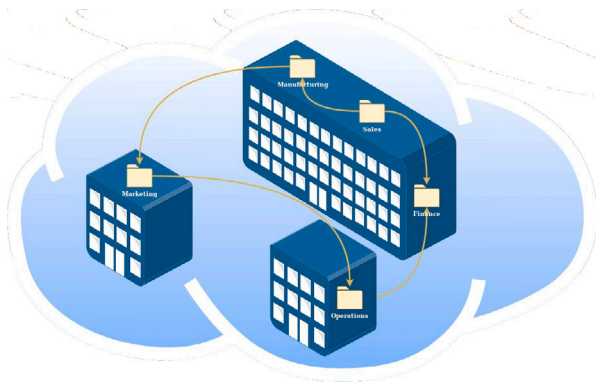


Fig. 1. Pod Distribution.

of artifacts for alternative uses that were previously unforeseen. This removes the limitations of server closet location, backbone network access, and large power requirements. Due to the distributed nature of the file systems, access may be performed over normal gigabit Ethernet or wireless connections, as it is possible to limit the number of workstations that access each storage pod. The limited power requirements allow for solar panels, USB charging stations, or normal wall power solutions.

To clarify the descriptive conventions used within this research, the terms: pod, node, and cluster should be defined. A pod is a collection of individual compute nodes in a single location. A node is a small computational system, such as an AtomicPi, that allows for client access. The term cluster can be used for the pod or the entire organization of pods that provide distributed access. An example of pod distribution across an enterprise is provided in Pod Distribution Fig. 1.

Physical security is always a concern for data centers, server and engineering closets, and other areas where computational resources are deployed. This takes an additional dimension for Internet of Things devices [5]. Traditionally, if a malicious actor gained access to the compute cluster, they could access the storage or destroy the systems, causing catastrophic failure. The low cost of deployment of pods of organic clusters, combined with their ability to replicate full data storage over hundreds of devices, prevents the catastrophic loss of access in many cases. If five of ten pods are damaged, then the load for the organization is automatically distributed to the remaining pods. Each node storage can be encrypted using 256 bit encryption within distributed or protected key centers. The key centers can be located on almost any system, from wifi-enabled ESP8266 ESP12F modules to a cloud-based solution. In this case, malicious attackers could steal an entire pod, and no information would be accessible due to the loss of the encryption keys. The entire pod could be a loss of less than \$1000 USD.

The cost, power, and access required for an organically distributed cluster is proposed to be within the budget of almost any organization or individual. This would open the viability of these systems to any organization that desires decentralized file storage, backup storage, or anyone who desires to move information from a cloud-based solution to an in-house controlled environment. As this is a small-scale test, the cost and hardware will be minimal; however, the cost is directly related to the scale of the solution, resulting in a continuous low-cost solution. This unit proposal cannot require ongoing maintenance or support during the course of the year before analysis. This will be similar to the ability to place data within the cloud with minimal administrative overhead. The cloud similarity also introduced the need for a web-based file access solution to allow users to access the files with minimal effort.

In the following sections, a Design Science Research Methodology (DSRM) is undertaken to create a viable instantiation of this concept.

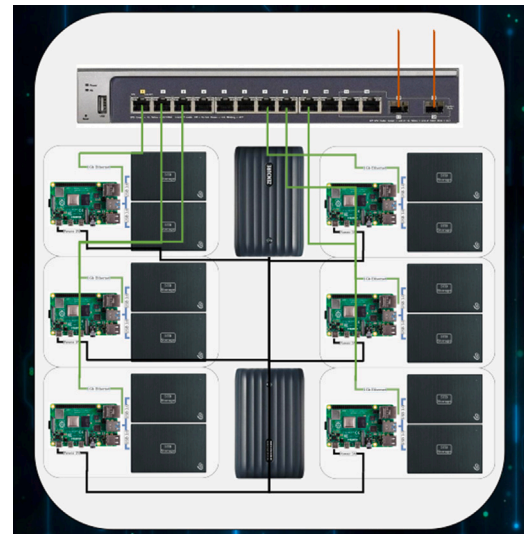


Fig. 2. Proposed Pod Configuration.

This artifact will be analyzed according to conventional usage, and common analysis tools available to any network or system administrator against a five year old existing storage area network. These tests will be run live over the period of one year in a production environment with the next analysis occurring at approximately 400 days. Afterwards an evaluation will be provided for discussion and further recommendation.

2. Materials and methods

2.1. Materials

The research required the planning and development of an artifact that will be used as a physical instantiation of the organically distributed clustering concept as seen in Proposed Pod Configuration Fig. 2. Software, Operating Systems, Hardware, and beneficial system services were selected based largely on cost, ease of administration, reliability, and performance. Every software-based solution was selected as open source and free software, and hardware components were selected to be inexpensive and self-contained. In each case, the systems, software, and hardware were selected for this research artifact as examples. They do not constitute a recommendation in and of themselves for future deployments or production use. The goal of this research is to investigate the viability of organically distributed clusters for data access within an organization.

2.1.1. Power

Power was a primary concern with the solution that this research attempted to achieve through the organic cluster artifact. The entire sustainable organic cluster node was powered using four Pofesun 30 Watt USB fast chargers connected to four Zendure X6 USB-C portable power banks. The power banks were rated at a maximum output of 45 Watts. Each of the systems required from .5 to 1.7 AMPs at 5 V, requiring a maximum power output of 39 Watts approximate usage: $5V \times ((2 \times .5 A) + (4 \times 1.7 A))$. This power requirement is less than what is provided by a small consumer solar panel array, such as the BLUETTI SP120, which outputs approximately 120 Watts of power. This places the organic cluster well within reach of a solar-powered environmental scope. The relationship between the power, production, maintenance, and cost would also be a consideration that allows this system to be within a sustainable realm [4].

2.1.2. Clustering software

CEPH and Gluster were evaluated for use within this project, and Gluster was selected. Although both solutions are good candidates, this selection was nontrivial, as CEPH had been used in a prior project and brought a level of familiarity to this research. In the end, Gluster was selected as it (1) offered almost unlimited scalability, which is a requirement for organic growth, (2) it provided a more straightforward approach for this research, and (3) it had not been implemented in earlier labs, allowing this research to be focused on a purely original environment. It should be mentioned that research was performed in prior years with CEPH on an unrelated project, and it was found to be a solid performer in all areas of clustering; however, this was at a small scale, and there is research that suggests limitations in its scalability [7].

2.1.3. Operating systems

Gluster performs well under several Linux distributions [8]. In some cases, there are reports of successful implementations of Gluster in BSD UNIX environments, as well [9,10]. Although not directly indicated in the literature, Linux Mint required less configuration and was selected for its robust reliability, low RAM usage, almost out-of-the-box compatibility with Gluster and the related services, which is a characteristic of its extensive application library based on Debian and Ubuntu repositories. Ubuntu was not selected due to its incorporation of netplan network services which provide for difficulty in defined or predictable interface names [11] and snapd application services that conflicted with initial losetup drive creation and configuration. FreeBSD was considered, but this was removed as this research desired to provide a web portal with the Gluster services that both had limited success with a standard BSD installation. In all cases of BSD UNIX and different distributions of Linux, it is believed that a usable and stable solution could be developed when provided additional time and skill.

2.1.4. System hardware

Several single board computers (SBCs) were considered for this research. The goal was to provide the least expensive, highest performance system, that would be compatible with the network architecture, software, and services required. The minimum requirements for Gluster were 2 CPU's, 2 GB of RAM, and 1 GbE [12]. In order to provide maximum compatibility, the AtomicPi x86 single board computer was selected. In addition to the Gluster requirements, it supported 5v power, x86 architecture software, onboard storage, as well as USB 3.0, which is common among Raspberry Pi systems and other SBCs. Details on the AtomicPi can be found here: <https://digital-loggers.com/api.html>. It is important to note that any ecological and sustainable system could be used for this research, and no single system would affect the underlying principle of organically distributed clusters. The research system was selected as an example only, not necessarily as a recommendation for future research.

2.1.5. Network hardware

The systems used physical Ethernet links at 1 Gbps speeds over a 5 port D-Link gigabit switch (DGS-105). This switch was powered by the same USB power source as the AtomicPis and Raspberry Pis using a conversion cable by TENINYU. This cable converted USB power to a standard barrel plug that worked with the D-Link switch without modification. The switches were configured for redundancy through the FortiGate firewall that was the gateway device that provided access for the sustainable cluster to the Internet and Lab computers.

2.1.6. System services

The services required for this form of distributed cluster range from Domain Naming Systems (DNS) to the web application front-end. The following services were used in the creation of the organically distributed cluster. Additional information on the configuration is provided in the Artifact Development subsection.

System Services:

- Domain Naming System, bind9 (DNS)¹
- Load-Balancing (LB)²
- Secure Shell (SSH)³
- Web Server (NGINX)⁴
- File Access Front-end (Cloud Commander)⁵

2.2. Methodology

2.2.1. Design Science Research Methodology

Design Science is the pursuit of research for both deliberate design methods and the scientific annotation of the processes and results for the betterment of the scientific community [13]. In similitude of qualitative or quantitative spheres that examine the existential, design science investigates pre-existential concepts into realization, through creation and innovation. The Design Science Research Methodology (DSRM) provides a process to create a solution based on a set of needs, called an artifact. Artifacts are critical to the design science research process, as they provide the created form of a solution. In this way, the artifact is an instantiation of the concept described as a solution for the need or challenge [14,15]. Throughout the process, the artifact will follow a design, evaluation, revision process until a conclusive outcome can be determined or until the research is concluded. In this, design science research combines exploratory and experimental methods to produce an artifact that provides the opportunity for descriptive and explanatory methods.

2.3. Artifact development

2.3.1. Clustering software

The initial iterations of the primary artifact were developed using the GlusterFS [8] system on (1) virtual machines (VMs), and (2) physical lab computers. After verification that gluster would work in a Linux Mint VM environment without issue, 24 physical machines were constructed and deployed in a laboratory. Several reliability tests were conducted by disabling between 1 and 10 systems in the cluster without issue. GlusterFS was able to recover without external intervention on each attempt. This validation allowed for the creation of the low-power organically distributed artifact to proceed.

2.3.2. Operating systems

Linux Mint 20 was installed on each of the four AtomicPis using an external USB flash drive. After installation, the storage was formatted using XFS. XFS was selected as it reportedly performed well with large file sizes and multi-threaded installations [16] on lower-performance systems. SSH was enabled and public private key pairs were created for remote access to the devices. These keys or new keys could be used for the LUKS cryptfs file system, if desired. GlusterFS was then installed with a single brick per node.

2.3.3. System hardware

To create the primary artifact, the AtomicPi units were assembled into a pod arrangement on a 24" by 36" wall-attached shadowbox as shown in Artifact Prototype Fig. 5. A PNY 480 GB Solid-State Drive (SSD) was attached to each AtomicPi via the USB 3.0 port, using a SATA3 to USB3 adapter. All of the power to every unit within the pod, including: the AtomicPis, network switches, Raspberry Pis, and pass-through batteries, was provided via 5v USB. The equipment was attached to the shadowbox using VELCRO® or similar hook and loop fasteners. This would allow for quick replacement of any component, if the need arose.

¹ DNS, <https://www.isc.org/bind/>.

² LB, <https://docs.fortinet.com/document/fortigate/6.0.0/handbook/154107/basic-load-balancing-configuration-example>.

³ SSH, <https://www.openssh.com/>.

⁴ NGINX, <https://www.openssh.com/>.

⁵ Cloud Commander, <https://cloudcmd.io/>.

Pass 01:
 Hosts Destination
 2 (10.10.51.21)
 3 (10.10.51.23)
 8 (10.10.51.22)
 10 (10.10.51.24)
 Pass 02:
 Hosts Destination
 5 (10.10.51.22)
 5 (10.10.51.24)
 6 (10.10.51.21)
 7 (10.10.51.23)
 Pass 03:
 Hosts Destination
 4 (10.10.51.21)
 5 (10.10.51.23)
 7 (10.10.51.22)
 7 (10.10.51.24)

Fig. 3. Round-Robin Verification.

2.3.4. System services

FreeBSD 12.1 was installed on the two Raspberry Pis for DNS after an initial attempt to install OpenBSD. The initial OpenBSD installation required numerous modifications that opened the installation to a greater risk of configuration errors. FreeBSD 12.1 was easily installed and worked with minimal configuration modifications, allowing the primary concentration to be on the purpose of the installation, which was DNS. The DNS servers were created in a split-brain configuration with round-robin functions as shown in Round Robin Verification, Fig. 3 for internal clients and a single designated external Internet Protocol (IP) address for the external clients. The single external IP address was configured to allow the FortiGate Load-Balancer to direct external hosts to available internal nodes using the FortiGate LB algorithm. Secure shell was enabled on the four AtomicPis, two Raspberry Pis, and the FortiGate firewall with key-based authentication.

2.3.5. Online file services

File access was provided via SSHFS and a web application front-end called, “Cloud Commander” which was designed to allow easy access to files over the web. This application imitates a local file explorer view with drag and drop capabilities. The foundational web server selected was NGINX for its low overhead and speed on low performance systems. Cloud Commander running on NGINX, allowed this artifact to be designed as a proof-of-concept inside and outside the class laboratory for validation of this research case. The VirtualHost function was used to create and redirect the Cloud Commander web application to the main page of the web server. This allowed easy access via a single URL to the file manager or explorer view. Each cluster node was named “vault[n]”, where “n” represented the number of the node between the numbers 1 and 4. A small visualization of Cloud Commander is represented in Fig. 4. The sustainable cluster was put into full production for over 400 days as a research experiment and classroom resource. One of the cluster units was rebooted at 157 days into the experiment to verify that the cluster would self-heal without external interaction. This configuration was employed both internally and externally for students and faculty for research and production without interruption.

2.4. Measurement

2.4.1. Overview

Performance testing was performed with a curl script directed to a download link in the Cloud Commander file repository. This allows for an accurate view of actual user download times to be simulated. The download was performed across 24 systems simultaneously to

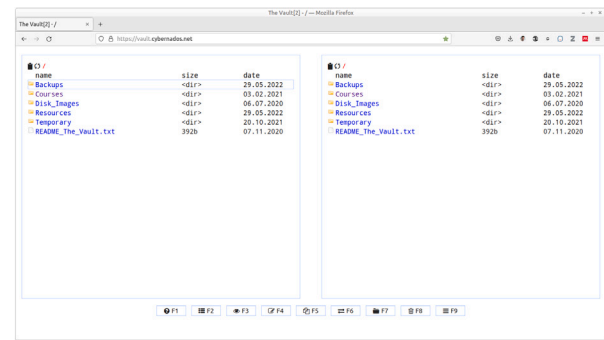


Fig. 4. Example of the Cloud Commander Web Application.

increase the network load and to test the round-robin load-balancer in DNS. After the research period, each node was rebooted with a five minute interval. The interval would allow the automatic discovery and synchronization of each rebooted node with the entire cluster pod. After the reboot, another performance test was initiated for comparison with the beginning and end evaluations. This process created an *initial* measurement at the beginning, a *final* measurement for the end of the year evaluation, and a *revisited* measurement for when the systems were rebooted after the year evaluation was completed.

2.4.2. Performance: Initial

After the construction of the shadowbox, the configuration of the operating systems and services, and the physical cabling was completed, and initial performance diagnostics were executed. This diagnostic would serve as the baseline for the next diagnostic test in approximately one year.

2.4.3. Performance: Final

After more than a year (463 + days) had passed, the final diagnostic was performed on the unmodified cluster. This test was performed using the same method as the initial diagnostic. Prior to the test, an uptime command was run against the cluster to verify the uptime and overall utilization of each node.

2.4.4. Performance: Revisited

After the final diagnostic was completed on the unmodified cluster, a reboot was conducted to provide a fresh platform for analysis. This test was performed using the same method as the initial and final diagnostics. Prior to the test, an uptime command was run against the cluster to verify the uptime and overall utilization. It should be noted that only 23 systems were available to download the diagnostic image from the cluster at the time when the cluster was rebooted.

3. Results

3.1. Overview

For over 400 days, the organic cluster operated without outage or required maintenance. The onboard USB pass-through power banks allowed the cluster to operate through more than three site-wide power outages that affected all of the other building systems. According to availability, each of the components are fully functional and do not require any form of maintenance to maintain operation. During the course of the past 400 + days, 9983 + files have been accessed both internally and externally on the sustainable cluster. The cluster currently stores 135 GB of data, replicated across four nodes.

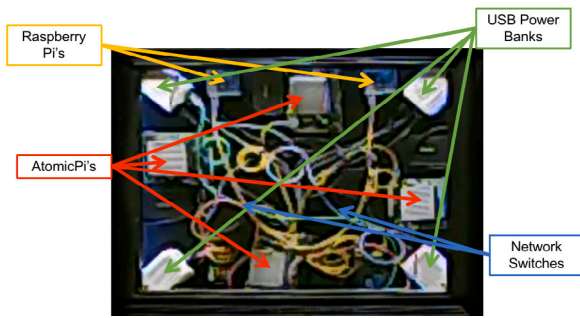


Fig. 5. Artifact Prototype.

Table 1

Total Cost per Cluster Pod — 2 TB Raw.

Description	Quantity	Cost each	Subtotal
ShadowBox	1	\$37.00	\$37.00
AtomicPi's	4	\$50.00	\$200.00
Network switches	2	\$30.00	\$60.00
Power banks	4	\$81.00	\$324.00
Cables	14	\$10.00	\$140.00
480GB SSDs	4	\$45.00	\$180.00
USB power	4	\$18.00	\$72.00
			\$1013.00

Table 2

Total Cost per Cluster Pod — 16 TB Raw.

Description	Quantity	Cost each	Subtotal
ShadowBox	1	\$37.00	\$37.00
AtomicPi's	4	\$50.00	\$200.00
Network switches	2	\$30.00	\$60.00
Power banks	4	\$81.00	\$324.00
Cables	14	\$10.00	\$140.00
4TB SSDs	4	\$300.00	\$1200.00
USB power	4	\$18.00	\$72.00
			\$2033.00

3.2. Location

The final artifact prototype was unobtrusive, residing in a shadow-box that was placed on a wall outside of a faculty office in a high-traffic area. This placement allowed for the possibility of student tampering or other unintentional interaction. This placement was intentional, as this supported the near universal placement options available for corporate data storage. A picture of the artifact is provided in Artifact Prototype, Fig. 5.

3.3. Cost

A pod reflects approximately the same cost, in USD, as a single traditional Network Attached Storage (NAS) unit; however, a pod consists of at least four redundant nodes and is scalable across an entire enterprise. The cost breakdown is listed in the tables: Total Cost per Cluster Pod — 2TB Raw Table 1 and Total Cost per Cluster Pod — 16TB Raw Table 2.

3.4. Availability

The performance of the cluster for the Initial: Cluster Uptime Fig. 6, Final: Cluster Uptime Fig. 7, and Revisited: Cluster Uptime Fig. 8 intervals represented a variation in performance that may be attributed to the long duration of activity without a reboot. This variability is visualized in the performance-duration chart: Performance in Mbps per Workstation Table 3. The total Mbps remained in the middle to low average rate due to a limitation in the AtomicPi Ethernet adapter transfer rate, a driver issue, or a combination of other factors. Note:

```
[OS:Linux Date:20200223.1817hrs EST IP:10.10.51.106]
+-[root@cybbak01]-[/organic]
+-# for ip in `cat list.vault`; do printf "$ip\t"; ssh root@$ip 'uptime'; done
10.10.51.21 18:17:35 up 8 days, 2:16, 0 users, load average: 0.05, 0.10, 0.06
10.10.51.22 18:17:36 up 8 days, 2:16, 0 users, load average: 0.03, 0.17, 0.09
10.10.51.23 18:17:37 up 8 days, 2:15, 0 users, load average: 0.01, 0.06, 0.03
10.10.51.24 18:17:37 up 8 days, 2:14, 1 user, load average: 0.11, 0.18, 0.09
```

Fig. 6. Initial: Cluster Uptime.

```
[OS:Linux Date:Thu,20220602.0600hrs FreeRAM:275M IP:10.10.51.24]
+-[root@cybbak01]-[/]
+-# for num in 1 2 3 4; do printf "$ip\t"; ssh 10.10.51.2$num 'uptime'; done
10.10.51.21 06:00:21 up 463 days, 9:50, 0 users, load average: 0.00, 0.00, 0.00
10.10.51.22 06:00:22 up 463 days, 9:49, 0 users, load average: 0.00, 0.00, 0.00
10.10.51.23 06:00:23 up 463 days, 9:48, 0 users, load average: 0.00, 0.03, 0.00
10.10.51.24 06:00:23 up 306 days, 5:45, 1 user, load average: 0.12, 0.07, 0.01
```

Fig. 7. Final: Cluster Uptime.

```
[OS:Linux Login:Fri,20220603.0423hrs FreeRAM:161M IP:10.10.10.101]
+-[root@workstation]-[/]
+-#
10.10.51.21 04:31:01 up 19 min, 0 users, load average: 0.09, 0.05, 0.03
10.10.51.22 04:31:02 up 13 min, 0 users, load average: 0.00, 0.03, 0.05
10.10.51.23 04:31:03 up 9 min, 0 users, load average: 0.18, 0.15, 0.10
10.10.51.24 04:31:03 up 1 min, 0 users, load average: 0.65, 0.48, 0.19
```

Fig. 8. Revisited: Cluster Uptime.

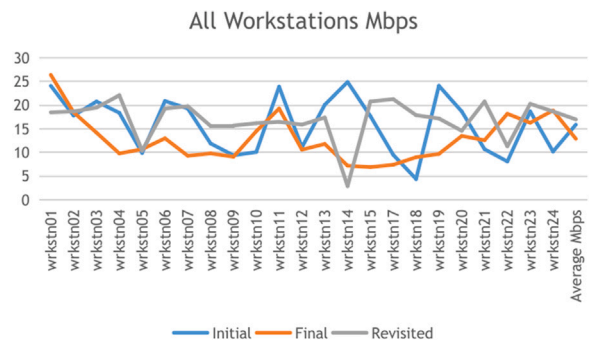


Fig. 9. All Workstations.

a single node, workstation 16, was removed from the analysis as that workstation was unavailable for the re-visitation of the performance measurement. This normalized the output to be consistent with the new 23 workstation analysis versus the original 24 workstation analysis.

3.5. Performance

Each workstation speed varied between each phase of measurement as indicated in All Workstations Fig. 9.

The initial and revisited phases offered the best performance, while the final phase was approximately 20% lower than the initial and revisited phases as seen in the Performance in Mbps per Workstation Table 3.

This degradation in performance seems directly related to the long duration between reboots of the organic cluster. This is more apparent when viewing the average readings for all of the workstations. The average over time for the initial, final, and revisited throughput was 15.3 Mbps. This average provides a variable performance percentage for each phase of the research, allowing for a clear view of the overall characteristics of the long duration between reboots. Based on the total average, the initial performance was rated at 104%, final performance was rated at 84%, and revisited performance was rated at 111%. This is also visible in the Average Mbps Fig. 10.

4. Conclusion

Based on the power requirements, physical presence, performance, and reliability of this artifact, an environmentally-friendly viable storage cluster is possible. This would provide an alternative to cloud-based

Table 3
Performance in Mbps per Workstation.

Workstation	Initial	Final	Revisited
wrkstn01	24.1	26.4	18.5
wrkstn02	17.8	18.5	18.7
wrkstn03	20.8	14.2	19.5
wrkstn04	18.4	9.8	22.1
wrkstn05	9.9	10.7	10.4
wrkstn06	20.9	13	19.3
wrkstn07	19.3	9.3	19.8
wrkstn08	11.9	9.8	15.6
wrkstn09	9.4	9.1	15.7
wrkstn10	10.1	14.5	16.2
wrkstn11	23.9	19.3	16.5
wrkstn12	11.1	10.6	15.9
wrkstn13	20.1	11.8	17.4
wrkstn14	24.9	7.2	2.9
wrkstn15	17.7	6.9	20.8
wrkstn17	9.5	7.4	21.3
wrkstn18	4.4	9	17.9
wrkstn19	24.1	9.7	17.2
wrkstn20	18.7	13.5	14.6
wrkstn21	10.7	12.6	20.8
wrkstn22	8.1	18.2	11.3
wrkstn23	18.7	16.3	20.3
wrkstn24	10.2	18.9	18.7
Average Mbps	15.9	12.9	17
Total Mbps	364.7	296.7	391.4

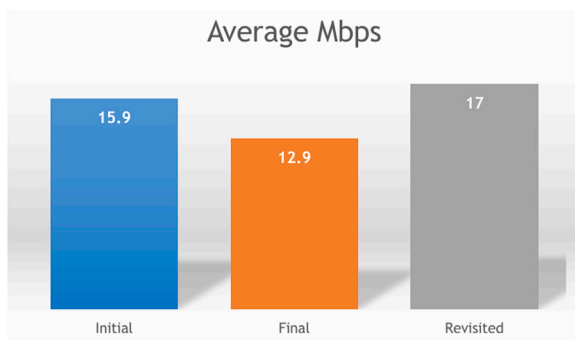


Fig. 10. Average Mbps.

data access, third-world library storage, off-grid computer resources, or other computational resources for areas with power fluctuations or limited capital for file storage. This would allow for near ubiquitous placement throughout an organization, based on individual needs or design requirements.

The ability for each compute node within the cluster pod to provide its own Redundant Array of Independent/Inexpensive Disks (RAID) configuration would be similar to each node operating as an individual Network Attached Storage (NAS) device. In this way, each pod would contain four to eight individual NAS arrays at the cost of approximately a single array or two arrays. Beyond cost, the ability to harness the clustering function of multiple individual storage nodes allowed each pod of four nodes to be fully redundant within itself. Although there were no failures for the 400 + days of research, during the first 180 days, a failure was simulated in the fourth node to test synchronization. The entire pod synchronized without issue, allowing for the next 220 days to continue naturally without interruption. The availability of the pod was also tested through three separate power outages that were unplanned for the building in which the lab was located. During those power outages, all other systems, except the sustainable cluster, failed due to lack of power after UPS batteries were depleted. The pod was able to function for the 8–24 h period on its power banks without any loss of power.

The performance of the sustainable cluster was approximately 350 Mbps total as referenced in Performance in Mbps per Workstation

Table 4
Pod distribution based on location.

PodSeq/Nodes	DNS/Share	File system
pod01/8 nodes	marketing.xyz.com	/mnt/mark
pod02/6 nodes	sales.xyz.com	/mnt/sale
pod02/4 nodes	operations.xyz.com	/mnt/oper
pod02/4 nodes	finance.xyz.com	/mnt/fina
pod02/4 nodes	manufacturing.xyz.com	/mnt/manu

Table 3. This is not a limitation of the organic configuration, but a characteristic of the AtomicPi Ethernet throughput with the default Linux driver configuration. It should be possible to achieve rates of 980Mbps per node when exchanging the core hardware for higher performance devices. These devices are already available; however, their cost is approximately three to four times greater than the \$40–\$50 USD AtomicPis, and thus they were excluded from this research. Due to the speed limitations, larger file transfers were problematic, as several workstations would quickly utilize all of the available bandwidth to an individual pod. This may be somewhat remediated through the use of USB 3 wireless transmitters, the incorporation of higher performance onboard network interface cards, or better configuration of the Linux device drivers on the system.

The ability to place cluster pods at different locations around an organization allows for rapid access to files for each department; however, the need for advanced DNS management would be essential for a base level of redirection in the case of a failure. This could also allow for proper sizing and file system hierarchy development as shown in reference Pod Distribution Based on Location [Table 4](#). This redirection would allow for any department that experiences a failure to access their files on another pod in another location without an outage. There could be a performance issue, if both departments were to access their files from a single pod during the outage, but there would not be any discernable impact to overall availability.

Changes in current technological capabilities and the increased need for local data control and availability may drive a move from cloud storage to local or hybrid storage solutions. This may not be attributed to a systematic or organized move from cloud computing due to privacy, cost, or poor performance, but the opportunity to have secure, local access, of organizational data at local facilities. Developing countries may also continue to benefit from the development of ecological systems that use alternative power sources. These opportunities, coupled with the evidence that some organizations are concerned with the security of their data within the cloud [17,18], may drive another technical migration of data back to the organizations from whence it was removed.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- [1] Mocigemba D. Sustainable computing. *Poiesis Praxis* 2006;4(3):163–84. <http://dx.doi.org/10.1007/s10202-005-0018-8>.
- [2] Herrmann K, Geihs K. Self-organization in mobile ad hoc networks based on the dynamics of interaction socio-aware applications. *Tech. rep., Berlin University of Technology*; 2003.
- [3] Gargiani R. Rem Koolhaas/OMA: The construction of merveilles. EPFL Press; 2008, p. 343, URL https://books.google.com/books/about/Rem_Koolhaas_OMA.html?id=ZbFyL_9jsFgC.

- [4] Reddy VD, Setz B, Rao GSV, Gangadharan GR, Aiello M. Metrics for sustainable data centers. *IEEE Trans Sustain Comput* 2017;2(3):290–303. <http://dx.doi.org/10.1109/TSUSC.2017.2701883>.
- [5] Li F, Shi Y, Shinde A, Ye J, Song W. Enhanced cyber-physical security in internet of things through energy auditing. *IEEE Internet Things J* 2019;6(3):5224–31. <http://dx.doi.org/10.1109/JIOT.2019.2899492>.
- [6] Al-Qaseemi SA, Almulhim HA, Almulhim MF, Chaudhry SR. IoT architecture challenges and issues: Lack of standardization. In: 2016 future technologies conference (FTC). 2016, p. 731–8. <http://dx.doi.org/10.1109/FTC.2016.7821686>.
- [7] Donvito G, Marzulli G, Diacono D. Testing of several distributed file-systems (HDFS, ceph and glusterfs) for supporting the HEP experiments analysis. *J Phys Conf Ser* 2014;513(TRACK 4). <http://dx.doi.org/10.1088/1742-6596/513/4/042014>.
- [8] Gluster Docs. Community packages - gluster docs. 2021, URL <https://docs.gluster.org/en/main/Install-Guide/Community-Packages/>.
- [9] Sellens J. Reliable replicated file systems with GlusterFS. In: *USENIX LISA* 28. 2014, p. 37.
- [10] Morante D. Setup a three node replicated GlusterFS cluster on FreeBSD | Unibia.net. 2021, URL <http://www.unibia.com/unibianet/freebsd/setup-three-node-replicated-glusterfs-cluster-freebsd>.
- [11] Unkilbeeg. Netplan for “predictable” interface name : linuxadmin. 2021, URL https://www.reddit.com/r/linuxadmin/comments/nshgl7/netplan_for_predictable_interface_name/.
- [12] Gluster Docs. Setting up on physical servers - Gluster Docs. 2022, URL https://docs.gluster.org/en/v3/Install-Guide/SetupBare_metal/.
- [13] Fuller RB. Everything I know. 1975, <http://dx.doi.org/10.2307/j.ctt1bkm5kc.4>.
- [14] Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S. A design science research methodology for information systems research. *J Manage Inf Syst* 2007;24(3):44–77. <http://dx.doi.org/10.2753/mis0742-1222240302>.
- [15] Vaishnavi V, Kuechler W. Introduction to design science research in information and communication technology. In: *Design science research methods and patterns-innovating information and communication technology*. Taylor & Francis Group; 2008, p. 7–30.
- [16] Red Hat Inc. How to choose your red hat enterprise linux file system - Red Hat customer portal. 2020, URL <https://access.redhat.com/articles/3129891>.
- [17] Tabrizchi H. Threats , and solutions. *J Supercomput* 2020;9493–532.
- [18] Kresimir P, Zeljko H. Cloud computing security issues and challenges tetraconv view project BusinessLogicIntegrationPlatform view project kresimir popovic siemens 4 publications 143 CITATIONS cloud computing security issues and challenges. In: *Ieeexplore.Ieee.Org*, Vol. June. 2010, p. 7, URL <https://www.researchgate.net/publication/224162841>.