SDC 20

BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

Real World Experiences with Storage System High Availability

Jody Glider (j.glider@sap.com)
Harsha Ravuri
Ashish Mahajan
Frank Schmitt
@SAP

What is SAP® ?

101,150+

Employees

440,000+

Customers

€27.55B

Revenue

180+

Countries

25

Industries

€4.74B+

R&D spend

SAP provides business solutions, for example:

Software

- HANA (in-memory database)
- S4HANA (ERP)
- C4HANA (Customer experience)
- SAP Data Intelligence (ML and AI for enterprise use cases)

Cloud IaaS/PaaS

- HANA Enterprise Cloud: HANA and infrastructure hosting
- SAP Cloud Platform: Includes HANA and many services to aid building of customer apps

Cloud SaaS

- Concur (travel and expense)
- Ariba (procurement and supply chain)
- Fieldglass (workforce management)
- Success Factors (HR)
- Callidus (sales execution)
- Hybris (customer experience management)

What are Global Cloud Services?

Global Cloud Services (GCS)

GCS is the core of SAP's cloud success. Our experience with SAP customers, Lines of Business (LoB), applications, processes, and services enables us to provide unique insights and expertise in cloud infrastructure, delivery, and end-to-end lifecycle management.

Cloud Infrastructure

+172k
Virtual Machines

+14.3k
Hypervisor

+243 PB/2168 systems
Storage Capacity
(physically available)

436 PB
Quarterly Backup Volume

+118.7 Gb/s
Internet Bandwidth

+10,8k
Devices

+738k
Ports

Our mission is to support SAP strategy by delivering world-class services for cloud infrastructure, life cycle management, and cloud operations to SAP Lines of Business and external customers.

Multi Cloud

118K Virtual Machines Supported

55 Lines of Business onboarded

7.6K Public Accts Supported

9K Line of Business customers on Public Cloud

GCS Team

1900+ Employees
Representing 53 Nationalities
From 24 Countries

GCS Storage

GCS storage service uses traditionally designed, well-hardened storage systems purchased from major vendors.

Storage service has become the most stable service in GCS.

But because storage systems are widely shared, a rare storage incident can still cause large impact.

Hence our interest in minimizing storage incidents.

Our study of storage incidents

Background

When an incident occurs:

- A team is formed composed of relevant GCS and vendor representatives.
- At the end a root cause analysis (RCA) is performed.

The RCA provides guidance for subsequent steps taken to address identified root causes and contributing factors.

Study motivation and methodology

Motivation: Decrease storage incidents without increasing cost

Methodology: Extend RCA analysis to study them as a group

- Looked at RCAs for storage incidents over a 2.5 year period.
- RCAs collected spanned 2000+ storage systems, with total capacity of 200+ petabytes.
- Each RCA analyzed to find common categories of root cause or contributing factors.

Some significant patterns

Role	Incident pattern
Root cause	Single misbehaving drive resulted in incident
Root cause	Other hardware single point of failure resulted in an incident
Root cause	I/O service delivery experienced interference, without automatic recovery
Contributing factor	Happened during upgrade sequence
Contributing factor	Human or process error involved

Some examples of incidents

Single points of failure

1. Single misbehaving drive resulted in panics in all connected controllers.
2. Direct bus connection between controllers caused low-level exception in one controller to create low level exception to the other controller.
3. There was common failure mode that turned off too many fans and caused controllers in a chassis to get overheated.

I/O service interference without automatic recovery

1. Single network port driver stopped sending packets, not detected and therefore not recovered.
2. Controller became partially disabled due to failure in local logging device, but no automatic recovery triggered.
3. Memory leak interfering with I/O in a controller but no automatic recovery triggered.

More examples

Human/process involvement

1. Previously set configuration incompatible with new software release.
2. Bad blocks not handled when they occurred, causing migration issues later.
3. Configuration not completed after motherboard replacement.

Happened during upgrade

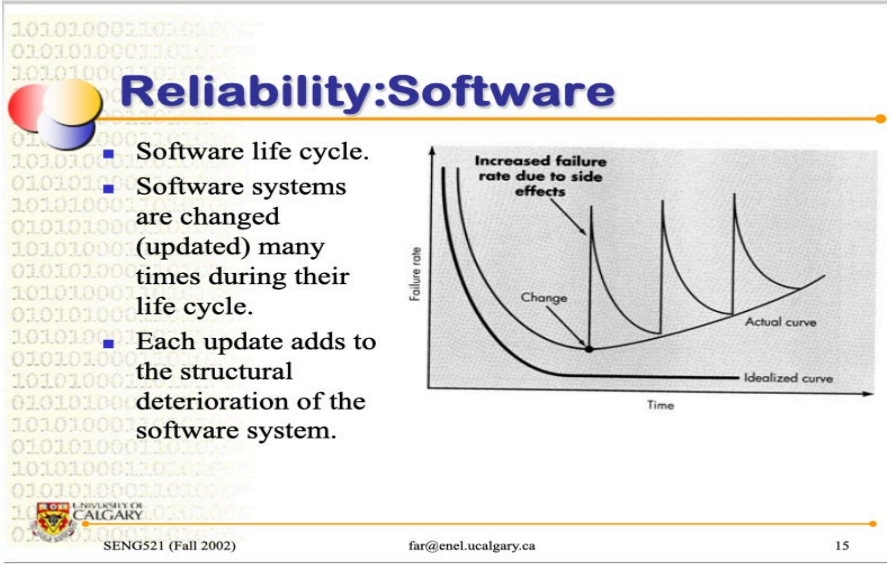
1. Hardware fault discovered during upgrade, not handled by failover/failback and leaving some volumes offline.
2. Existing mount methods not compatible with upgraded firmware.
3. Configuration performed during upgrade was not done correctly.

Looking to the future

Going forward, can we expect fewer incidents?

All past issues have been dealt with (internally or by vendors).

But evolving system functionality will introduce new corner cases.



Reliability:Software

- Software life cycle.
- Software systems are changed (updated) many times during their life cycle.
- Each update adds to the structural deterioration of the software system.

Increased failure rate due to side effects
Failure rate
Change
Actual curve
Idealized curve
Time
UNIVERSITY OF CALGARY

SENG521 (Fall 2002) far@enel.ucalgary.ca 15

Can changes in architecture/design reduce vulnerability?
With little extra cost?

Looking forward

Can the patterns identified indicate some areas to explore?

Here are some thoughts.

Human/process error

Complete automatic pre-maintenance health and compatibility check of the system.

Increased installation and maintenance automation.

Where not automated, more simple, intuitive, tested guidance.

Back-door configuration should be impossible or highly guarded.

AIOps pro-active analytics solutions wherever possible.

Upgrades

We should feel as confident about upgrades as with our iPhone!

An upgrade sequence should always provide recovery from at least a single fault in the active controller(s) –throughout the entire sequence.

Pre-upgrade configuration checking should be complete and zero-touch.

Upgrades should have no dependencies and restrictions:

- Major releases too often have complications that cause overruns in maintenance windows.
- Mandatory <u>disruptive</u> upgrades should absolutely be minimized.

Surveilling the system

Is it possible to introduce agents outside the system to detect anomalous behavior that might not be detected by the system’s internal auto-healing capabilities?

Single points of failure

HA-pairs can be a SPOF.

- Drives or drive shelves connected to all controllers, can be a vulnerability.
- Common chassis or midplane can be a vulnerability.
- Direct connections between nodes can be a vulnerability.
- And, as mentioned, upgrades can remove a controller from service, temporarily removing redundancy and increasing vulnerability of the system.

Is HA-pair architecture the best storage system solution for today’s scale and complexity?

The dominant enterprise storage architecture is still some variant of HA-pair.

HA storage system beginnings

Highly available storage, accessibility assured by two redundant data paths, was first introduced in 1978 –the Memorex 3673 Dual Intelligent String Control.

Then and now

Forty years later, the HA-pair structure remains dominant for delivering high availability to enterprises for their critical business applications.

But scale of deployment and complexity of implementation have skyrocketed.

Metric	System area	1980	2020	Magnitude change
Hardware reliability	Drive MTBF hours	~100,000	~1,000,000	10 ⇧
	Controller MTBF hours	~10,000	~100,000	10 ⇧
System complexity	Drive firmware lines of code	~500	~500,000	100 ⇩
	Controller lines of code	~4,000	~1M	250 ⇩
Scale of deployment	Number of drives per enterprise	~100	~50,000	500 ⇩
	Number of controllers	~20	~2,000	100 ⇩

40 years later

Result: the probability of incidents using HA-pair architecture has increased dramatically.

Factor	Risk influence
Scale of deployment in an enterprise	500 times higher
Complexity (e.g. lines of code, number of gates in hardware) within a storage system	250 times higher
Storage system failure rates	10 times lower

Final thoughts

Data availability will continue to be critical:

- Cloud-native apps that require high transactional performance rely on methods such as separate copies per availability zone.
- And many critical enterprise apps still rely on the primary copy of the data to be available.

To minimize incidents related to primary data copies, there are two choices:

- Shrink impact zone for a storage system incident.
- Enhance availability of the primary storage systems
  - Through design of the system.
  - Through design of how the user interacts with the system
  - Through how the system is monitored

Our path forward: *look carefully at storage systems, focusing on SPOFs, with upgrade sequences that result in least exposure during upgrade, with well-developed AIOps, with zero-touch where possible and otherwise simplest operation.*

Please take a moment
to rate this session.

Your feedback matters to us.