SUSE Linux Enterprise Mission Critical
Simple Can Be Powerful

Neil Barnett
Senior Product Manager
Mission Critical Computing
nbarnett@suse.com

Diego Akechi
Engineering Manager
Mission Critical Computing
dakechi@suse.com
Planned or unplanned, IT downtime is expensive, especially for mission critical applications and services.

Studies show that 31% of downtime is caused by hardware failure, with over half of customers asked in a state of planning to upgrade hardware.
More and more systems considered critical

“What percentage of your applications and data fall into the following tiers?”

Non-critical Applications 29%
Mission Critical 36%
Business Critical 36%

*Source Forrester*
What causes downtime?

• Hardware failure
• Software failure
• Human error
• Climate change??
Top causes of unplanned downtime

“What was the cause(s) of your most significant business disruption?”

- Operational failures
  - Power failure: 43%
  - IT hardware failure: 31%
  - Network failure: 16%
  - IT software failure: 15%

- Natural disasters
  - Hurricane: 12%
  - Flood: 10%
  - Winter storm: 9%
  - Fire: 4%
  - Earthquake: 1%

- Human-caused events
  - Human error: 13%
  - Security: 3%
  - Malicious outsider: 1%
  - Malicious insider: 1%
  - Chemical spill: 1%
  - Terrorism: 1%
  - Other: 5%

Base: 94 global disaster recovery decision-makers and influencers (does not include “don’t know” responses; multiple responses accepted)
Where do we start to solve this problems?

Design with the idea that downtime will happen

Prepare for hardware failures

Assume software is just waiting for the chance to bite you in the…

Remove the opportunity for human error
How are SUSE attacking these problems?
Where do we start to solve this problems?

Design with the idea that downtime will happen

Prepare for hardware failures

Assume software is just waiting for the chance to bite you in the…

Remove the opportunity for human error
What if you could use global resources?
SUSE Linux Enterprise for High Availability Extension

Integrated suite of robust open source clustering technologies that enable enterprises to implement highly available Linux clusters and eliminate single points of failure. Also included is geo clustering, designed to manage cluster servers in data centers anywhere in the world.

What’s New:

- Easily manage clustered Linux servers and monitor the clustered environment
- Prevent mission-critical application downtime by replicating data across multiple clusters
- Maximize mission-critical service availability in mixed clustering environments of both physical and virtual Linux servers
- Protect against regional disasters with geo clustering for service failover over any distance

Geo Clustering is included in SLE HA 15, protecting workloads across globally distributed data centers and providing rules-based failover for automatic and manual transfer of a workload to another cluster outside of the affected area.

- Improved graphical interface (HAWK) saves time and easily manages, configures and monitors clustered Linux servers
- Faster time to value with enhanced and continuous data replication via DRBD (Distributed Replicated Block Device), enabling locking and synchronization across multiple systems on the cluster
- Ability to configure containers as cluster resources for high availability (tech preview)
- Enhanced Pacemaker support for flexible, policy-driven clustering, enabling probes of guest nodes for resource status to prevent any possible concurrency violations
- AutoYaST support for Geo Clustering instead of installing manually, use AutoYaST to clone the HA configuration of existing nodes
SUSE Approach to HA

Easy to use
- Bootstrap clusters, manage with Hawk2, batch mode tests, history reporting

Build to suit your component, application or environment
- Service agents, fencing mechanisms, OCF compliant

Platform agnostic
- x86_64, AArch64, POWER, and IBM Z, virtual, containers or a mix & match

Self-healing and self-repair (policy based autonomy)
- UPS, VM host, storage based, automatic and manual

Protect application, component or data integrity
- Cluster-aware file systems, volume management, data replication

Manage multiple clusters (even across Geo)
- Manage many clusters in a single tool, see all cluster health and reports

Disaster resilience
- Failover in cases of major outage, no distance limitations
Mission Critical Use Cases

Active/Active
- OCFS2, databases, Samba file servers

Active/Passive (with fail-over)
- Traditional databases, SAP setups, regular services, Geo dispersed

High availability for all workloads
- Fine granular monitoring and HA on top of virtualization, management, all regardless of platform

Network load balancing
- Transparent failover

All topologies
- Local, metro, and geographical area clusters

SAP
- SLES for SAP is built on SLE HA
- The reference architecture for SAP with HA
- High Availability for SAP HANA
- Enables HA for SAP in public clouds

Cloud
- OpenStack HA built on SLE HA
- HA for the control plane
- HA for the compute plane using pacemaker_remote
Three-Server Cluster

- Web Server 1
  - Web Site A
  - Web Site B

- Web Server 2
  - Web Site C
  - Web Site D

- Web Server 3
  - Web Site E
  - Web Site F

Switch

Pacemaker Cluster

Shared Storage
Three-Server Cluster after server failure

Web Server 1

Web Server 2

Web Server 3

Web Site A

Web Site C

Web Site D

Web Site B

Web Site E

Web Site F

Switch

Shared Storage

Pacemaker Cluster
Where do we start to solve this problems?

Design with the idea that downtime will happen

Prepare for hardware failures

Remove the opportunity for human error

Assume software is just waiting for the chance to bite you in the…
Can’t we patch software while it runs?

Mankind already flew to the moon…
SUSE Linux Enterprise Live Patching

SUSE Linux Enterprise Live Patching is a technology that improves business continuity and saves costs by reducing downtimes, increasing service availability and enhancing security & compliance.

- Keep your mission critical services running by reducing planned or unplanned downtime by patching systems when you need to (or are forced to).
- Shorten response times when fixing critical security issues. No longer need to wait until a maintenance window (or force one) to fix a problem.
- Maintain your security & compliance position by ensuring systems stay up-to-date with security patches, supported with the capability to audit any applied patches.
Update and reboot or…
Maybe we should just Live Patch....
Key Highlights

- Available for SLE 12 onwards for x86-64, 12 SP2 and SP3 for ppc64le
- Provides fixes for Kernel bugs which affect
  - Security (CVSSv2 >=6)
  - Security (CVSSv3 >=7)
  - Data integrity or system stability
- No runtime performance impact or no interruption of applications
- Deploy using existing package management frameworks
- Patches available for maintained kernels
Virtualization environment running Oracle DB

For virtualization solution, minimizing planned downtime is a top requirement.

SVHW uses SUSE Linux Enterprise Live Patching to maintain continuous uptime for its virtualization layer. SUSE Manager is used to keep patching synchronized across all virtual servers.

“Without Live Patching, we would be faced with bringing down all 40+ virtual machines each time we needed to apply critical patches—at least once a month.”

Saves 2 days per month in administration time
24 days per year!
Live Patching in the user space

- Unlike the kernel, user space has no good boundary
- Code must not be used at the time you are patching
  - “the stack must be empty”
  - ftrace can not guarantee correct address
- Three categories of user space:
  - Transient
    - A program that runs and stops
  - Unknown
    - Solutions will have a penalty overhead as currently counting entrance/exit is required
  - Annotated
    - Amendment to the code/function to include the ability to patch
- Starting with glibc and OpenSSL
Live Patching in the user space

• Demonstrated **libpulp** at SUSELABS CONF 2018.
  
  • [https://www.youtube.com/watch?v=KK776qOajWk](https://www.youtube.com/watch?v=KK776qOajWk)

• Despite the complexity, plan to have all the same value as we do for Live Patching the kernel.
  
  • Same thresholds for patches.
  
  • Non-disruptive patching.
  
  • Use existing management frameworks.

• Looking to work with you on testing and feedback in 2019!
Where do we start to solve this problems?

Design with the idea that downtime will happen

Prepare for hardware failures

Assume software is just waiting for the chance to bite you in the…

Remove the opportunity for human error
ABORT MISSION!

I REPEAT! ABORT MISSION!
System Rollback

Fixing EBKAC and those other pesky gremlins

• Rollback to a good state with one click for faster
• Recovery from planned and unplanned downtime
• Support for service pack rollback
• Support for kernel upgrade
• Based on btrfs and Snapper, bootloader integration
SUSE is leading for RAS capabilities on Linux

Reliability, Availability, Serviceability

- Systems and applications survive hardware failure
- Systems that don’t require constant reboots
- Choice of hardware vendor/architecture
- Integration into systems management frameworks
Aren’t we missing something here?
<table>
<thead>
<tr>
<th>IT Infrastructure</th>
<th>Mode 1</th>
<th>Mode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datacenter</td>
<td>Hosted / Managed</td>
<td>Cloud</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application Deployment</th>
<th>Mode 1</th>
<th>Mode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Servers</td>
<td>Virtual Servers</td>
<td>Containers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application Architecture</th>
<th>Mode 1</th>
<th>Mode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolithic</td>
<td>N-Tier</td>
<td>Microservices</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Development Process</th>
<th>Mode 1</th>
<th>Mode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterfall</td>
<td>Agile</td>
<td>DevOps</td>
</tr>
</tbody>
</table>
Can we go Mode 2 tomorrow….

…when 72% of our applications running today are mission critical?

• These are mission critical applications or services – do not touch!
• Application vendors need to be ready
• Hardware landscape will need to be ready
• Internal processes need to be ready
What do we miss out on....

....because Mission Critical is still very much a Mode 1 play?

• Harder to leverage the public clouds
• Automation takes a hit
• Deployment (Day-1) and management (Day-2) are often two different tasks
• The unicorn that is CI/CD takes a back seat
• We spend our effort reacting rather than being proactive!
10,000ft view of “the modes”

Focus on IT Infrastructure and Application Deployments

Traditional Infrastructure

- Multiple use cases
- Manual and automatic installation
- Variety of updates, upgrades, legacy
- Variable packaging and installation
- Might become huge in size and management

Software-defined Infrastructure

- Single use case, multiple systems
- Automatic and centralized installation
- Always up-to-date
- Fit one purpose
- Small as possible for size and management
Shall we call it Mode 1.5?

We promised simple could be powerful (no one said easy)

- Use private and public infrastructures
- Design new resilient Mode 1 environments
- Automate everything possible
- Remove, where possible, the need for reboots
- Supercharge your monitoring and management
What are SUSE doing in the Mode 1.5 space?
Transform Day-1

Reusable blocks or “lego kits”

• Adapt for your needs
• Integrate on existing deployments
• Deploy complex solutions at once
• Reproduce as many times you want
Enhance Day-1

Use Configuration Management tools

• **Idempotence**: Run many times, get the same final state

• **Infrastructure as Code**: version, change and control your infrastructure as your source code

• **Modernise Processes**: CI/CD and other DevOps possibilities
Your Global Datacenter

- Automate your deployments
- Make Day-1 and Day-2 the same work
- Bridge your zones with SLE HA
- Monitor, monitor, monitor
Enhance Day-2

It’s never been more important to monitor

- Based on modern technologies
- Wide existing applications and languages coverage
- Flexible and easy to use
- Easy to integrate
- Active alerts and reporting
Sneak peak at what is coming….

Come to the HA booth at the Technology Showcase for a live demo!
Take your datacenter with you....

....and with SUSE maintain your control
You can be part of the transformation!

https://github.com/SUSE/ha-sap-terraform-deployments
https://github.com/SUSE/saphanabootstrap-formula
https://github.com/SUSE/habootstrap-formula
https://github.com/SUSE/shaptools
https://github.com/ClusterLabs/hawk-apiserver
https://github.com/ClusterLabs/hawk

Visit our open-source projects and give us your feedback
What’s next from the SUSE Mission Critical teams?
Mission Critical Roadmap

2018 2019 2020 2021

High Availability & GEO Clustering
- Merge Geo clustering into HA
- Rebase stack on SUSE Linux Enterprise 15 common code base
- Support for pacemaker 2
- Extended RAID support for cluster storage**
- Cluster API**

Live Patching
- More architecture support
  (following market needs)

Live Patching
- Support for s390x

Live Patching
- Quorum Node Support
- Softdog as a Watchdog service
- Google Cloud Fencing Support
- Cluster configuration API
- API cluster status support
- Nagios/Prometheus/Grafana Support
- Improve integration into single sign-on
- Cluster Status for SUSE Manager**
  - Role based cluster setup**

Live Patching
- Userland LP
- IBM Z support
- SDK**

** Items are tech preview
* Information is forward looking and subject to change at any time.
Questions ?
Unpublished Work of SUSE LLC. All Rights Reserved.
This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer
This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.