System Recovery Benchmarking

Ji Zhu, James Mauro, Ira Pramanick
Sun Microsystems, Inc.
{ji.zhu, james.mauro, ira.pramanick}@sun.com

Abstract

This paper presents the rationale and usefulness of developing a system recovery benchmark. The speed with which a system can return to service following an outage is a critical factor in overall system availability. General purpose computer systems, such as UNIX based systems, tend to execute the same sequence or series of steps during outage recovery and system startup. Our experience has shown that these steps are repeatable and measurable, and can thus be benchmarked, much like performance benchmarks (e.g. TPC, SPEC). A defined set of measurements, coupled with a specification for representing the results and system variables, provides the foundation for system recovery benchmarking.

1. Introduction

In [7], a hierarchical framework, named R^3, is established to benchmark availability. R^3 represents the three system attributes that are key to availability identified in the framework. These attributes are Rate (rate of fault and maintenance events), Robustness (a system's ability to handle fault, maintenance, and system-external events, and the resulting degree to which it remains available in the face of these events), and Recovery (the speed with which a system can return to operation following an outage). The R3 framework provides for a benchmark that incorporates all defined attributes, thus yielding a downtime-per-year metric. Alternatively, benchmarks can be defined that focus on some subset of the framework attributes and their sub-metrics. This is very similar to the performance benchmarking space, where there are benchmarks that measure the system as a whole, as well as benchmarks that measure a specific subsystem.

In this paper, we discuss our rationale for creating a benchmark specification designed to measure the recovery attribute of general purpose computer systems. Our objective is to convey two basic assertions: recovery time can be benchmarked; and a recovery time benchmark is useful and meaningful. Previous research in availability benchmarking has been focused on benchmarking system robustness attribute [1, 2, 3, 6]. There has been a lack of research in benchmarking recovery aspect of system availability.

2. Background and Motivation

Historically, there have been several design approaches to building computer systems that can meet strict business requirements for availability. Fault tolerant systems implement lock-step execution with results comparison across redundant hardware components, providing the ability to detect and recover from faults without a service disruption. Such designs have carved out a niche market in the industry, but have not seen broad adoption due to cost (overall price/performance) and scalability considerations.

General purpose computer platforms, such as Unix-based servers, offer designs that allow system to recover quickly from an outage. In the context of this paper, quick recovery refers to a computer platform returning to service in an automated fashion.

On a standalone server, quick recovery is facilitated through system firmware and component blacklisting capabilities. When a fault event occurs, the system panics and reboots. During the reboot process, hardware diagnostic software configures around the failed component, allowing the system to return to service quickly and without human intervention. Clustered systems provide redundancy by clustering two or more systems together with a software framework that does cluster management, failure detection and automated recovery. When a cluster node fails, the cluster software initiates a failover of the services that were being provided by the failed node, to other nodes in the cluster, thereby minimizing service disruption.

There are several major benefits of choosing quick recovery over fault tolerance. General purpose computers provide better price/performance, better scalability, and a much larger selection of commercially available software. From an availability perspective, general purpose computers have been proven adequate even in environments demanding very high levels of availability. General purpose computers have been installed across a broad range of industry segments to run mission critical applications for years.

3. Can We Benchmark Quick Recovery Time?

General purpose computer systems tend to execute the same sequence or series of steps during outage recovery and system startup. On a standalone server, quick recovery is mostly in the form of a kernel panic call and a core dump, followed by a system reboot, which usually includes firmware-based hardware diagnostics prior to the operating system boot. On clustered systems, quick recovery is in the form of a reconfiguration of cluster framework and a restart of services on the surviving nodes. Our experience has shown that recovery times are repeatable and measurable, and can thus be benchmarked, much like performance benchmarks (e.g. SPEC[4], TPC[5]).

4. Is A Quick Recovery Time Benchmark Meaningful?

For general purpose computers, outages that can quickly be recovered from, account for most of the system outages. On mid-range and high-end Unix servers, systems can quickly recover from faults of most its components (processors, memory, cache, interconnect, controller, etc.). Together these components account for up to 80% of total system hardware failure rate. Software faults can also be worked around by rebooting the system. For clustered systems, failover is the dominant mode of recovering from a hardware or a software fault.

Most outages on mid-range and high-end general purpose computer servers fall into the quick recovery category. This makes a system quick recovery benchmark useful for evaluating a broad range of systems, as it provides a meaningful representation of system outage duration.

5. Issues

A full discussion of the issues and their solutions in a quick recovery benchmark is outside the scope of this position paper. We are working on them and will report our progress in the future. The following is a list of a few important issues we have identified in the quick recovery benchmark for a standalone server.

- System Size - The number of processors, amount of physical memory, number of IO channels and the number of actual storage devices (disks) directly impact system recovery time.
- Service Processor - Systems with service processors add a level of complexity to a full restart cycle if the service processor must be restarted as well.
- Domains - Single physical systems able to run multiple operating system kernel instances have multiple recovery scenarios; those effecting the domain, and those effecting the entire system.
- Firmware Setting - Most systems offer parameters that determine the level of diagnostic testing the system will do at startup/restart. Going from least intensive to most intensive settings can dramatically alter restart time.
- File Systems - File system integrity checking is a critical phase of system startup/restart. The number and size of file systems, as well as options such as logging, can have a significant impact of startup/restart time.

We are working on defining outstanding issues for clustered systems.

6. Conclusions

It is our contention that quick recovery time measurement is a meaningful availability benchmark since it represents one important aspect of system availability - the outage duration when a recoverable fault occurs. We believe that it is an attainable goal to develop a benchmark on quick recovery time much like performance benchmarks. We are currently working on system quick recovery benchmark specifications for a standalone server and a clustered system.

7. References

[1] P. Koopman et al, "Comparing Operating Systems Using Robust Benchmarks", *Proceedings of the 16th Symposium on Reliable Distributed Systems*, pp. 72-79, Oct. 1997.

[2] H. Madeira and P. Koopman, "Dependability Benchmarking: making choices in an n-dimensional problem space." *Proceedings of the first workshop on Evaluating and Architecting System Dependability*, July, 2001.

[3] D. P. Siewiorek et al, "Development of a Benchmark to Measure System Robustness", *Proceedings of the 1993 International Symposium on Fault-Tolerant Computing*, pp. 88-97, June 1993.

[4] Standard Performance Evaluation Corporation, http: //www.spec.org/, *Current*.

[5] Transaction Processing Council, *http://www.tpc.org/*, Current.

[6]. T. Tsai et al, "An Approach Towards Benchmarking of Fault Tolerant Commercial Systems", *Proceedings of the 1996 Symposium on Fault-Tolerant Computing*, pp. 314-323, June 1996.

[7] J. Zhu et al, "R-Cubed: Rate, Robustness and Recovery - An Availability Benchmark Framework". To appear in *Technical Report Series*, Sun Microsystems Inc.