IBM Director Software Rejuvenation

A software tool for On Forever computing

Executive Summary

As companies continue to rapidly deploy Intel® processor-based servers in mission-critical areas of their businesses where continuous operation is expected, managing and controlling the costs associated with these servers will increase in importance. IT professionals across all segments of the market will expect servers to be enabled for manageability in any environment, right out of the box, and will depend on manageability tools that easily allow them to deploy, manage and support the entire server configuration over its entire life cycle. Rapid deployment, high availability, proactive management and the ability to remotely solve problems quickly and efficiently will be of the utmost importance in order to minimize costs associated with lost productivity and server downtime. GartnerGroup estimates that the annual total cost of ownership (TCO) of Intel processor-based servers is roughly four times that of desktops, or between $40,000 and $50,000 per year. Depending on the application, the total cost of server downtime is severe—it can be as high as $27,000 per minute for critical applications.¹

For Intel processor-based servers, experience has shown that downtime caused by software failures far exceeds downtime associated with hardware failures. In fact, system crashes are two times or more likely to be the result of software failures than hardware failures². Software often exhibits an increasing failure rate over time typically because programming errors generate increasing and unbounded resource consumption. Other causes can include data corruption and numerical error accumulation, such as round-off errors. These errors can cause memory leaks, file systems that fill up over time, and spawned threads or processes that are never terminated. These effects constitute a phenomenon known as software aging, and may be caused by errors in application, middleware or operating system software. Furthermore, as servers expand to include multiple processors and become grouped in multiple nodes, the level of software complexity grows significantly, which compounds this effect. The greater the complexity of software, the more likely it is that it will suffer from the phenomenon of software aging. The problems that can lead to software aging over time resulting in software failure and server downtime include the following:

- Memory leaks—memory leaks from software can cause virtual memory, non-paged pool and paging space to increase, and over time this degrades performance to the point at which the system is unresponsive.
- Unreleased file locks—the number of file locks can increase, putting a tremendous burden on the file system.

• Data corruption—internal memory or file system data or both becomes corrupted, resulting in application failures and loss of data.
• Non-terminated threads—the number of threads increases over time, causing memory and performance problems.

This paper gives a brief overview of IBM Software Rejuvenation and how it can help greatly improve system reliability and provide benefit to customers.

**What Is Software Rejuvenation?**

To date, industry initiatives addressing the problems of software aging have focused on reactive solutions, such as fault-tolerant systems. Software rejuvenation is a proactive technology designed to predict and manage aging. Note that software rejuvenation does not replace other failure recovery techniques but complements them by helping reduce the possibility that these types of failures will occur.

Software rejuvenation is the concept of gracefully terminating an application and immediately restarting it with a refreshed internal state. Software rejuvenation solves software aging by minimizing failures through periodic, preemptive rollback of applications and server reboot. The most straightforward way is to manually reboot the server periodically, and users often do this. Manual reboot minimizes the above-mentioned problems or prevents them from occurring by not allowing the allocation of internal resources to reach critical levels. However, the problem with this approach is that the server is down during the reboot process. By coupling two technologies—software rejuvenation and clustering—system availability can be dramatically improved.

![Figure 1. Software rejuvenation reduces downtime.](image)

There are two approaches to software rejuvenation: time-based software rejuvenation (TSR) and symptom-based selective software rejuvenation (SSR). Both approaches provide a trigger that can be used to initiate a graceful and rejuvenating reallocation of workload within a clustered environment.

TSR relies on the skill of an experienced operator to assess the approximate frequency of unplanned outages due to resource exhaustion. Using tools such as IBM Director, an operator can monitor server subsystems and software processes to ascertain common trends accompanying regular failures. This knowledge can then be applied to automate rejuvenation of
the affected system on a regular and planned basis that minimizes the impact of system downtime to the business.

SSR provides a more sophisticated method of estimating when a system or application is in need of rejuvenation. A rejuvenation agent monitors the behavior of a software system for evidence of resource exhaustion. Most operating systems provide ways to monitor system behavior: for example, Microsoft® Windows NT® exports information to the registry, allowing an agent to monitor a range of resources such as semaphores, events and virtual memory.

One or more such critical parameters can be monitored for signs of exhaustion, which can precede a software failure. When a threshold or high-risk area is reached, an alert can be generated, acting as a trigger for system recovery before the application or system suffers an unplanned failure. It is even possible to associate the detected resource exhaustion with a specific application, process or thread once a problem has been identified.

This approach can be taken a step further by using statistical trending techniques to extrapolate data gathered by the rejuvenation agent and provide a prediction of application or system failure. Because there are many complex algorithms involved in this kind of process and there is a strong association among the operating system, application and workload, it is sensible that the system should be self-educating, choosing the most appropriate analytical procedures on its own experience of system outages. Predictions of this kind give system administrators adaptive software Predictive Failure Analysis® (PFA), allowing time to schedule rejuvenation in anticipation of a pending software failure just as they have been able to do with hardware PFA on critical components such as disk drives, fans and power supplies, memory and processors.

IBM Software Rejuvenation

IBM has been pioneering software rejuvenation technology in conjunction with Duke University to improve reliability in both server and telecommunications environments.3 With IBM’s innovative Software Rejuvenation, you can automatically reset select xSeries servers gracefully before failure, avoiding costly downtime. TSR for IBM @server xSeries servers lets you schedule software rejuvenation for times that are least disruptive for your system users. The rejuvenation can be based on the time elapsed since the last rejuvenation, or it can be based on having completed a particular workload, such as a selected number of batch jobs. Because TSR requires no modifications to the application software, it can be used for any application running on the server. As an added benefit, TSR provides an automatic and continuous check on your system’s capability to tolerate an unplanned failure. When software rejuvenation is invoked within a clustered environment, cluster management failover services can be used to gracefully stop the offending subsystem and restart it on the same or another node in the cluster, in a controlled manner.

Today IBM delivers a software rejuvenation tool with IBM Director. IBM Director is a comprehensive workgroup hardware manager designed for IBM and non-IBM server, desktop, workstation and notebook systems. It is built upon industry standards, designed for ease of use, and allows for smooth integration into leading workgroup and enterprise systems management environments.

As part of the IBM Director console, Software Rejuvenation is a task that can be invoked on an individual system or an MSCS cluster. This task provides a GUI to configure the Software Rejuvenation settings. The first release of Software Rejuvenation simply used time as the determining factor of when the servers should be rebooted (TSR). This is done without restarting or monitoring individual applications. The default state of Software Rejuvenation is disabled, requiring the customer to indicate if and when the servers will be rebooted. The administrator can specify the rejuvenation interval and the time of day to rejuvenate (default is 12:00 A.M.). IBM software rejuvenation can operate in an MSCS environment, eliminating downtime during the

3 This project is funded through the Center for Advanced Computing and Communication (CACC) in cooperation with Duke University.
recycling of the server. Software Rejuvenation will reboot the failing node and MSCS detects that failure and restarts the resources on another node.

**IBM Software Rejuvenation: New Enhancements**

The latest release of Software Rejuvenation adds predictive analysis (SSR) based on advanced research within IBM. The tool is built on a model that determines the optimal interval for periodic, preemptive rollback of continuously running applications in order to prevent software failures. An algorithm is employed which takes into account not only time, but application and system behavior as well. Predictive alerts are sent to notify the system administrator of a pending failure. The introduction of the software PFA capability will complement the extensive hardware PFA available on current xSeries servers.

The latest release of Software Rejuvenation also has the ability to isolate faulty software within the system. It can be used in conjunction with IBM Director’s Event Action Plans to pinpoint and reset individual software components, such as a specific application, instead of the entire system. The benefits of doing this are:

- Continued operation of other, healthy applications on the same server while the failing application is being reset
- Faster turnaround time on resetting the server

A potential future enhancement includes factoring performance characteristics into the model to allow software rejuvenation to help ensure that systems operate at their optimal levels. Because Software Rejuvenation is an integrated task within IBM Director, it can be used in conjunction with the base management features of IBM Director. For example, an administrator
can set up an event action plan on how and when to be notified of a pending failure predicted by the Software Rejuvenation tool. This is illustrated in the following screen capture.

![Figure 3. Predictive failure notification](image)

**Customer Benefits**

The most obvious benefit of IBM Software Rejuvenation is that, by helping to reduce software failure rates, system availability is improved. The downtime due to an unplanned software failure is far greater than any time lost due to graceful, scheduled rejuvenation. As a system's resources gradually approach a critical level, the server becomes more likely to suffer downtime. To help avoid downtime, you can schedule Software Rejuvenation to reset the software system with no operator intervention. The reset can be just the application that is responsible for the aging, further reducing any impact on system operation, or it could include the entire server system. As your software ages, the benefits of Software Rejuvenation increase. When Software Rejuvenation reinitiates a server, the server's failure rate falls back to its initial, lower level because resources have been freed and the effects of software aging have been removed (see Figure 1). This has a dramatic effect on overall system availability of your xSeries server. Furthermore, you can enable a clustered xSeries server to gracefully fail over to another server in the cluster, so that users need never know that you are resetting a server.

Software Rejuvenation also reduces the labor cost associated with administering servers by improving diagnosis time. An administrator is proactively warned of an impending system software failure and is able to accurately predict problems associated with aging software. Software rejuvenation also results in fewer calls to the support center, which can also lower costs.

**Conclusion**

As part of its X-architecture initiative, IBM continues to enhance its xSeries line of industry-standard servers with technologies and capabilities for business-critical computing and continuous availability. IBM Software Rejuvenation is a tool to help increase server availability by proactively addressing software and operating system aging.
Software Rejuvenation is one of the integrated tools provided with IBM’s workgroup hardware manager, IBM Director, and allows for the scheduling of graceful failovers before system failures. New enhancements of this tool provide predictive alerting based on the characteristics of the individual software components running on the server and factors in performance characteristics to optimize server operation.

With Software Rejuvenation you can fix possible problems before they occur, further increasing the availability and reliability of your xSeries server systems. And as your software ages and software errors become more likely, Software Rejuvenation can continue to help you reduce server downtime, making xSeries servers capable of handling the most business-critical applications.

**Additional Information**

For more information on IBM @server xSeries direction, products and services, visit our Web site at [ibm.com/us/eserver/xseries](http://ibm.com/us/eserver/xseries). From the xSeries home page, select Library and you will see a list of the different types of documentation available.
Prevent software-related downtime with automated refreshing