



## ➤ A Software Rejuvenation Framework for Distributed Computing

**This framework supports graceful degradation of services at best possible performance levels.**

*NASA's Jet Propulsion Laboratory, Pasadena, California*

A performability-oriented conceptual framework for software rejuvenation has been constructed as a means of increasing levels of reliability and performance in distributed stateful computing. As used here, "performability-oriented" signifies that the construction of the framework is guided by the concept of analyzing the ability of a given computing system to deliver services with gracefully degradable performance. The framework is especially intended to support applications that involve stateful replicas of server computers.

Software rejuvenation has been recognized as a simple yet effective means of preventing accumulation of software errors that, if allowed to accumulate, could degrade the capacity or cause failure of a computer system. When a software system is voluntarily rebooted, with high probability, errors accumulated during previous execution are eliminated and the system regains its full capacity. Although software rejuvenation has been investigated extensively, it has not, until now, been considered for stateful applications that involve server replicas. The problem of software rejuvenation in such applications is complicated by the

following considerations: When software rejuvenation temporarily stops a long-running replica server, R, the post-rejuvenation performance of R may be reduced because the stoppage may cause the state of R to become inconsistent with the nominal state of other replicas. In that case, R would be unable to provide services at its full capacity until consistency with the states of the other replicas was restored.

The present performability-oriented framework is based on three building blocks: a rejuvenation algorithm, a set of performability metrics, and a performability model. The performability metrics and model both take account of the reduced nature of post-rejuvenation performance pending restoration of consistency. The performability model also takes account of the possibility that post-rejuvenation consistency-restoration processes could be vulnerable to failures because of the potential performance stress caused by service requests accumulated during rejuvenation.

The basic version of the rejuvenation algorithm uses pattern-matching mechanisms to detect pre-failure conditions. To compensate for the inability

of pattern-matching mechanisms to detect pre-failure-condition patterns other than those known *a priori*, an enhanced version of the algorithm accommodates a random timer and provides for synergistic coordination of both detection-triggered and timer-triggered rejuvenation. It has been demonstrated, via model-based evaluation, that this performability-oriented framework enables error-accumulation-prone distributed applications to continuously deliver gracefully degradable services at the best possible performance levels, even in environments in which the affected systems are highly vulnerable to failures. It has also been shown that software rejuvenation can be realized as an integral part of the infrastructures in stateful distributed computing applications that guarantee eventual consistency of the states of server replicas.

*This work was done by Savio Chau of Caltech for NASA's Jet Propulsion Laboratory.*

*The software used in this innovation is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-42352.*

## ➤ Kurtosis Approach to Solution of a Nonlinear ICA Problem

**A gradient-descent algorithm minimizes the kurtosis of an output vector.**

*NASA's Jet Propulsion Laboratory, Pasadena, California*

An algorithm for solving a particular nonlinear independent-component-analysis (ICA) problem, that differs from prior algorithms for solving the same problem, has been devised. The problem in question — of a type known in the art as a post nonlinear mixing problem — is a useful approximation of the problem posed by the mixing and subsequent nonlinear distortion of sensory signals that occur in diverse scientific and engineering instrumentation systems.

Prerequisite for describing this particular post nonlinear ICA problem is a de-

scription of the post nonlinear mixing and unmixing models depicted schematically in the figure. The mixing model consists of a linear mixing part followed by a memoryless invertible nonlinear transfer part. The unmixing model consists of a nonlinear inverse transfer part followed by a linear unmixing part. The source signals are recovered if each operation in the unmixing sequence is the inverse of the corresponding operation in the mixing sequence.

More specifically, in the models,

$$\mathbf{s}(n) = [s_1(n), s_2(n), \dots, s_N(n)]^T$$

is an  $N \times 1$  column vector representing  $N$  independent source signals at time  $n$  that one seeks to estimate. This vector is multiplied by  $\mathbf{A}$ , an initially unknown  $N \times N$  matrix that represents the linear mixing of the source signals. The  $N$  signals resulting from the mixing are represented by  $N \times 1$  column vector

$$\mathbf{v}(n) = [v_1(n), v_2(n), \dots, v_N(n)]^T.$$

Each of these signals is then subjected to nonlinear distortion represented by a function that is initially unknown and could differ from the functions that represent the distortions of the other sig-