SRC Technical Note
1998 - 006

March 10, 1998

Abstracts of SRC Technical Notes

Compiled by James Mason

Systems Research Center
130 Lytton Avenue
Palo Alto, California 94301

http://www.research.digital.com/SRC/

Abstract

This document presents abstract and title listings of all SRC Technical Notes published to date. The technical notes series provides a fast track for the publication of work in progress, position papers, and interim results. The series is accessible at:

`www.research.digital.com/SRC/publications/`

Also at this web site is the SRC Research Report series and a feature articles section. The research report series consists of formal publications that are peer reviewed while the feature articles provide overviews of research projects for general audiences.

Information on how to retrieve technical notes and subscribe to publication announcements is given in section two of this paper.

Contents

1 Abstracts of SRC Technical Notes

- SRC Technical Note 1994-001

  Introduction to TLA
  Leslie Lamport
  December 16, 1994. 7 pages.

  A short introduction to what TLA formulas mean. It should allow you to understand TLA specifications.

- SRC Technical Note 1996-001

  Refinement in State-Based Formalisms
  Leslie Lamport
  December 18, 1996. 7 pages.

  A note explaining what refinement and dummy variables are all about. It also sneaks in an introduction to TLA.

- SRC Technical Note 1996-002

  The Module Structure of TLA+
  Leslie Lamport
  September 12, 1996. Corrected January 5, 1998. 15 pages.

  An HTML document that informally describes the revised syntax and semantics for the module structure of TLA+. This is a preliminary draft; comments are welcome.

- SRC Technical Note 1997-001

  Virginity: A Contribution to the Specification of Object-oriented Software
  K. Rustan M. Leino and Raymie Stata
  April 30, 1997. 11 pages.

  In object-oriented programs built in layers, an object at a higher level of abstraction is implemented by objects at lower levels of abstraction. It is usually crucial to correctness that a lower-level object not be shared among several higher-level objects. This paper unveils some difficulties in writing procedure specifications strong enough to guarantee that a lower-level object can be used in the implementation of another object at a higher level of abstraction. To overcome these difficulties, the paper presents "virginity", a convenient way of specifying that an object is not globally reachable and thus can safely be used in the implementation of a higher-level abstraction.

- SRC Technical Note 1997-002

  Average-Case Analysis of First Fit and Random Fit Bin Packing
  Susanne Albers and Michael Mitzenmacher
  February 21, 1997. 16 pages.

  We prove that the First Fit bin packing algorithm is stable under the input distribution $U\{k-2, k\}$ when k is at least three, settling an open question from the recent survey by Coffman, Garey, and Johnson. Our proof generalizes the multi-dimensional Markov chain analysis used by Kenyon, Rabani, and Sinclair to prove that Best Fit is stable under this distribution. Our proof is motivated by an analysis of Random Fit, a new simple packing algorithm related to First Fit, which we also show is stable under this distribution.

- SRC Technical Note 1997-003

  Hypermedia Presentation and Authoring System
  Jin Yu and Yuanyuan Xiang
  May 6, 1997. 15 pages.

  This paper describes the Hypermedia Presentation and Authoring System (HPAS), a Web browser with built-in authoring capabilities. In contrast to traditional browsers, it manages time-based hypermedia presentations and the associated dynamic spatial layout with respect to time. The system manipulates various media types by embedding object presenters, editors, and converters, which combine to provide transparent operations on hypermedia objects, such as MPEG videos, GIF images, rich text files, etc.

- SRC Technical Note 1997-004a

  Fully Dynamic 2-Edge Connectivity Algorithm in Polylogarithmic Time per Operation
  Monika Rauch Henzinger and Valerie King
  June 27, 1997. 18 pages.

  This paper presents the first dynamic algorithm that maintains 2-edge connectivity in polylogarithmic time per operation. The algorithm is a Las-Vegas type randomized algorithm.

  The expected time for $p = \Omega(m + n)$ insertions or deletions of edges is $O(p \log^5 n)$, where m is the number of edges in the initial graph with n nodes. The worst-case time for a query is $O(\log n)$. If only deletions are allowed then the cost for p updates is $O(p \log^4 n)$ expected time.

- SRC Technical Note 1997-005a

  The Vesta-2 Software Description Language
  Allan Heydon, Jim Horning, Roy Levin, Timothy Mann, Yuan Yu
  June 24, 1997. Revised January 9, 1998.

  Vesta-2 is a software configuration management system. Developers use Vesta-2 to build and manage potentially large scale software. The instructions for building a software artifact are written in the Vesta-2 software description language (SDL). Evaluating a Vesta-2 SDL program causes the artifact to be constructed.

  This note describes the syntax and semantics of the Vesta-2 SDL. It is a reference manual for use by Vesta-2 users. The language is functional, uses lexical scoping, and is dynamically typed. It includes a single primitive function for invoking external tools like compilers and linkers as function calls.

- SRC Technical Note 1997-006a

  The Operators of TLA+
  Leslie Lamport
  April 12, 1997. Corrected June 8, 1997. 20 pages.

  This document is an introduction to the syntax and semantics of the operators of TLA+. It assumes that you are familiar with ordinary mathematics (sets and functions) and are at least acquainted with TLA. It should enable you to understand the expressions that appear in TLA+ specifications.

  This is a preliminary document; suggestions are welcome.

- SRC Technical Note 1997-007

  Checking Object Invariants
  K. Rustan M. Leino and Raymie Stata
  January 2, 1997. 17 pages.

  When writing computer programs, programmers make assumptions about the relations among variables. In object-oriented programs, these assumptions include relations among the instance variables of a single object, relations often referred to as object invariants. It is a good idea to explicitly annotate a program with these assumptions. Then, a static program-analysis tool can inspect the annotated program to check that routines preserve object invariants. This paper considers two issues that affect what object invariants a program analysis tool can check: object construction and modular checking. The paper suggests some programming idioms and program annotations that widen the range of object invariants that a static program checker can check. The paper also suggests a simple extension to the Java programming language that makes the language more amenable to object-invariant checking.

- SRC Technical Note 1997-008

  A Small Dual-automorphic Lattice with no Involutory Dual Automorphism
  K. Rustan M. Leino and Lyle Ramshaw
  May 27, 1997. 4 pages.

  An exercise in Garrett Birkhoff's renowned book on lattice theory asks for a lattice with 18 elements and of length 5 that has a dual automorphism, but no involutory dual automorphism. This note constructs a smaller lattice, 15 elements and length 4, with the same property.

- SRC Technical Note 1997-009

  Juno-2 Language Definition
  Greg Nelson and Allan Heydon
  June 30, 1997. 15 pages.

  Juno-2 is a constraint-based language intended for graphics applications. A Juno-2 program describes a picture; a Juno-2 implementation renders the picture. The paper describes the syntax and semantics of the Juno-2 language.

  The Juno-2 language is useful for drawing pictures, and also interesting for its simplicity, uniformity, and its provisions for solving constraints. We hope the paper will be useful to Juno-2 users, and also of interest to programming language users and designers.

- SRC Technical Note 1997-010

  Focus+Context Displays of Web Pages: Implementation Alternatives
  Marc H. Brown, Hannes Marais, Marc A. Najork, William E. Weihl
  May, 1997.

  This paper describes an outline processor display of Web pages. We attach icons, called zippers, to the HTML heading tags (H1, H2, ...), and the user can dynamically include or elide the body of each section by clicking on the zipper. We have implemented zippers in three different ways: zippers that are inserted by a custom-built Web browser and that control the browser's display engine; zippers that are inserted into the HTML document by a Web proxy and that use the proxy to generate modified HTML reflecting the changed state of the zipper; and zippers that trigger a JavaScript program which redisplays an appropriately modified version of the page.

- SRC Technical Note 1997-011

  Tight Thresholds for the Pure Literal Rule
  Michael Mitzenmacher
  June 17, 1997. 6 pages.

  We consider the threshold for the solvability of random k-SAT formulas using the pure literal rule. We demonstrate how this threshold can be found by using differential equations to determine the appropriate limiting behavior of the pure literal rule.

- SRC Technical Note 1997-012

  Online Throughput-Competitive Algorithm for Multicast Routing and Admission Control
  Ashish Goel, Monika Rauch Henzinger and Serge Plotkin
  June 24, 1997. 21 pages.

  We present the first polylog-competitive online algorithm for the general multicast problem in the throughput model. The ratio of the number of requests accepted by the optimum offline algorithm to the expected number of requests accepted by our algorithm is $O(\log M(\log n + \log\log M)\log n)$, where M is the number of multicast groups and n is the number of nodes in the graph. We show that this is close to optimum by presenting an $\Omega(\log n \log M)$ lower bound on this ratio for any randomized online algorithm against an oblivious adversary, when M is much larger than the link capacities. We also show that it is impossible to be competitive against an adaptive online adversary.

  As in the previous online routing algorithms, our algorithm uses edge-costs when deciding on which is the best path to use. In contrast to the previous competitive algorithms in the throughput model, our cost is not a direct function of the edge load. The new cost definition allows us to decouple the effects of routing and admission decisions of different multicast groups.

- SRC Technical Note 1997-013

  Mobile Ambient Synchronization
  Luca Cardelli
  July 10, 1997.

This note describes a non-distributed implementation of the basic operations of the Ambient Calculus, which is a calculus for mobile computing developed together with Andrew Gordon. The implementation uses standard shared-memory concurrent programming technology in the form provided by Java. The presentation is self-contained, but previous familiarity with the Ambient Calculus is useful for motivation and intuitions.

- SRC Technical Note 1997-014

  Exploring Unknown Environments
  Susanne Albers, Monika Rauch Henzinger
  July 1997. 30 pages.

  We consider exploration problems where a robot has to construct a complete map of an unknown environment. We assume that the environment is modeled by a directed, strongly connected graph. The robot's task is to visit all nodes and edges of the graph using the minimum number R of edge traversals. Koutsoupias gave a lower bound for R of $\Omega(d^2m)$, and Deng and Papadimitriou showed an upper bound of $d^{O(d)}m$, where m is the number of edges in the graph and d is the minimum number of edges that have to be added to make the graph Eulerian. We give the first sub-exponential algorithm for this exploration problem, which achieves an upper bound of $d^{O(\log d)}m$. We also show a matching lower bound of $d^{\Omega(\log d)}m$ for our algorithm. Additionally, we give lower bounds of $2^{\Omega(d)}m$, resp. $d^{\Omega(\log d)}m$ for various other natural exploration algorithms.

- SRC Technical Note 1997-015

  Syntactic Clustering of the Web
  Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, Geoffrey Zweig
  July 25, 1997.

  We have developed an efficient way to determine the syntactic similarity of files and have applied it to every document on the World Wide Web. Using this mechanism, we built a clustering of all the documents that are syntactically similar. Possible applications include a "Lost and Found" service, filtering the results of Web searches, updating widely distributed web-pages, and identifying violations of intellectual property rights.

- SRC Technical Note 1997-016a

  Continuous Profiling: Where Have All the Cycles Gone?
  Jennifer M. Anderson, Lance M. Berc, Jeffrey Dean, Sanjay Ghemawat, Monika R. Henzinger, Shun-Tak A. Leung, Richard L. Sites, Mark T. Vandevoorde, Carl A. Waldspurger, and William E. Weihl
  July 28, 1997. Modified September 3, 1997.

  This paper describes the DIGITAL Continuous Profiling Infrastructure, a sampling-based profiling system designed to run continuously on production systems. The system supports multiprocessors, works on unmodified executables, and collects profiles for entire systems, including user programs, shared libraries, and the operating system kernel. Samples are collected at a high rate (over 5200 samples/sec per 333-MHz processor), yet with low overhead (1-3slowdown for most workloads).

  Analysis tools supplied with the profiling system use the sample data to produce a precise and accurate accounting, down to the level of pipeline stalls incurred by individual instructions, of where time is being spent. When instructions incur stalls, the tools identify possible reasons, such as cache misses, branch mispredictions, and functional unit contention. The fine-grained instruction-level analysis guides users and automated optimizers to the causes of performance problems and provides important insights for fixing them.

- SRC Technical Note 1997-017

  Dynamic Coscheduling on Workstation Clusters
  Patrick G. Sobalvarro, Scott Pakin, William E. Weihl, Andrew A. Chien
  March 14, 1997.

  Coscheduling has been shown to be a critical factor in achieving efficient parallel execution in timeshared environments. However, the most common approach, gang scheduling, has limitations in scaling, can compromise good interactive response, and requires that communicating processes be identified in advance.

  We explore a technique called dynamic coscheduling (DCS) which produces emergent coscheduling of the processes constituting a parallel job. Experiments are performed in a workstation environment with high performance networks and autonomous timesharing schedulers for each CPU. The results demonstrate that DCS can achieve effective, robust coscheduling for a range of workloads and background loads. Empirical comparisons to implicit scheduling and uncoordinated scheduling are presented. Under spin-block synchronization, DCS reduces job response times by up to 20% over implicit scheduling while maintaining fairness; and under spinning synchronization, DCS reduces job response times by up to two decimal orders of magnitude over uncoordinated scheduling. The results suggest that DCS is a promising avenue for achieving coordinated parallel scheduling in an environment that coexists with autonomous node schedulers.

- SRC Technical Note 1997-018

  The 1995 SQL Reunion: People, Projects, and Politics
  Edited by Paul McJones
  August 20, 1997.

  A reunion of people who worked on System R and its derivatives, including SQL/DS, DB2, and R*, was held at Asilomar on May 29, 1995. This is an edited transcript of the day's discussions, incorporating changes provided by the speakers. It provides an informal but first-hand account of the birth of SQL, the history of System R, and the origins of a number of other relational systems inside and outside IBM.

- SRC Technical Note 1997-019

  Maintaining Minimum Spanning Trees in Dynamic Graphs
  Monika Rauch Henzinger and Valerie King
  September 5, 1997.

  We present the first fully dynamic algorithm for maintaining a minimum spanning tree in time $o(\sqrt{n})$ per operation. To be precise, the algorithm uses $O(n^{1/3} \log n)$ amortized time per update operation. The algorithm is fairly simple and deterministic. An immediate consequence is the first fully dynamic deterministic algorithm for maintaining connectivity and bipartiteness in amortized time $O(n^{1/3} \log n)$ per update, with $O(1)$ worst case time per query.

- SRC Technical Note 1997-020

  Improved Data Structures for Fully Dynamic Biconnectivity
  Monika Rauch Henzinger
  September 8, 1997.

  We present fully dynamic algorithms for maintaining the biconnected components in general and plane graphs.

  A fully dynamic algorithm maintains a graph during a sequence of insertions and deletions of edges or isolated vertices. Let m be the number of edges and n be the number of vertices in a graph. The time per operation of the previously best deterministic algorithms were $O(\min(m^{2/3}, n))$ in general graphs and $O(\sqrt{n})$ in plane graphs for fully dynamic biconnectivity. We improve these running times to $O(\sqrt{m}\log n)$ in general graphs and $O(\log^2 n)$ in plane graphs. Our algorithm for general graphs can also find the biconnected components of all vertices in time $O(n)$.

- SRC Technical Note 1997-021

  Certificates and Fast Algorithms for Biconnectivity in Fully-Dynamic Graphs
  Monika Rauch Henzinger and Hans La Poutre
  September 30, 1997.

  In this paper, we present sparse certificates for biconnectivity together with algorithms for updating these certificates. We thus obtain fully-dynamic algorithms for biconnectivity in graphs that run in $O(\sqrt{n}\log n\log(m/n))$ amortized time per operation, where m is the number of edges and n is the number of nodes in the graph. This improves upon the results in the paper "Improved Data Structures for Fully Dynamic Biconnectivity" (M. H. Rauch, Proc. 26th Annual Symposium on Theory of Computing) in which algorithms were presented running in $O(\sqrt{m}\log n)$ amortized time, and solves the open problem to find certificates to speed up biconnectivity, as stated in "Sparsification - A technique for speeding up dynamic graph algorithms," (D. Eppstein et al. Proc. 33nd Annual Symp. on Foundations of Computer Science, 1992).

- SRC Technical Note 1997-022

  Short-Length Menger Theorems
  Monika Rauch Henzinger, Jon Kleinberg, Satish Rao
  November 24, 1997.

  We give short and simple proofs of the following two theorems by Galil and Yu. Let s and t be two vertices in an n-node graph G. (1) There exist k edge-disjoint $s - t$ paths of total length $O(nk^{1/2})$. (2) If we additionally assume that the minimum degree of G is at least k, then there exist k edge-disjoint $s - t$ paths, each of length $O(n/k)$.

- SRC Technical Note 1997-023

  Each to Each Programmer's Reference Manual
  Paul McJones and John DeTreville
  October 1, 1997.

  Each to Each applies collaborative filtering techniques to the problem of making subjective recommendations to consumers faced with "infoglut". The basic idea is to ask people to vote for items on a numeric scale, then perform a statistical analysis of the collection of all people's votes, and use the results of the analysis to predict additional items of potential interest to a particular person. Unlike some competitive approaches, the Each to Each technology separates prediction from analysis, allows predictions to be made using compact "models" produced by the analysis, and provides meaningful predictions after a person has provided just a few votes. This manual documents the Each to Each APIs and shows how to use them in a complete recommendation application.

- SRC Technical Note 1997-024

  Studying Balanced Allocations with Differential Equations
  Michael Mitzenmacher
  October 1, 1997.

  Using differential equations, we examine the GREEDY algorithm studied by Azar, Broder, Karlin, and Upfal for distributed load balancing. This approach yields accurate estimates of the actual load distribution, provides insight into the exponential improvement GREEDY offers over simple random selection, and allows one to prove tight concentration theorems about the loads in a straightforward manner.

- SRC Technical Note 1997-025a

  Recursive Object Types in a Logic of Object-oriented Programs
  K. Rustan M. Leino
  October 27, 1997. Revised January 12, 1998.

  This paper formalizes a small object-oriented programming notation. The notation features imperative commands where objects can be shared (aliased), and is rich enough to allow subtypes and recursive object types. The syntax, type checking rules, axiomatic semantics, and operational semantics of the notation are given. A soundness theorem, showing the consistency between the axiomatic and operational semantics is stated and proved. A simple corollary of the soundness theorem demonstrates the soundness of the type system. Because of the way types, fields, and methods are declared, no extra effort is required to handle recursive object types.

- SRC Technical Note 1997-026

  Specifying the Modification of Extended State
  K. Rustan M. Leino
  October 30, 1997.

  This paper explores the interpretation of specifications in the context of an object-oriented programming language with subclassing and method overrides, for example like Java. In particular, the paper considers annotations for describing what variables a method may change and the interpretation of these annotations. The paper shows that there is a problem to be solved in the specification of methods whose overrides may modify additional state introduced in subclasses. As a solution to this problem, the paper introduces data groups, which enable modular checking and rather naturally capture a programmer's design decisions.

- SRC Technical Note 1997-027

  A Simple, Intuitive Hypermedia Synchronization Model and its Realization in the Browser/Java Environment
  Jin Yu
  October 27, 1997. 12 pages.

  This paper presents a simple and intuitive hypermedia synchronization model – the Media Relation Graph (MRG), and an alternative implementation of the Hypermedia Presentation and Authoring System (HPAS), which is the testbed for MRG. Our model combines the power of both interval-based and point-based synchronization mechanisms. The new implementation exploits many rich features of commercial web browsers and reuses existing browser components, such as plugins and Java applets. (An overview of HPAS and its original Unix/C implementation is available.)

- SRC Technical Note 1997-029

  WebL – A Programming Language for the Web
  Thomas Kistler and Hannes Marais
  December 1, 1997.

  In this paper we introduce a programming language for Web document processing called WebL. WebL is a high level, object-oriented scripting language that incorporates two novel features: service combinators and a markup algebra. Service combinators are language constructs that provide reliable access to web services by mimicking a web surfer's behavior when a failure occurs while retrieving a page. The markup algebra extracts structured and unstructured values from pages for computation, and is based on algebraic operations on sets of markup elements. WebL is used to quickly build and experiment with custom web crawlers, meta-search engines, page transducers, shopping robots, etc.

- SRC Technical Note 1997-030

  Composition: A Way to Make Proofs Harder
  Leslie Lamport
  December 9, 1997. Corrected January 5, 1998.

Compositional reasoning about a system means writing its specification as the parallel composition of components and reasoning separately about each component. When distracting language issues are removed and the underlying mathematics is revealed, compositional reasoning is seen to be of little use.

- SRC Technical Note 1997-031

  Fast Integrated Tools for Circuit Design with FPGAs
  Stephan W. Gehring and Stefan H.-M. Ludwig
  December 16, 1997.

  To implement high-density and high-speed FPGA circuits, designers need tight control over the circuit implementation process. However, current design tools are unsuited for this purpose as they lack fast turnaround times, interactiveness, and integration. We present a system for the Xilinx XC6200 FPGA, which addresses these issues. It consists of a suite of tightly integrated tools for the XC6200 architecture centered around an architecture-independent tool framework. The system lets the designer easily intervene at various stages of the design process and features design cycle times (from an HDL specification to a complete layout) in the order of seconds.

- SRC Technical Note 1997-032

  A Semantic Approach to Secure Information Flow
  K. Rustan M. Leino and Rajeev Joshi
  December 17, 1997.

  A classic problem in security is the problem of determining whether a given program has secure information flow. Informally, this problem may be described as follows: Given a program operating on public and private variables, check whether observations of the public variables before and after execution reveal any information about the initial values of the private variables. Although the problem has been studied for several decades, most of the previous approaches have been syntactic in nature, often using type systems and compiler data flow analysis techniques to analyze program texts. This paper presents a considerably different approach to checking secure information flow, based on a semantic characterization of security. A semantic approach has several desirable features. Firstly, it gives a more precise characterization of security than that possible by conservative methods based on type systems. Secondly, it applies to any programming constructs whose semantics are definable; for instance, nondeterminism and exceptions pose no additional problems. Thirdly, it can be applied to reasoning about indirect leaking of information through variations in program behavior (e.g., whether or not the program terminates). The method is also useful in the context of automated verification, since it can be used to develop a mechanically-assisted technique for checking whether the flow of a given program is secure.

- SRC Technical Note 1997-033

  Strengthening Passwords
  Martín Abadi, T. Mark A. Lomas, and Roger Needham
  September 4, 1997 (with minor revisions on December 16, 1997).

  Despite their notorious vulnerability, traditional passwords remain important for security. In this paper we describe a method for strengthening passwords. Our method does not require users to memorize or to write down long passwords, and does not rely on smart-cards or other auxiliary hardware. The main cost of our method is that it lengthens the process of checking a password.

- SRC Technical Note 1998-001

  On the Analysis of Randomized Load Balancing Schemes
  Michael Mitzenmacher
  February 8, 1998.

  It is well known that simple randomized load balancing schemes can balance load effectively while incurring only a small overhead, making such schemes appealing for practical systems. In this paper, we provide new analyses for several such dynamic randomized load balancing schemes.

  Our work extends a previous analysis of the supermarket model, a model that abstracts a simple, efficient load balancing scheme in the setting where jobs arrive at a large system of parallel processors. In this model, customers arrive at a system of n servers as a Poisson stream of rate $\lambda n, lambda < 1$, with service requirements exponentially distributed with mean 1. Each customer chooses d servers independently and uniformly at random from the n servers, and is served according to the First In First Out (FIFO) protocol at the choice with the fewest customers. For the supermarket model, it has been shown that using $d = 2$ choices yields an exponential improvement in the expected time a customer spends in the system over $d = 1$ choice (simple random selection) in equilibrium. Here we examine several variations, including constant service times and threshold models, where a customer makes up to d successive choices until finding one below a set threshold.

Our approach involves studying limiting, deterministic models representing the behavior of these systems as the number of servers n goes to infinity. Results of our work include useful general theorems for showing that these deterministic systems are stable or converge exponentially to fixed points. We also demonstrate that allowing customers two choices instead of just one leads to exponential improvements in the expected time a customer spends in the system in several of the related models we study, reinforcing the concept that just two choices yields significant power in load balancing.

- SRC Technical Note 1998-002

  How Useful is Old Information?
  Michael Mitzenmacher
  February 8, 1998.

  We consider the problem of load balancing in dynamic distributed systems in cases where new incoming tasks can make use of old information. For example, consider a multi-processor system where incoming tasks with exponentially distributed service requirements arrive as a Poisson process, the tasks must choose a processor for service, and a task knows when making this choice the processor loads from T seconds ago. What is a good strategy for choosing a processor, in order for tasks to minimize their expected time in the system? Such models can also be used to describe settings where there is a transfer delay between the time a task enters a system and the time it reaches a processor for service.

  Our models are based on considering the behavior of limiting systems where the number of processors goes to infinity. The limiting systems can be shown to accurately describe the behavior of sufficiently large systems, and simulations demonstrate that they are reasonably accurate even for systems with a small number of processors. Our studies of specific models demonstrate the importance of using randomness to break symmetry in these systems and yield important rules of thumb for system design. The most significant result is that only small amounts of load information can be extremely useful in these settings; for example, having incoming tasks choose the least loaded of two randomly chosen processors is extremely effective over a large range of possible system parameters. In contrast, using global information can actually degrade performance unless used correctly; for example, unlike most settings where the load information is current, having tasks go to the least loaded server can significantly hurt performance.

- SRC Technical Note 1998-003

  A Digital Fountain Approach to Reliable Distribution of Bulk Data
  John W. Byers, Michael Luby, Michael Mitzenmacher, Ashutosh Rege
  February 8, 1998.

  The proliferation of applications that must reliably distribute bulk data to a large number of autonomous clients motivates the design of new multicast and broadcast protocols. We describe an ideal, fully scalable protocol for these applications that we call a digital fountain. A digital fountain allows any number of heterogeneous clients to acquire bulk data with optimal efficiency at times of their choosing. Moreover, no feedback channels are needed to ensure reliable delivery, even in the face of high loss rates.

  We develop a protocol that closely approximates a digital fountain using a new class of erasure codes that are orders of magnitude faster than standard erasure codes. We provide performance measurements that demonstrate the feasibility of our approach and discuss the design, implementation and performance of an experimental system.

- SRC Technical Note 1998-004

  Substitution: Syntactic versus Semantic
  Leslie Lamport
  March 11, 1998.

  A formalism with quantifiers permits two kinds of substitution: syntactic substitution that allows the capture of bound variables and semantic substitution that does not. When quantification is explicit, all substitution can be made semantic. When quantification is implicit, as in some formalisms used to reason about programs, both types of substitution are needed.

- SRC Technical Note 1998-005

  Reduction in TLA
  Ernie Cohen and Leslie Lamport
  March 10, 1998

  Reduction theorems allow one to deduce properties of a concurrent system specification from properties of a simpler, coarser-grained version called the reduced specification. We present reduction theorems based upon a more precise relation between the original and reduced specifications than earlier ones, permitting the use of reduction to reason about a larger class of properties. In particular, we present reduction theorems that handle general liveness properties.

2 Ordering Information

SRC Technical Notes are available from SRC's external publications web site:

`http://www.research.digital.com/SRC/publications/`

They are also available via anonymous ftp from internet node:

`gatekeeper.dec.com` (16.1.0.2)

The ftp path to them is:

`/pub/DEC/SRC/technical-notes/`

These notes are intended for electronic retrieval only.

If you'd like to be notified by electronic mail whenever a new publication of any kind is made available on SRC's web, please send email to:

`src-report@pa.dec.com`

with the words: "add to pub email" in the subject line of your message.

3 List of SRC Technical Notes

- 1994-001. *Introduction to TLA*
  Leslie Lamport

- 1996-001.*Refinement in state-based formalisms*
  Leslie Lamport

- 1996-002a. *The module structure of TLA+*
  Leslie Lamport

- 1997-001. *Virginity: A contribution to the specification of object-oriented software*
  K. Rustan M. Leino and Raymie Stata

- 1997-002. *Average-case analysis of first fit and random fit bin packing*
  Susanne Albers and Michael Mitzenmacher

- 1997-003. *Hypermedia presentation and authoring system*
  Jin Yu and Yuanyuan Xiang

- 1997-004a. *Fully dynamic 2-edge connectivity algorithm in polygarithmic time per operation*
  Monika Henzinger and Valerie King

- 1997-005a. *The Vesta-2 software description language*
  Allan Heydon, Jim Horning, Roy Levin, Timothy Mann, and Yuan Yu

- 1997-006a. *The Operators of TLA+*
  Leslie Lamport

- 1997-007. *Checking object invariants*
  K. Rustan M. Leino and Raymie Stata

- 1997-008. *A small dual-automorphic lattice with no involutory dual automorphism*
  K. Rustan M. Leino and Lyle Ramshaw

- 1997-009. *Juno-2 language definition*
  Greg Nelson and Allan Heydon

- 1997-010. *Focus+context displays of web pages: Implementation alternatives*
  Marc Brown, Johannes Marais, Marc A. Najork, and William E. Weihl

- 1997-011. *Tight thresholds for the pure literal rule*
  Michael Mitzenmacher

- 1997-012. *Online throughput-competitive algorithm for multicast routing and admission control*
  Ashish Goel, Monika Henzinger, and Serge Plotkin

- 1997-013. *Mobile ambient synchronization*
  Luca Cardelli

- 1997-014. *Exploring unknown environments*
  Susanne Albers and Monika Henzinger

- 1997-015. *Syntactic clustering of the Web*
  Andrei Broder, Steve Glassman, Mark Manasse, and Geoffrey Zweig

- 1997-016a. *Continuous profiling: Where have all the cycles gone?*
  Jennifer M. Anderson, Lance Berc, Jeffrey Dean, Sanjay Ghemawat, Monika Henzinger, Shun-Tak A. Leung, Dick Sites, Mark Vandevoorde, Carl Waldspurger, and William E. Weihl

- 1997-017. *Dynamic coscheduling on workstation clusters*
  Patrick G. Sobalvarro, Scott Pakin, William E. Weihl, and Andrew A. Chien

- 1997-018. *The 1995 SQL Reunion: People, Projects, and Politics*
  Paul McJones

- 1997-019. *Maintaining minimum spanning trees in dynamic graphs*
  Monika Henzinger and Valerie King

- 1997-020. *Improved data structures for fully dynamic biconnectivity*
  Monika Henzinger

- 1997-021. *Certificates and fast algorithms for biconnectivity in fully-dynamic graphs*
  Monika Henzinger and Han La Poutre

- 1997-022. *Short-length Menger theorems*
  Monika Henzinger, Jon Kleinberg, and Satish Rao

- 1997-023. *Each to Each programmer's reference manual*
  Paul McJones and John DeTreville

- 1997-024. *Studying balanced allocations with differential equations*
  Michael Mitzenmacher

- 1997-025a. *Recursive object types in a logic of object-oriented programs*
  K. Rustan M. Leino

- 1997-026. *Specifying the modification of extended state*
  K. Rustan M. Leino

- 1997-027. *A simple, intuitive hypermedia synchronization model and its realization in the browser/Java environment*
  Jin Yu

- 1997-028. *1997 SRC Intern Projects*
  Compiled by James Mason

- 1997-029. *WebL-a programming language for the Web*
  Thomas Kistler and Johannes Marais

- 1997-030a. *Composition: A way to make proofs harder*
  Leslie Lamport

- 1997-031. *Fast integrated tools for circuit design with FPGAs*
  Stephan W. Gehring and Stefan Ludwig

- 1997-032. *A semantic approach to secure information flow*
  K. Rustan M. Leino and Rajeev Joshi

- 1997-033. *Strengthening passwords*
  Martín Abadi, T. Mark A. Lomas, and Roger Needham

- 1998-001. *On the analysis of randomized load balancing schemes*
  Michael Mitzenmacher

- 1998-002. *How useful is old information?*
  Michael Mitzenmacher

- 1998-003. *A digital fountain approach to reliable distribution of bulk data*
  John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege

- 1998-004. *Substitution: Syntactic versus Semantic*
  Leslie Lamport

- 1998-005. *Reduction in TLA*
  Ernie Cohen and Leslie Lamport