UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

----------x

CAPITOL RECORDS, LLC,

Plaintiff,

Civil Action File No.
12 Civ 0095 RJS/AJP

-against-

REDIGI INC.,

Defendant.

DECLARATION

----------x

LARRY RUDOLPH (aka Lawrence S. Rogel) declares under penalty of perjury:

1. I am Chief Technical Officer and a founder of ReDigi Inc. I received my PhD in Computer Science from the Courant Institute of Mathematical Science, NYU in 1981. After a post-doc at University of Toronto, I joined the Computer Science faculty at Carnegie-Mellon University and then the Hebrew University, Jerusalem, Israel. After attaining the rank of full professor, I then moved to the Massachusetts Institute of Technology as a Principle Research Scientist. After 12 years, I joined VMWare as a Senior Staff Engineer in 2007 to start the Mobile Virtualization Project. In 2009, I left VMWare to found ReDigi where I serve as the CTO. A copy of my CV is attached hereto as exhibit A.

WHAT REDIGI DOES

(A) The upload and storage process

2. After signing up for a ReDigi account, accepting ReDigi's terms of service, downloading and installing ReDigi's proprietary "Music Manager" software ("Music Manager"),

and logging into the account using secure login information, a ReDigi user may upload an eligible music file ("Eligible File") from the user's computer to the user's personal storage locker ("Cloud Locker") in ReDigi's cloud-based storage system (the "ReDigi Cloud"). We have applied for a patent for our technology, and the patent application is pending (An abstract of our patent application attached hereto as exhibit B).

3. A user's Cloud Locker consists of file pointers, or record locators, that associate particular Eligible Files with a particular user's account and indicate in which Cloud Locker those files are located. *Only the user associated with a particular Cloud Locker has access to its contents.*

4. *The only Eligible Files are files originally and legally downloaded from iTunes*, or subsequently from ReDigi, thereby excluding music tracks copied from CDs, or downloaded from other online vendors or file sharers, or obtained from any other source. Moreover, only one instance of an Eligible File can ever be sold on ReDigi.

5. Music Manager analyzes each music file that a user seeks to upload to determine that it was legally downloaded from iTunes or ReDigi by the user and eligible to be uploaded. Before a file is accepted for upload, the file is subjected to an initial validation process on the user's computer, which includes analyzing file ownership, source, purchase dates, UITS code if it exists, metadata, and changes and modifications, if any. If the file passes the initial validation stage, the file is uploaded to the ReDigi server where additional and more intensive analysis takes place to confirm eligibility, including validating file source and ownership, and verifying that the file was not modified or tampered with.

6. Music Manager continuously runs in the background on a user's computer.

Upon the upload of an Eligible File to a user's Cloud Locker, such file and all copies thereof residing on the user's computer, and on attached synchronization and storage devices, are deleted. If the user were to attempt to upload the file without first accepting the prompt to delete the other copy or copies detected by Music Manager, the upload would be blocked.

7. If any storage or synchronization device is connected to the user's computer subsequent to the upload, Music Manager automatically searches the device for instances of copies of any Eligible File previously uploaded by the user to his or her Cloud Locker and, if a copy of such Eligible File is detected, the user is prompted to authorize the deletion of such file from that device. If the user fails to provide such authorization, his or her account is suspended.

8. Similarly, if a user downloads a file from his or her Cloud Locker, the file in the Cloud Locker is deleted.

9. After the upload of an Eligible File to a user's Cloud Locker, the user can privately listen to the file by "streaming" it from the user's Cloud Locker to an internet-connected device using the user's secure login to access his or her ReDigi account.

(B) The sale process

10. A user can choose to offer an Eligible File stored in his or her Cloud Locker for resale to other ReDigi users through the ReDigi used music marketplace. *This is the only type of used music sale transaction permitted on the ReDigi site. No copy of the file in the ReDigi Cloud is made when the Eligible File is sold by one ReDigi user to another ReDigi user.* When such a file is purchased by another user, the file pointer associating the Eligible File with the Cloud Locker of the selling user is modified to associate the file with the Cloud Locker of the purchasing user. In such a transaction only the pointer is changed; the Eligible File remains in

the same location in the ReDigi Cloud and is not copied.

11. After such a sale, the selling user no longer has any access to the file so sold. The Eligible File is stored in the purchasing user's Cloud Locker where he or she can continue to store it, listen to it, offer it for sale, or download it, in which case the file is deleted from the purchaser's Cloud Locker.

12. ReDigi earns a transaction fee on the sales.

(C) Streaming of 30 second clips and linking to artwork

13. ReDigi's website has links to 30-second clips which are streamed by a third-party source and to associated artwork maintained by that source, all pursuant to license. (See agreement with Rdio, Inc., John Ossenmacher declaration, exhibit B). No copies of such clips are made or stored on ReDigi's website or in the ReDigi Cloud, nor are any of the artwork files which are displayed on the website stored on the website or in the ReDigi Cloud. A user's ReDigi "memory bank" keeps track of links to which a user has listened. All that is stored in the memory bank are these bookmarks.

HOW IT WORKS

(A) Locker storage

14. The advent of inexpensive, nearly pervasive, highspeed, and continually accessible internet access has given rise to the rental of network storage, sometimes referred to as cloud storage or lockers in the cloud. Access to data stored over the internet is now sufficiently fast, that it can act like local storage. A user rents, or is otherwise provided with, space on machines located in a datacenter, and can upload or download the data.

15. This is a rapidly growing industry. Many companies provide cloud locker

storage to consumers, some for all kinds of files, some specifically designed for MP3's and designed to make it possible for users to privately stream their cloud-stored music (see John Ossenmacher declaration, exhibit A).

16. In addition to the cloud storage locker for mp3 and mp3-type aac (collectively "MP3") files ReDigi (a) offers a used digital music marketplace, and (b) provides technology and has adopted a business model which prevents users from maintaining duplicates, even on their home computers and connected devices, of recordings deemed eligible for ReDigi's services.

17. The locker is a conceptual idea. There is no actual locker. Rather, there are disk drives with a logical organization of a file system. Each uploaded file is stored in this file system. Access to the files in this file system is restricted. A file can be accessed only by the owner of that file. Ownership is signified by a system of file pointers which associate each file with a particular user. There is a database that controls access to the file as well as the needs of the used digital music marketplace. The database contains several tables which in turn contain records, with each record containing fields. The locker, pointers to files, ownership, and other details are realized through the values stored in the database.

(B) Verification of unique files

18. A ReDigi database contains records that specify details about each mp3 file, such as: the source, the track title, artist, album, year produced, as well as the user name of the file owner and the date of purchase, and the hash for the acoustic portion. Each music file has exactly one record in the database. In some cases the file has "UITS" code, which is an encrypted version of certain of the key data ReDigi would look at it in its verification process. The user's locker is the collection of all the files owned by the user, which have passed the verification

process, and are stored in the file system. It is specified as the collection of all the records which contain the user's identification. When a file is uploaded to a locker, several things happen: (a) the metadata and the acoustics are checked for validity; (b) the metadata are checked to ensure that no other ReDigi user has sold, has offered for sale, or is storing, a file with the same relevant metadata; (c) if the file fails either of these tests, it is discarded; (d) if it passes, then the file is stored on disk along with all the other music files; (e) the user's locker is updated to include the relevant metadata of the file and a "pointer" to the 'location' of the file; and (f) a record is made of the file's relevant metadata to ensure no other user attempts to upload a file with this same relevant metadata.

19. The file is the only instance of that unique file that remains on the server. The entire ReDigi cloud may contain multiple, unique copies of the same recording, although each unique file will have certain identifiable differences in the metadata, as each was uploaded by a different user at a different date and time. The source, iTunes, is also present in the metadata. The industry is introducing a "UITS" code into the meta-data that is a form of "purchase receipt" that uniquely identifies the instance of the song purchased. When present, we use that to detect if two files are the same purchase or two different purchases. The database records also record the status of the file, including whether this file is currently for sale or not. A file, while it is offered for sale, cannot be streamed or downloaded by the user. Only if the user withdraws the offer to sell, can the user stream or download the file.

20. The basic process used by our technology at this time is as follows: our software keeps track of (a) Purchaser (bought-by), (b) Store (apple), (c) Date Purchased, and (d) Hash value of the acoustics (we use a Sha1sum of the music portion of the music file). When this

data on one file matches the data on another, there would be a "violation". Different tests are done at two different locations, the client and the server.

21. At the client level, if the Music Manager software resident on the user's machine finds a file matching a file that is in the user's locker on the ReDigi site, or a file which has already been sold on the ReDigi site, it would term that a "violation", and require the user to delete the file from the client or delete it from the locker (and cancel any pending offer for sale); the user's failure to comply would mean suspension of the user's account.

22. At the server level, the software searches the entire server for a "violation". Files are ineligible to be in a user's locker or offered for sale, if that same file (same hash, bought-by, bought-from, and purchase date, or same hash and UITS) is in someone else's locker, offered by sale for someone else, or has been sold by someone else.

23. Files are ineligible to be uploaded if they do not contain bought-by, bought-from, and purchase date fields in the metadata, or the bought-from store is not iTunes or ReDigi (when one buys a used song through ReDigi, the bought-by is the ReDigi User ID and the bought-from is ReDigi).

24. Basically, every music file that the manager encounters is stored in the database, indexed by its hash. So, when the music manager sees a new file on the user's computer or attached device, it computes the hash and checks to see if this hash matches the hash of any song in the user's locker. If so, the other fields (purchaser, store, purchase date) are evaluated. If they match, a "violation" is found.

25. Apple iTunes has begun to use something called "UITS", encrypted code which includes the bought-by, bought-from, and purchase date information, as well as other data,

in encrypted form. If the file contains UITS code, our software will rely on the UITS, since the UITS has proven to be highly reliable. We do not decrypt the UITS, we rely on the encrypted code alone, since each UITS is unique.

26. Each ReDigi user is allocated locker space that can be used to store music files. Only music files that have passed ReDigi's verification process both by the client application on the user's personal computer and the server application on the ReDigi servers can be placed in the locker. Moreover, when a file is stored in the locker, there are no other copies of that file on the user's personal computer or attached devices. The locker can be thought of as the "final resting place for unwanted music files", unless of course the owner chooses to resurrect the file by downloading it to the owner's machine or listening to it while it is in the locker but not offered for sale. The ReDigi system is made for songs that the user no longer actively uses. A user might not be ready to offer a particular song for sale, but no longer wants it to take up space on his or her local computer. The marketplace may already have many instances of a particular recording offered for sale and so may not accept additional offers for sale of that recording. In such a case, the user can leave his or her song in the locker until supply and demand catch up with each other. ReDigi is also a "rebirth" place where the purchased file can reside before being downloaded by the new owner. It is, to the best of our knowledge, unique in that all the songs in the locker have passed verification. Many cloud storage systems tell their users to only upload items they own but, unlike ReDigi, offer no mechanism for verifying such ownership.

(C) Offering music for sale or to purchase

27. The ReDigi marketplace only allows songs to be offered for sale that meet two criteria. First, they must be in a user's locker. This ensures that they have passed all the

ReDigi verification steps. Second, they must be the only instance of the file that the user owns. The marketplace also allows a user to place an offer to purchase a song. Such a purchase request is either satisfied immediately or is queued up until some other user offers that song for sale. The user's client application, known as the music manager, communicates with the ReDigi server to synchronize its local information with the centralized information on the ReDigi database. A ReDigi user can at any time access a list of the songs in his or her locker that are not offered for sale, any of which can be downloaded, in which case they are deleted from the locker. Alternatively, the user may offer one of the songs for sale, in which case it can no longer be downloaded or streamed. The user can also at any time access a list of the songs in his or her locker that are offered for sale. The user can choose to cancel the offer for sale. In this case, it could then be downloaded, kept in storage, or re-offered for sale at a later time.

28. The client application program, the music manager, actually consists of two programs. One, the application program, interacts with the user. The second, is a continually running background "service" that constantly monitors music file creation, deletion, and renaming, as well as the files on removable devices when they become attached. This "service" program monitors the files on the user's computers and attached devices to ensure that the files sold, offered for sale, or residing in the locker, do not reappear on the computer. It automatically starts each time the user's machine is rebooted. Whenever the client application program is running, it makes sure the "service" program is also running, and if not, it attempts to restart it. The client application program cannot be used if the "service" program is not running.

29. The purchase and sale orders for a recording are organized on a first-in first-out basis. It may be the case that there are no outstanding orders, in which case the offer for

sale is placed at the end of a queue of offers for sale for this particular recording. Database records are used to maintain the desire-to-purchase and the offer-to-sell queues.

(D) Performing a transfer of ownership

30. The transfer of ownership is accomplished without making a copy of the recording. There is never a time when both seller and buyer have access to, or ownership of, the file. The record locator, or pointer, to the file that is in the seller's locker list is modified, so that it points to the seller's locker list. The transaction is, in computer parlance, an "atomic transaction". A "transaction" is a set of actions, insertions, deletions, or deletions, on the records of a database. In an "atomic" transaction, there is a guarantee that either all the transactions occur or none of them occur. An atomic transaction is set to commit at a particular instance or to fail. All the actions happen at the exact same moment the transaction commits. The transfer of ownership involves several actions. The song is transferred between the buyer and seller through the action of modifying the user identifier of the records in "box items", "fingerprints", "market sell items", and "market buy items" database tables. At no point is the copyrighted music file touched. Money or credit is transferred between the buyer and the seller, ReDigi, and other third parties. These transfers happen simultaneously with the transfer of ownership. They are all part of the atomic transaction.

WHEREFORE it is respectfully requested that plaintiff's motion be in all respects denied.

Dated: Cambridge, Massachusetts
January 27, 2012

LARRY RUDOLPH

Exhibit A

Curriculum Vita

Name: Larry Rudolph
Address: Redigi Inc
1 Broadway, 14th Floor
Cambridge, MA 02142
+1 617 800-3095
Email: larry@redigi.com, rudolph@csail.mit.edu

Education:

Courant Institute of Mathematical Sciences, New York University	PhD	1982
Courant Institute of Mathematical Sciences, New York University	MS	1978
Queens College, Flushing,	BA	1976

Title of Thesis for Most Advanced Degree:
Software Structures for Ultra-parallel Computing, 1981, Profs. J.T. Schwartz and A. Gottieb, Courant Institute; New York University, 1981.

Profession Appointments:

Founder and CTO	Redigi Inc	April 2010	present
Senior Staff Engineer	VMware Inc.	Sept 2007	April 2010
Affiliate	Mass Institute of Technology	Sept. 2008	present
Principal Research Scientist	Mass Institute of Technology	Sept. 1997	August 2008
Full Professor	Hebrew University	June 1995	Sept 1997
Professor	Hebrew University	June 1989	June 1995
Senior Lecturer	Hebrew University	June 1986	June 1989
Lecturer	Hebrew University	Oct. 1985	June 1986
Associate Researcher	Carnegie Mellon University	Sept. 1982	Oct. 1985
Postdoctoral Fellow	University of Toronto	Sept. 1981	Aug. 1982
Research Fellow	Singapore-MIT Alliance	2001	2006
Professor	New England Complex Systems Institute	Jan 1999	Present
Visiting Scientist	Mass Institute of Technology	Sept. 1994	Sept. 1997
Visiting Scientist	IBM TJ Watson	Sept. 1994	Sept. 1995
Visiting Scientist	IBM TJ Watson	June 1990	Oct. 1991
Adjunct Lecturer	New York University	Sept. 1979	Aug 1981

Other Relevant Experience:

Expert Witness (1997 --- Present) About a dozen cases, mostly dealing with computer architecture patent
Silicon Graphics Inc (6/03--9/03) Consultant, Parallel Processing; Contact Marty Deneroff
IBM Corp (6/96-1/00) Consultant: Parallel Processing Architectures; Contact: Dr. M. Snir
Thinking Machines Corp (1/95-8/95) Consultant: Scheduling and Job Management Software
Mercury Corp (12/94-3/95) Consultant: Network optimizations for their high-speed raceway interconnect.
Scitex Corp (11/92-6/93) Consultant: Strategic planing for next generation image processing systems.
Israeli Aircraft Industries (10/86-9/92) Consultant: Hardware and software development for real-time avionics.
Western Summer Institute (7/86-8/86) Lecturer: Advanced course on high performance computing for industrial.
AT&T Bell Labs (7/85-8/85) Consultant: Extracted statistical behavior from telephone trace log database
Digital Equipment Corp (9/84-9/85) Consultant: Designed a parallel processor, snoopy cache coherency mechanism

American Health Corp (4/75-9/77) Systems Analyst: Designed, coded, and implemented a time-sharing OS

Principal Fields of Interest:
Digital Personal Property, Digital Music, E-Books, Mobile phones and PDAs, Pervasive Computing, Parallel Computer Architectures, Job Scheduling, Operating Systems, Optical Communication, Complex Systems, Bluetooth

Recent Industry Products:
1. Mobile Phone Virtualization: I started research project in Mobile Phone Virtualization. The project has grown and evolved into a commercial product, to be installed on Samsung phones, 1st quarter 2012.
2. ReDigi: A startup company to provide A Marketplace for Used Digital Music. Founder and CTO.
3. Skatehub is developing hardware (accelerometers, gyros, bluetooth, battery) that is attached to a skateboard and communicates with cell phones. It detects the tricks and gestures performed by the board, for competition, training, and game control. Founder and CTO

Summary of my research direction

Just like people, computers can accomplish much more when they work together provided the synchronization, coordination, and communication overheads do not undermine the productivity gains. My research has always been concerned with understanding this tradeoff. My PhD thesis introduced the idea of combining memory update requests within the interconnection network in order to avoid "hot-spots." My research involving snoopy caching, dynamic scheduling, amortized cost-analysis, and free-space optical communications networks. All share the same feature, in that performance is improved through replication and resilience is improved though the ability of one component to pick up the slack when some other component is either slow, overloaded, or faulty. My more recent work on pervasive computing, applied this same theme to the arena in which people, computers, and mobile devices all work together.

Teaching Experience

MIT

Term	**Title**		**Course Type**
S 08, S 09, S 12	Computer System Engineering (6.033)		Undergrad
S 09	Virtualization Technology		4th year + Graduate
S 07	Mobile and Pervasive Computing		4th year + Graduate
S 06, S 05, F 04, S 04	Pervasive Human-Centric Computing		4th year + Graduate
F 03	Projects in Project Oxygen		Graduate
S 03, S 02, S 01	Computer System Engineering (6.033)		Undergrad
F 97, F 98, F 99	Structure and Interpretation of Computer Programs	(6.001)	Undergrad

Hebrew University (1986-1996)
Advanced Logic Design
Introduction to Computer Hardware
Compilers
Operating Systems
Parallel Algorithms
Parallelizing Compilers and systems issues in parallel processing,
Electro-holography uses in CS
Computers and Intellectual Property

Carnegie-Mellon University (1983-1986)

Current trends in algorithmic analysis and computer architecture

University of Toronto (1981-1982)
Theoretical Aspects of Concurrency

Other Teaching

Fall 1999	**New England Complex Systems Institute (NECSI)**	Complex Systems with Application to Aging
1986-1987	**Western Summer Institute (Stanford University)**	Parallel Processing
1984 & 1985	**Institute of Retraining in Computer Science**	Teaching math professors to teach CS

Public Service

2007:	Co-Organizer of "*Experimental Computer Science*" a Workshop at FCRC, 2007
2006:	Class on "Programming Cell Phones" open to the public
2005:	*"Must Smart Phones Catch Viruses"* Invited talk: FCC Technology Advisory Council
2005	Program Chair, International Conference of Supercomputing (ICS) 2005
2000-2005	Member of the CSAIL Oxygen Executive Committee
2001-present	General Chair of the CSAIL Student Workshop
2001-2002	Organized a seminar series on Pervasive Computing
2002	General Chair, Architectural Support of Programming Languages and Operating Systems (ASPLOS)

Service work in Israel – National Supercomputer Evaluation Committee; Organized Country-wide Conference in Computer Architecture; Governing Board Computational Neurobiology Center; Founding Member of the Ministry of Science Optical Computing Initiative.

Patents and Patent Applications:
"The Supersphere Electroholographic Architecture," US Patent # 6,542,264
"Optical Network" US 5,887,090
"Electro-holographic Optical Switch" WO 2000/002098, EP 1095317, US 6,542,364
"Adaptive cache coherence protocols" US Patent # 6,526,481, 6,757,787
"Computer architecture for shared memory access" US Patent # 6,636,950, 7,392,352
"System and method for performing memory operation in a computing system" US 7,925,839
"Virtualization with Merged Guest Page Table and Shadow Page Directory, US 20090300263
"Virtualization with Fortuitously Sized Shadow Page Tables, US 200903030264
"Virtualization with In-place Translation" US 20090300645
"Distributing Virtualization Software Address Space in Guest OS Address Space" US 20090300612
"In-place Shadow Tables for Virtualization", US 20090300611
"Preventing malware attacks in virtualized mobile devices" US 20100328064
"Controlling Usage in Virtualized Mobile Devices" US 20100330953
"Providing Security in Virtualized Mobile Devices" US 20100330961
"Migrating Functionality in Virtualized Mobile Devices" US 20100332635
"Virtualized Mobile Devices: US 20100333088
"Methods and Apparatus for Sharing, Transferring and Removing Previously Owned Digital Media" US 20110162086

Publications

Books

- Huang, A. and L. Rudolph, *Bluetooth Essentials for Programmers*, 2007 Cambridge University Press
- D. Feitelson,, E. Frachtenberg, L. Rudolph, and U. Schwiegelshohn editors *Proceedings of the 11th Workshop on Supercomputing Job Scheduling*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 3834. 2005.
- D. Feitelson,, L. Rudolph, and U. Schwiegelshohn editors *Proceedings of the 10th Workshop on Supercomputing Job Scheduling*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 3277. 2004.

- D. Feitelson, L. Rudolph, and U. Schwiegelshohn editors *Proceedings of the 9th Workshop on Supercomputing Job Scheduling*, Springer-Verlag, Lecture Notes in Computer Science, Vol. . 2003.
- D. Feitelson, L. Rudolph, and U. Schwiegelshohn editors *Proceedings of the 8th Workshop on Supercomputing Job Scheduling*, Springer-Verlag, Lecture Notes in Computer Science, Vol. . 2002.
- D. Feitelson and L. Rudolph, editors *Proceedings of the 7th Workshop on Supercomputing Job Scheduling,* Springer-Verlag, Lecture Notes in Computer Science, Vol. 2537. 2001.
- D. Feitelson and L. Rudolph, editors *Proceedings of the 6th Workshop on Supercomputing Job Scheduling*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 1911. 2000.
- D. Feitelson and L. Rudolph, editors *Proceedings of the 5th Workshop on Supercomputing Job Scheduling*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 1659. 1999.
- D. Feitelson and L. Rudolph, editors *Proceedings of the 4th Workshop on Supercomputing Job Scheduling*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 1459. 1998.
- D. Feitelson and L. Rudolph, editors *Proceedings of the 3rd Workshop on Supercomputing Job Scheduling*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 1291. 1997.
- D. Feitelson and L. Rudolph, editors *Proceedings of the 2nd Workshop on Supercomputing Job Scheduling*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 1162. 1996.
- D. Feitelson and L. Rudolph, editors *Proceedings of the Workshop on Supercomputing Job Scheduling*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 998. 1995.
- G. Lerman and L. Rudolph, "*Parallel Evolution of Parallel Machines*" Plenum Press, 1993.

Papers in Refereed Conferences and Journals.

2009

- Larry Rudolph, A Virtualization Infrastructure that Supports Pervasive Computing, Pervasive Computing, Vol 8 Issue 4, pp. 8-13
- Hui Fang, Wen-Jing Hsu, Larry Rudolph: Mining User Position Log for Construction of Personalized Activity Map. ADMA 2009: 444-452

2008

- Qin Zhao, Rodric M. Rabbah, Saman P. Amarasinghe, Larry Rudolph, Weng-Fai Wong: How to Do a Million Watchpoints: Efficient Debugging Using Dynamic Instrumentation. CC 2008: 147-162
- Hui Fang, Wen-Jing Hsu, Larry Rudolph: Controlling Uncertainty in Personal Positioning at Minimal Measurement Cost. UIC 2008: 468-481
- Chris Hill, L. Rudolph, "Zen" and the art of petascale ocean modeling" HPCVirt'08 ACM, pp. 32-39

2007

- Zhao, Qin, R. Rabbah, S. Amarasinghe, W.F. Wong, and L. Rudolph, "Ubiquitous Memory Introspection" 2007 International Symposium on Code Generation and Optimization (CGO) March 2007.
- Fang, H. W.J. Hsu, and L. Rudolph, "Group Protocols for Peers-Based Systems: A Case Study," 15th Euromicro Conference on Parallel, Distributed and Network-based Processing (PDP), Feb, 2007.
- Larry Rudolph, Daniel A. Menascé, Fabián E. Bustamante, Catherine C. McGeoch, Ethan L. Miller: Report on education roundtable: experimentaion in the computer science curriculum. Experimental Computer Science 2007: 9

2006

- Zhao, Qin, J.E. Sim, W.F. Wong, and L. Rudolph, "DEP: detailed execution profile," Proceedings of the 15th International Conference on Parallel Architectures and Compilation Techniques (PACT), June, 2006, pp154-163.
- Adam J. Oliner, Larry Rudolph, Ramendra K. Sahoo: Cooperative checkpointing: a robust approach to large-scale systems reliability. ICS 2006: 14-23
- Peserico, E. and L. Rudolph, "Robust Network Connectivity: When it's the big picture that matters," International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS), June, 2006, pp. 290-310
- A.J. Oliner, L. Rudolph, and R.K. Sahoo, "Cooperative Checkpointing Theory" International Parallel and Distributed Symposium (IPDPS), April 2006
- Fang, H. W.J. Hsu, and L. Rudolph, "Relaxing Routing Table to Alleviate Dynamism in P2P Systems, " SMA 2006.
- Zhenghao, C. and L. Rudolph, "Modeling Information Flow in Face-to-face Meetings While Protecting Privacy", 2006

2005

- Huang, A. and L. Rudolph, A Privacy Conscious Bluetooth Infrastructure for Location Aware Computing, SMA 2005 Symposium, Singapore, Jan. 2005
- Ong, C.H., N. Kasim, S.K.B. Jayasena, L. Rudolph, and T.J. Cham, "Proactive Detection and Recovery of Lost Mobile Phones", SMA 2005 Symposium, Singapore, Jan 2005
- J. Oliner, L. Rudolph, R. K. Sahoo, J. E. Moreira, and M. Gupta. "Probabilistic QoS Guarantees for Supercomputing Systems". In Proceedings of the International Conference on Dependable Systems and Networks (DSN 2005). Yokohama, Japan.
- Huang, A., K. Puli, and L. Rudolph, "Kimono: Kiosk-Mobile Phone Knowledge Sharing" International Conference on Mobile and Ubiquitous Multimedia (MUM 2005)

2004

- G. E. Suh, L. Rudolph, and S. Devadas, "Dynamic Partitioning of Shared Cache Memory", The Journal of Supercomputing, 28(1), pages 7-26, April 2004.
- Kim, Hana, Nancy Kho, Emily Yan, and Larry Rudolph, "Commanimation: Creating and Managing Animations via speech," Proceedings of 4th Annual SMA Symposium, 2004 pp 50-59.
- D. G. Feitelson, L.Rudolph, and U. Schwiegelshohn, ``Parallel job scheduling --- a status report". In Job Scheduling Strategies for Parallel Processing, D. G. Feitelson, L. Rudolph, and U. Schwiegelshohn (Eds.), pp. 1-16, Springer-Verlag, 2004. Lecture Notes in Computer Science Vol. 3277.
- Champaneria, A., and Rudolph, L., "PADCAM: A Human-Centric Perceptual Interface for Temporal Recovery of Pen-Based Input," Proceedings 2004 AAAI Fall Symposium "Making Pen-Based Interaction Intelligent and Natural," October 21-24, Washington DC, AAAI Press, 2004, pp. 35-41.

2003

- Rudolph, L., "What to do when debugging is not an option" Technology Review, April 2003
- Rudolph, L., "What I did on my Fall Vacation – A Pervasive Computing Class" IEEE Pervasive, Vol 2, No. 2, June 2003, pp 100-104.
- Chen, D., Peserico, E., and Larry Rudolph, "A Dynamically Partitionable Compressed Cache," Proceedings of 3rd Annual SMA Symposium, 2003 pp 128-135.

2002

- Maciej Drozdowski, Ioannis Milis, Larry Rudolph, Denis Trystram: Scheduling and Load Balancing. Euro-Par 2002, pp. 187-188
- G. E. Suh, S. Devadas, and L. Rudolph, "A New Memory Monitoring Scheme for Memory-Aware Scheduling and Partitioning", Proceedings of the High Performance Computer Architecture 8 (HPCA8), pages 117-118, Boston, MA, February 2002.

2001

- Rudolph, L., "Project Oxygen: Pervasive, Human-Centric Computing - An Initial Experience," Conference on Advanced Information Systems Engineering (CaiSE) 2001, pp. 1-12
- Prabhat Jain, Srinivas Devadas, Daniel W. Engels, Larry Rudolph: Software-Assisted Cache Replacement Mechanisms for Embedded Systems. Proceedings of the Int'l Conference on Computer-Aided Design (ICCAD) 2001, pp. 119-126
- Burkhard Englert, Larry Rudolph, Alexander A. Shvartsman: Developing and Refining an Adaptive Token-Passing Strategy. ICDCS 2001, pp. 597-605
- G. E. Suh, S. Devadas, and L. Rudolph, "Analytical Cache Models with Application to Cache Partitioning", Proceedings of the 15th International Conference on Supercomputing, pages 1-12, Sorrento, Italy, June 2001.
- G. E. Suh, L. Rudolph, and S. Devadas, "Effects of Memory Performance on Parallel Job Scheduling", Proceedings of the 7th International Workshop on Job Scheduling Strategies for for Parallel Processing (JSSPP 2001), Lecture Notes in Computer Science (LNCS), Vol. 2221, pages 116-132, Cambridge, MA, June 2001.
- Derek Chiou, Srinivas Devadas, Josh Jacobs, Prabhat Jain, Vinson Lee, Enoch Peserico, Peter Portante, Larry Rudolph, Scheduler-Based prefetching for Multilevel Memories, JSSPP 2001 pp. 53-62.
- G. E. Suh, L. Rudolph, and S. Devadas, "Dynamic Cache Partitioning for Simultaneous Multithreading Systems", Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'01), pages 635-641, Anaheim, CA, August 2001. Best paper award.

2000

- Chiou, D., P. Jain, S. Devadas and L. Rudolph, "Application-Specific Memory Management for Embedded Systems", In proceedings of the Design Automation Conference (DAC), June 2000, pp. 416-419.
- Rudolph, L. and P. Smith, "Valuation of Ultra-scale Computing Systems," Job Scheduling Strategies for Parallel Processing, IPDPS 2000, pp. 39-55
- Englert, B., L. Rudolph, and A. Shvartsman "Developing and Refining an Adaptive Token-Passing Strategy" Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS) 2001, pp. 597-605.
- Feitelson, D.G. and L. Rudolph: Job Scheduling Strategies for Parallel Processing, IPDPS 2000 Workshop, JSSPP 2000, Cancun, Mexico, May 1, 2000, Proceedings Springer 2000
- Ang, B.S., D. Chiou, L. Rudolph, and Arvind, "Micro-Architectures of High Performance, Multi-User System Area Network Interface Cards." International Parallel and Distributed Systems (IPDPS) 2000: pp. 13-20

1999

- Xiaowei Shen, Arvind, and Larry Rudolph, Commit-Reconcile & Fences (CRF): A New Memory Model for Architects and Compiler Writers, in proceedings of the 26th International Symposium on Computer Architecture (ISCA), May 1999, pp. 150-161.
- Xiaowei Shen, Arvind, and Larry Rudolph, CACHET: An Adaptive Cache Coherence Protocol for Distributed Shared-Memory Systems, In proceedings of the 13th ACM SIGARCH International Conference on Supercomputing (ICS), June 1999, pp. 135-144.

1998

- Allan Gottlieb, Ralph Grishman, Clyde P. Kruskal, Kevin P. McAuliffe, Larry Rudolph, Marc Snir: The NYU Ultracomputer - Designing a MIMD, Shared-Memory Parallel Machine. 25 Years ISCA: Retrospectives and Reprints 1998: 239-254
- Daniel Citron, Dror G. Feitelson, Larry Rudolph: Accelerating Multi-Media Processing by Implementing Memoing in Multiplication and Division Units. ASPLOS 1998: 252-261
- Boon Seong Ang, Derek Chiou, Larry Rudolph, Arvind: The StarT-Voyager Parallel System. In proceedings of International Conference on Parallel Architectures and Compilation Techniques (PACT) 1998: pp. 13-17.
- Dror G. Feitelson, Larry Rudolph: Metrics and Benchmarking for Parallel Job Scheduling. JSSPP 1998: pp. 1-24
- Boon S. Ang, Derek Chiou, Larry Rudolph, and Arvind, Message Passing Support on StarT-Voyager, In proceedings of the 5th International Conference on High-Performance Computing (HiPC), 1998, pp. 17-20
- Larry Rudolph and D.G. Feitelson, Metrics and Benchmarking for Parallel Job Scheduling, Appears in Job Scheduling Strategies for Parallel Processing, Springer Verlag LNCS Volume 1459(1998), pp. 1-12.
- Larry Rudolph, Do parallel Computers Really Need Optical Interconnection Networks?, In proceedings of The Fifth International Conference on Massively Parallel Processing Using Optical Interconnections, Las Vegas, June 1998.
- Larry Rudolph and D.G. Feitelson, Job scheduling for parallel supercomputer, In Encyclopedia of Computer Science and Technology, A. Kent and J. G. Williams (eds.), Vol. 38, pp. 287-314, Marcel Dekker, Inc., 1998.
- Boon S. Ang, Derek Chiou, Daniel Rosenband, Mike Ehrlich, Larry Rudolph, and Arvind, StarT Voyager: A Flexible Platform for Exploring Scalable SMP Issues, In proceedings of SuperComputing, 1998,

1997

- Hubertus Franke, Pratap Pattnaik, Larry Rudolph: Gang Scheduling for the IBM SP{2 Workstation Cluster. HICSS (1) 1997: pp. 630-631
- Dror G. Feitelson, Larry Rudolph, Uwe Schwiegelshohn, Kenneth C. Sevcik, Parkson Wong: Theory and Practice in Parallel Job Scheduling. JSSPP 1997: pp. 1-34
- Walter Lee, Matthew Frank, Victor Lee, Kenneth Mackenzie, Larry Rudolph: Implications of I/O for Gang Scheduled Workloads. JSSPP 1997: pp. 215-237

1996

- Hubertus Franke, Fang Wang, Pratap Pattnaik, Larry Rudolph “Gang Scheduling for Multiprocessor Systems” ICSE, Las Vegas, 1996.
- Hubertus Franke, Pratap Pattnaik, Larry Rudolph “Gang Scheduling for the IBM SP2 Workstation Cluster” 30th Hawaii International Conference on System Sciences January 1996.
- Ayyad, Yakov Exman, Martin Land, Larry Rudolph “An Experimental Crosspoint Switch For Support Of Collective Communications In Parallel Processing” The 19th Convention of IEEE in Israel, November, 1996.
- Hubertus Franke, Pratap Pattnaik, Larry Rudolph “Gang Scheduling for Highly Efficient Multiprocessors,” Sixth Symposium on the Frontiers of Massively Parallel Computation, Oct. 1996, Annapolis, Maryland.
- Y. Ben Asher, D. Feitelson, and L. Rudolph, “PARC --- An Extension of C for Shared Memory Parallel Processing,” Software Practice and Experience, Vol. 26(5), May 1996, pp. 581-612.
- Dror G. Feitelson and Larry Rudolph, “Evaluation of design choices for gang scheduling using distributed hierarchical control,” Journal Parallel and Distributed Computing, Vol. 35(1), May 1996, pp. 18-34.

1995

- Citron, D. and L. Rudolph, “Creating A Wider Bus Using Caching Techniques,” First International Symposium on High Performance Computer Architecture, Jan., 1995.
- D. Feitelson and L. Rudolph, “Parallel Job Scheduling: Issues and Approaches,” in Proceedings of the Workshop on Supercomputing Job Scheduling, SprInger-Verlag, Lecture Notes in Computer Science, Vol. 998. 1995.
- L. Rudolph, “Bit-Parallel, Free-Space, Optical Communication,” in The Houses That Jack Built; Essays in honor of Jack Schwartz, M. Davis, A. Gottlieb, E. Schonberg eds. pp. 271-286, 1995.
- Y. Romem, L. Rudolph and J. Stein, “Adapting Multilevel Simulated Annealing for Mapping Dynamic Irregular Problems,” Workshop for Parallel Algorithms for Irregular Mesh Problems, associated with IPPS ‘95. Sanjay RankaEds.
- Dror G. Feitelson and Larry Rudolph “Coscheduling based on runtime identification of activity working sets,” Intl. J Parallel Programming Vol. 23(2), Apr. 1995, pp. 135-160.
- D. G. Feitelson, L. Rudolph, and E. Schenfeld, “A three-dimensional optical interconnection network with distributed control”. Intl. J. Optoelectronics Vol. 10 no. 3, 1995, pp. 163--177.
- L. Rudolph, “Bit-Parallel, Free-Space, Optical Communication,” Communications on Pure and Applied Mathematics, Vol. XLVIII, Number 9/10, September/October 1995, pp. 1157-1173.

1994

- L. Rudolph and E. Upfal, “Efficient Dynamic Broadcasting Schemes—Analysis, Simulations, and Experiments,” DIMACS workshop on Parallel Algorithms and Network Problems,, Feb. 1994, Rutgers, NJ.
- L. Rudolph, “Adaptivity—The Future of Parallelism,” in Utrecht Computational Science 1994 Symposium R. Bisseling ed. Nov. 1994.
- N. Beaucoudrey, P. Chavel, I. Exman, and L. Rudolph, “Optical Implementation of Collective Communications,” 9th Annual Optical Engineering Conference, Jerusalem, Oct. 1994.

1993

- Rudolph, L., “Hardware Support for Collective Communication Operations,” in it Parallel Architectures and their Efficient Use (Heinz Nixdorf Workshop on Parallel Processing Paderborn, Germany, 1992) Lecture Notes in Computer Science #678, F Meyer auf der Heide, B. Monien, and A. L. Rosenberg (Eds.), Springer Verlag, 1993, pp. 110-119.
- Bala, V., S. Kipnis, L. Rudolph, M. Snir, “Designing Efficient, Scaleable and Portable Collective Communication Libraries,” SIAM conference on High Performance Parallel Processing, March 1993.
- Parnas, H. and L. Rudolph, “Representation of Synapses in Neuronal Networks,” in Mathematics Applied to Biology and Medicine J. Demongeot and V. Capasso (Eds)., Wuerz Publishing, Winnipeg, Canada, 1993, pp. 137-148.
- Sarukkai,S., D. Kimelman, and L. Rudolph, “A Methodology for Visualizing Performance of Loosely Synchronous Programs,” Accepted for publication in Journal of Parallel and Distributed Computing, 1993.
- Pickover, C. and L. Rudolph, “Electronic Kaleidoscopes for the Mind,” Computer Graphics Forum, Vol. 12, March 1993, pp. 41-47.

1992

- Beaucoudrey, N, A. Bellemain, D. Phalippou, P. Chavel, D. Fortin, I. Exman, L. Rudolph, J.P. Schnell, J.P. Pocholle, "Optical Interconnects for Parallel Systems: Demonstration of an Optical Link using Multiple-Quantum Wells Opto-Electronic Arrays," Parallel Architectures and Languages Europe (PARLE '92), Paris, France, June, 1992
- Sarukkai,S., D. Kimelman, and L. Rudolph, "A Methodology for Visualizing Performance of Loosely Synchronous Programs," Scaleable High Performance Computing Conference (SHPCC '92), Williamsburg, Virginia, April, 1992.
- Feitelson D., Y. Ben Asher, M. BenEzra, I. Exman, L. Picherski, L. Rudolph, D. Zernik, "Issues in Run-Time Support for Tightly Coupled Parallel Processing," Symposium on Experiences with Distributed and Multiprocessor Systems (SEDMS) III, March 26-27, 1992 pp. 27-42.
- Shteiman, K., D. Feitelson, L. Rudolph, I. Exman, "Envelopes in Adaptive Local Queues for MIMD Load Balancing," CONPAR92-VAPP V, July 1992, pp 479-484.
- Exman and L. Rudolph, "On-the-fly message patching in free space optical interconnects," 8th Meeting on Optical Engineering in Israel: Optoelectronic and Applications in Industry and Medicine, Tel Aviv, SPIE Vol. 1972, pp.198-210.
- D, Zernik and L. Rudolph "Toward Coherent Debugging of Shared-Memory Programs: a Systematic Approach," To appear in IEEE Software, Special Issue on Testing and Debugging, 1992.
- D. Feitelson and L. Rudolph, "Gang Scheduling Performance Benefits for Fine-Grain Synchronization," Journal of Parallel and Distributed Computing, Vol. 16, 1992, pp. 306-318.

1991

- L. Rudolph, M. Slivkin-Allalouf, and E. Upfal, "A Simple Load Balancing Scheme for Task Allocation in Parallel Machines," Symposium on Parallel Algorithms and Architectures, July 1991, pp. 237-245.
- L. Rudolph, "The Role of Optics in Future Parallel Processors," The International Congress on Optical Science and Engineering The Hague, SPIE vol. 1505, pp: 175-185, March 1991.
- D. Zernik and L. Rudolph, "Using Visual Tools for Developing Asynchronous Parallel Layout Algorithms," International Conference on Parallel Processing, Vol. II pp:280-281, 1991.
- S. Sprinzak, H. Parnas, and L. Rudolph, "Experiments and Simulations using the SONN system," From Network to Neuron Bat-Sheva Conference, Jerusalem, Israel, 1991.
- D. Zernik and L. Rudolph, "Animating Work and Time for Debugging Parallel Programs - Foundation and Experience," ACM ONR Workshop on Parallel and Distributed Debugging May 1991 .
- D. Zernik, L. Rudolph, and M. Snir, "Aspects of the Parallel Program Execution: Work, Time, and the Current State," The Fifth Israel Conference on Computer Systems and Software Engineering, Herzlia, Israel, May 1991, pp.162-172.

1990

- D. Feitelson, E. Schenfeld, and L. Rudolph, "Limitations on Free-Space Optical Interconnection Networks," to appear in Proceedings of the 3rd International Congress on Optical Science and Engineering, The Hague, The Netherlands, March 1990.
- L. Rudolph, D. Feitelson, and E. Schenfeld, "A 3-D Interconnection Network Based on Static and Dynamic Routing Devices," to appear in Optical Interconnections and Networks, SPIE Proc. 1281, Mar 1990.
- E. Schenfeld, D. Feitelson, and L. Rudolph, "An Optical Free-Space Interconnection Network for Parallel Computers," to appear in Optical Interconnections and Networks, SPIE Proc. 1281, Mar 1990.
- D. Feitelson and L. Rudolph, "Architecture for a Multi-User General-Purpose Parallel System," Computer Architecture News 17, Dec 1989 pp. 50-56.
- D. Feitelson and L. Rudolph, "Mapping and Scheduling in a Shared Parallel Environment Using Distributed Hierarchical Control," International Conference on Parallel Processing, Aug 1990.
- D. Feitelson and L. Rudolph, "Wasted Resources in Gang Scheduling," Proceedings of the Fifth Jerusalem Conference on Information Technology Oct. 1990, pp 127-136.
- C. Kruskal, L. Rudolph and M. Snir, "Efficient parallel algorithms for graph problems," Algorithmica Vol. 5 (1990), pp. 43-64.
- D. Feitelson and L. Rudolph, "Distributed Hierarchical Control for Parallel Processing," Computer, Vol. 23, No. 5, (May 1990), pp 65-77.

1989

- R. Rubin, L. Rudolph, and D. Zernik, “Holistic Debugging” Ilah Conference on Software Engineering, Jerusalem, Israel, 1989, pp. 75-87. It was one of the papers selected to be published in the Lecture-Notes in Computer Science by Springer-Verlag.
- D. Fietelson and L. Rudolph, “Algoritms and Architectural Support for Paging and Virtual Memory Management with Distributed hierarchical Control.” International Symposium on Computer Architecture (1989).
- C. Kruskal, L. Rudolph and M. Snir, “Techniques for Parallel Manipulation of Sparse Matrices,” Theoretical Computer Science, Vol. 64 (1989), pp. 135-157.
- D. G. Feitelson and L. Rudolph “Implementation of a Wait-Free Synchronization Primitive That Solves n-Process Consensus.” Information Processing Letters, Vol. 32 (July 1989), pp. 81-83.
- Kalay, S. Hochstein, and L. Rudolph, “Simulation of Early Processing of Visual Contours,” in Seeing Contour and Colour, J.J. Kulikowski, C.M. Dickinson, and I.J. Murray, editors, Pergamon Press, 1989, pp.687-697.
- M. Dowd, Y. Perl, L. Rudolph, and M. Saks, “The Periodic Balanced Sorting Network,” Journal of the ACM, Vol. 36, (Oct. 1989), pp. 738-757.

1988

- C. Kruskal, L. Rudolph and M. Snir, “Techniques for Parallel Manipulation of Sparse Matrices,” in High Performance Computer Systems, proceedings of the Intl. Symp. on High Performance Computer Systems, Paris, France, Elsevier Science Publishers B.V. (North-Holland), Ed. E. Gelenbe, 1988, pp. 3-14;
- C. Kruskal, L. Rudolph, and M. Snir “A Complexity Theory of Efficient Parallel Algorithms” ICALP (1988).
- R. Rubin, L. Rudolph, and D. Zernik, “Debugging Parallel Programs in Parallel,” Workshop on Parallel and Distributed Debugging, ACM Sigplan Sigops, Chicago, 1988.
- R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator “Optimal algorithms for distributed shared memory,” Algorithmica Vol. 3 (1988), pp. 79-119.
- Kruskal, L. Rudolph and M. Snir, “Efficient synchronization on multiprocessors with shared memory,” ACM Transactions on Programming Languages and Systems Vol. 10, No. 4, (1988) pp. 579-601.

1987

- Kalay, S. Hochstein, and L. Rudolph, “Neural Patterns as Detection Units in Sensory Systems,” First Annual International Conference on Neural Networks, (1987).
- Kalay, S. Hochstein, and L. Rudolph, “Simulation of Early Processing of Visual Contours,” International Conference on Vision and Computation, (1987).
- Mitrani, A. Greenberg, and L. Rudolph, “Analysis of Snoopy Caching,” Performance 87: The 12th IFIPWG 7.3 International Symposium on Computer Performance, P.-J. Courtois and G. Latouche editors, (1987)Brussels, Belgium, North Holland, pp. 345-362.
- Peters and L. Rudolph “Parallel Approximation Schemes for Subset Sum and Knapsack Problems,” ACTA INFORMATICA Vol. 24 (1987), pp. 417-432.
- R. Kannan, L. Rudolph, and G. Miller, “Sublinear Parallel Algorithm for Computing the Greatest Common Divisor of Two Integers,” SIAM Journal on Computing, Vol. 16, No. 1 (1987), pp. 7-16.
- Rudolph, “The importance of 2-dimensional shuffle-exchange in visual processing,” Bat-Sheva Seminar on Selective Attention in Sensory Processing, Jan. 5-15, 1987, held at the Hebrew University, Jerusalem, Israel.
- Gottlieb, R. Grishman, C.K. Kruskal, K.P. McAuliffe, L. Rudolph, and M. Snir, “The NYU Ultracomputer -Designing a MIMD, shared-memory parallel computer” in Tutorial: Computer Architecture edited by D.D. Gajski, V.M. Milutinovic, H.J. Siegel, and B.P. Furht, IEEE Computer Press, 1987.

1986

- Nudler and L. Rudolph, “Tools for the Efficient Development of Efficient Parallel Programs,” First Israeli Conference on Computer Systems Engineering, (1986) pp. 4.1.1-4.1.2.
- C. Kruskal, T. Madej, and L. Rudolph, “Parallel Prefix on Fully Connected Direct Connection Machines,” International Conference on Parallel Processing (1986), pp. 278-284.
- C. Kruskal, L. Rudolph, and M. Snir, “Efficient Parallel Algorithms for Graph Models,” International Conference on Parallel Processing (1986), pp 869-876.

- Y. Gil and L. Rudolph, “Counting and Packing in Parallel” International Conference on Parallel Processing (1986), pp 1000-1003.
- R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator “Optimal algorithms for distributed shared memory,” 27th Annual Symposium on Foundations of Computer Science, (1986) pp.
- Kruskal, L. Rudolph and M. Snir, “Efficient synchronization on multiprocessors with shared memory,” Fifth Annual Conference on Principles of Distributed Processing, (1986) pp. 218-228.

1985

- C. P. Kruskal, R. Cytron, and L. Rudolph, “The Architecture of Parallel Computers,” presented at the International Summer School on Control Flow and Data Flow: Concepts of Distributed Programming July 1984, and N.A.T.O. Scientific Series, Springer-Verlag, 1985.
- J. Edler, A. Gottlieb, C. Kruskal, K. McAuliffe, L. Rudolph, M. Snir, P. Teller, and J. Wilson, “Issues Related to MIMD, Shared-Memory Computers: The NYU Ultracomputer Approach,” 12th International Conference on Computer Architecture (1985), pp. 126-135.
- L. Rudolph and W. Steiger, “Subset Size in Parallel,” International Conference on Parallel Processing (1985) pp. 11-13.
- C. P. Kruskal, L. Rudolph and M. Snir, “The Power of Parallel Prefix,” International Conference on Parallel Processing (1985) pp. 180-185.
- L. Rudolph, “A Robust Sorting Network,” IEEE Transactions on Computers, Vol. C-34, No 4(1985), pp. 326-343.
- C. Kruskal, L. Rudolph and M. Snir, “The Power of Parallel Prefix,” IEEE Transactions on Computers Vol. C-34, No. 10 (1985), pp. 965-968.
- Z. Segall and L. Rudolph, “PIE - A Programming and Instrumentation Environment for Parallel Processing,” IEEE Software, Vol. 2, No. 6 (1985) pp. 22-37.

1984

- Frieze and L. Rudolph “A Parallel Algorithm for All Pairs Shortest Paths in a Random Graph,” Twenty-Second Annual Allerton Conference on Communication, Control, and Computing (1984), pp.663-670.
- R. Kannan, L. Rudolph, and G. Miller, “Sublinear Parallel Algorithm for Computing the Greatest Common Divisor of Two Integers,” 25th Annual Foundations of Computer Science (1984) pp. 7-11.
- Gottlieb, R. Grishman, C.K. Kruskal, K.P. McAuliffe, L. Rudolph, and M. Snir, “The NYU Ultracomputer -Designing a MIMD, shared-memory parallel computer” In Kai Hwang and Bob Kung (eds.), Supercomputers: Design and Applications IEEE Computer Press, 1984.

1983

- Gottlieb, R. Grishman, C.K. Kruskal, K.P. McAuliffe, L. Rudolph, and M. Snir, “The NYU Ultracomputer -Designing a MIMD, shared-memory parallel computer” IEEE Transactions on Computers C-32 (1983), pp. 175-189.
- E. Fiume, A. Fournier, and L. Rudolph, “A Parallel Scan Conversion Algorithm with Anti-Aliasing for a General Purpose Ultracomputer,” Computer Graphics, Vol. 17, No. 3, (1983), pp. 141-149.
- L. Rudolph and Z. Segall, “A Dynamic Decentralized Cache Scheme for an MIMD Parallel Processor,” 11th Annual Computing Architecture Conference (1983), pp. 340-347.
- M. Dowd, Y. Perl, L. Rudolph, and M. Saks, “The Balanced Sorting Network,” Second Annual Symposium Principles of Distributed Computing (1983), pp. 161-172.

1982

- Gottlieb, B. Lubachevsky, and L. Rudolph, “Efficient Techniques for Coordinating Cooperating Sequential Processors,” ACM Transactions on Programming Languages, 1982, pp. 164-189.

1981

- Gottlieb, B. Lubachevsky, and L. Rudolph “Coordinating Large Numbers of Cooperating Processors,” International Conference on Parallel Processing (1981), pp. 341-349.

Theses Supervised

MIT 6A (Master's Degree with Industrial Affiliation):

Perry Huang	(2009)	Binary Translation
Adam Rogal	(2008)	Predicting Machine Failures
Jacob Stultz	(2008)	Optimizing Live Virtual Machine Migrations using Page Hashes
James Rowe	(2005)	Fault Tolerant Dynamic Agent Systems
Adam Oliner	(2005)	Cooperative Checkpointing for Supercomputing Systems
Siddhartha Goyal	(2003)	A Service Discovery Framework for a Peer-to-Peer Network
Adam Meyerson	(1998)	
Luke Douglas	(1999)	
Geoff Gustafson	(1997)	

Master's Degrees Completed Within MIT

Perry Hung	2009	Varmosa: just-in-time binary translation of operating system kernels
Jacob Stultz	2008	Optimizing live virtual machine migrations using content-based page hashes
Calvin On	2007	Binary Translation of ARM ISA
Harel Williams	2007	Mobile Phone interaction with Kiosks
Ning Song	2006	Discovering User Context With Mobile Devices: Location and Time
Xiao Yu	2006	Learning Significant User Locations with GPS and GSM
Emily Yan	2006	A Mobile Phone GuiDE: a GUI Decoder and Enricher
Chris Leung	2005	Handling Ambiguous User Input on Touchscreen Kiosks
Jessica Huang	2004	Solving Bluetooth Deficiencies through Publish and Subscribe Systems
Debbie Wan	2004	Implementation for Simplifying Bluetooth Device Connection Methods
Atish Nigam	2004	Analytical Techniques for Debugging Pervasive Computing Environments
Nancy Kho	2004	Commanimation: A Speech-Controlled Animation System
Leon Orlando	2003	Handheld and Pervasive Computers
Johnathan Brunsman	2003	The application and design of the communication oriented routing networks
Glenn Eguchi	2003	Extending CORE for Real World Appliances
Hana Kim	2003	Speech controlled animation
Ang-chih (Brendan) Kao	2002	Design and Implementation of a generalized device interconnet
Amay Champaneria	2002	PADCAM: A Portable Human-Centric System for Handwriting
Arjun Naarayanswamy	2002	Real-time Visualization of Abstract Relationships Between Devices
Josh Jacobs	2002	Improving Memory Performance through Runtime Optimization
Sonia Garg	2001	The Design and Implementation of a Multi-User Interactive Public Display
David Chen	2001	Partitioned Compressed L2 Cache
Shalini Agarwal	2001	A Framework for Multi-Modal Input in a Pervasive Computing Environment
Daisy Paul	2001	Reducing Cache Pollution in Time-Shared Systems
Ilia Lisansky	2000	

MIT Master's (of PhD students)

Albert Huang	(2005)	The Use of Bluetooth in Linux and Location Aware Computing
Vinson Lee	(2002)	Instruction Set and Simulation Framework for Transactional Memory
Ed Suh	(2001)	Analytical Cache Models with Applications to Cache Partitioning

Hebrew University Master's Completed

Dror G. Feitelson	"Optical Computers"
Yaron Farber	"An Interleaved File System for a Highly Distributed MIMD Architecture"
Dan Bar Dov	"Caching Schemes in Parallel Computers"
Ofer Faigon	"Asynchronous Summing"
Benjamin F.D. Shaibe	"Performance of Cache Memory in Shared-Bus Multiprocessor Archtiectures"
Yaacov Fenster	"Detecting Parallel Access Anomalies"
Abby Rick Kehat	"Segmentation: Fast and Sloppy Pipeline Design"
Yoav Ossia	"Adaptive Algorithms: An Economy Oriented Model"
Daniel Citron	"Creating a Wider Bus Using Caching Techniques"

Doctoral Theses, Supervisor

PhD – Completed

Dror G. Feitelson (HU)	"In Support of Gang Scheduling"
Ari Rappoport (HU)	"Data Structures and Algorithms for Computer Graphics and Geometric Model ing"
Eugen Schenfeld (HU)	"A Parallel Architecture for a Digital Optical Computer"
Dror Zernik (HU)	“Holographic Displays for Debugging Parallel Programs”
Rafi Ben Ami (HU)	
Derek Chiou (MIT)	"Extending the Reach of Microprocessors: Column and Curious Caching" co-advised with Arvind
Boon Ang (MIT)	"Design and Implementation of a Multi-purpose Cluster System Network Interface Unit" co-advised with Arvind
Enoch Peserico (MIT)	Intelligent Fabrics
Fang Hui (SMA)	Network coding & Information Dispersal
Zhao, Qin (SMA)	Ameliorating the Overhead of Dynamic Optimization
Wu Wei (SMA)	Adaptive peer-to-peer networks

Exhibit B

United States Patent Application **20110162086**
Kind Code **A1**
Rogel; Lawrence S. ; et al. **June 30, 2011**

METHODS AND APPARATUS FOR SHARING, TRANSFERRING AND REMOVING PREVIOUSLY OWNED DIGITAL MEDIA

Abstract

The invention provide systems and methods for management of digital media objects, comprising first and second client digital data processors (e.g., personal (or private) computers, laptops, dedicated music devices, electronic book readers, and so forth) that are in communications coupling with one or more stores (e.g, dedicated disk drives, flash drives, cloud storage, etc.). At least one digital media object (DMO) or copy thereof is stored in one or more of those stores and is accessible by at least one of the first and second client digital data processors.

Inventors: **Rogel; Lawrence S.**; *(Brookline, MA)* **; Ossenmacher; John M.**; *(Balboa Island, CA)* **; Moffie; Micha**; *(Somerville, MA)* **; Viks; Amihai**; *(Ramat Gan, IL)*

Assignee: **INTELLISYSGROUP, INC.**
Cambridge
MA

Serial No.: **983257**

Series Code: **12**

Filed: **December 31, 2010**

Current U.S. Class: **726/28**; 726/27
Class at Publication: **726/28**; 726/27
International Class: G06F 21/24 20060101 G06F021/24

Claims