

Time-Saving Tool: Google Galvanizes Invention By UCSD Student During Summer Of Code

August 30, 2005

Doug Ramsey

When users update a file they wish to use on other computers, they must remember to copy the latest version manually to all the machines. Now, a student at UCSD has come up with a time-saving convenience that allows you to save the file on one device and have it updated automatically on other PCs, laptops, personal digital assistants or even third-generation cell phones.

James Anderson calls his solution 'transparent synchronization' - Tsync for short - and thanks to Google, Inc., he was able to complete his code this summer and launch a beta version of the software under the open-source GNU General Public License (GPL).

"My solution synchronizes multiple devices so that you never have to move data back and forth by hand," said Anderson, a second-year Ph.D. student in Computer Science and Engineering at UCSD's Jacobs School of Engineering. "And it's all updated automatically."

Tsync - pronounced *sink* - keeps a set of files consistent across many machines, even if those devices involve differing degrees of connectivity and availability. "It does so while requiring minimal effort from the user," added Anderson. "At the same time it maintains security, robustness to failure, and fast performance."

Anderson and roughly 400 other college students around the world are racing to complete software coding ahead of a September 1 deadline. That is the cutoff date imposed by Google when it launched what the search-engine giant calls the 'Summer of Code' - an ambitious effort to promote open-source software by motivating student programmers to "work on their own time to crank out elegant code that benefits us all."

"The Summer of Code was expressly designed to get the brightest minds on campus contributing code to open-source initiatives and inventing new open-source programs," said Chris DiBona, open-source programs manager for Mountain View, California-based Google. "James has certainly proven himself up to the challenge and is a credit to his university."

The company agreed to pay a stipend of \$4,500 to students age 18 and over who create innovative or useful open-source software while working for one of 40 sponsoring organizations. The \$2 million program made an exception for an elite group of 13 students - including UCSD's Anderson - who were invited to work directly with Google as the sponsoring agency.

Google allowed Anderson to continue working with his Ph.D. advisor, Jacobs School computer science and engineering professor Amin Vahdat. "James has done ground-breaking work in an area that could have major implications for developers and users of data devices," explained Vahdat. "I think Google will be impressed by the elegance of his solution, and James benefited because Google's support allowed him to do work that will eventually be incorporated into his dissertation."

Anderson's invention grew out of his own frustration with keeping documents updated across the four PCs and laptops he uses between work and home. "I found it very tedious to constantly synchronize my data among the four machines," said Anderson. "It was cumbersome to keep track of which files I had last modified on which machines, and when to synchronize them." From that personal experience, Anderson realized that keeping more than a few files consistent on more than two machines was impractical. "That gave me motivation to write a tool for my own use," he said, "and I had many friends and colleagues who expressed similar frustrations with existing tools."

Traditional synchronization tools, such as Rsync and Unison, require that the user manually synchronize any files after changing them, and those tools are designed only to synchronize a pair of devices. The Tsync model, on the other hand, allows the user to write a simple configuration file describing which directories should be synchronized, and listing one or more other hosts that are part of the Tsync group.

The design of Tsync uses peer-to-peer and overlay techniques to provide scalable and efficient mechanisms for transparently synchronizing many hosts. Tsync organizes a user's machines into an overlay network with a tree topology. The overlay network, through probing and a root fail-over protocol, ensures that each node remains connected with all other connected nodes, ensuring that all updates are eventually propagated to all computers even in the case where all of the machines are never simultaneously connected to one another.

Once all the devices are configured, whenever the user creates, modifies, or deletes files on one machine, those changes are automatically propagated to all the others. "So if the user were to add a bookmark on his or her machine at the university, it would be reflected on the laptop at home as well," explained Anderson. "Even if that laptop were powered off, then the next time the disconnected machine regains connectivity, it will automatically learn about the change and update itself."

Over the summer, Anderson implemented Tsync in C++ and Mace, an open-source language and toolkit for developing distributed systems. He tested it on clusters and over wide-area networks, and extended the basic system to perform transactional updates, conflict resolution, and improved monitoring of the status of updates.

"I have completed all the goals set forth in my proposal," claimed Anderson. "But I intend to continue working with and extending the system as I explore related problems for my research." One of those problems is helping the user avoid conflicts and making conflict resolution easy - and possibly even automated.

Anderson is a graduate student in CSE's Systems and Networking Group. He completed his Master's degree in electrical engineering and computer science at MIT in 2004, after receiving undergraduate degrees at MIT in computer science and literature. Anderson's research interests include distributed systems, peer-to-peer systems, data availability, computer networks, and operating systems.

Media Contact: Doug Ramsey, (858) 822-5825

