

# 2008-04-14 OSC NEWS: Project Ideas from the Ohio Supercomputer Center (Distributed Computing IT Journal)

## **\*\* Project Ideas from the Ohio Supercomputer Center**

### 1. **Improved scalability in pbsdcp scatter implementation**

**Mentor:** Troy Baer

**Programming Language(s):** Perl, C with MPI

**License:** GPL

[pbsdcp](#) is a distributed copy command for [PBS](#) and [TORQUE](#) batch environments that is part of OSC's [pbstools](#) package. It is used to copy files between shared directories (e.g. NFS home directories) and local storage on a set of compute nodes (e.g. /tmp). It has two modes of operation: scatter, in which files in a shared directory are copied into local file systems on each of the compute nodes; and gather, in which files in local file systems on each of compute nodes are collected into a shared directory

The scatter mode in pbsdcp is currently implemented in a rather naive fashion: for each node, it forks an rcp on the source files with a destination directory on that node's local storage. This means that the amount of data which must be transferred from the shared storage scales linearly with the number of nodes. We would like to replace that implementation with something more scalable, such as a tree-based or store-and-forward distribution scheme. Moreover, we would like this to use MPI for communication between nodes if possible, so that the high-performance Infiniband and Myrinet networks in our (and similar) clusters will be used for as much of the data transfer as possible.

### 1. **Improved scalability in all**

**Mentor:** Rick Mohr

**Programming Language(s):** C

**License:** GPL


[all](#)

is a distributed shell command built on top of rsh used by OSC and other sites. It allows commands to be run on either all or an arbitrary subset of the nodes in a cluster, either sequentially or in parallel.

The parallel mode of all currently has a scalability problem on clusters larger than about 200 nodes. Because all uses rsh and rsh wants to use privileged ports (i.e. port numbers below 1024), parallel executions of all run out of the necessary ports for node counts above 200 or so. One solution to this problem would be "chunking" or "batching"; that is, starting up at most a fixed number (say 128) of rsh

connections and then only starting more once the first few rshes have completed. (Similar logic can be seen in OSC's [parallel command processor](#).)

Alternately, a project to add some of all's features, such as its relatively simple syntax and PBS/TORQUE integration, to another distributed shell command such as [pdsh](#) would also be considered.

Posted by daskuza at [1:18 PM](#) 

<http://dcitj.blogspot.com/2008/04/google-summer-of-code.html>