Parallel Rsync

When copying a lot of files with fast disk and network IO, I have often found it more efficient to copy the files as multiple threads. Copying 4-8 sets of files at the same time can better saturate IO and usually sees a 4x or more improvement in speed of the transfer.

`rsync` is often the easiest choice for efficiently copying over lots of files, but unfortunately it doesn't have an option for parallel threads that is built in. So, here's a rather simple way to do this using `find`, `xargs`, and `rsync`.

Parallel Rsync (bash)

```
#!/bin/bash

# SETUP OPTIONS
export SRCDIR="/folder/path"
export DESTDIR="/folder2/path"
export THREADS="8"

# RSYNC DIRECTORY STRUCTURE
rsync -zr -f"+ */" -f"- *" $SRCDIR/ $DESTDIR/ \
# FOLLOWING MAYBE FASTER BUT NOT AS FLEXIBLE
# cd $SRCDIR; find . -type d -print0 | cpio -0pdm $DESTDIR/
# FIND ALL FILES AND PASS THEM TO MULTIPLE RSYNC PROCESSES
cd $SRCDIR  &&  find . ! -type d -print0 | xargs -0 -n1 -P$THREADS -I% rsync -az % $DESTDIR/%


# IF YOU WANT TO LIMIT THE IO PRIORITY,
# PREPEND THE FOLLOWING TO THE rsync & cd/find COMMANDS ABOVE:
#   ionice -c2
```

The `rsyncs` above can be extended to work through `ssh` as well. When using `rsync` over `ssh`, I've found that setting the ssh encryption type to `arcfour` is a critical option for speed.

rsync over ssh

```
rsync -zr -f"+ */" -f"- *" -e 'ssh -c arcfour' $SRCDIR/ remotehost:/$DESTDIR/ \
  && \
cd $SRCDIR  &&  find . ! -type d -print0 | xargs -0 -n1 -P$THREADS -I% rsync -az -e 'ssh -c arcfour' %
remotehost:/$DESTDIR/%
```