

# Cool tool of the month: The *ts* task spooler Datto Engineering

1

Presented By: Fred Mora –  
August 2014

# Agenda

2

- Why do you need a task spooler?
- How ts works
- Examples

# Why such a tool?

- When we run a task from the shell, it is interactive:
  - Runs as soon as we hit Enter
  - If it's too long, can be run in the background with the '&' operator  
Also, Ctrl-Z to stop a long running job, bg to background it.
- But multiple background jobs compete with each others
- Example: We have a portable hard drive containing thousand of pictures arranged in daily directories. We want to upload a selection to our online archive. Typical workflow:
  - Browse pictures, remove bad ones
  - Upload directory to archive server
  - Getting slower and slower as bandwidth gets consumed by parallel transfers

# Solution: queue the jobs

- We could write a script. Loop over a list of directories to copy perhaps.
- But while we explore the drive, we waste transfer time. So write smaller scripts?
- What we need is a queuing system: Upload only one directory at a time.
- Enter *ts*.

# The ts tool

- The ts tool is used to queue a shell command.
- Just prefix the command with ts. Example:

```
# Clean up some bad pix with lots of sunsets
ts scp -r /mnt/disk/tahiti-vac/day3/ archiveserver:~/images/tahiti-vac
# More cleanup in a different dir
ts scp -r /mnt/disk/tahiti-vac/day4/ archiveserver:~/images/tahiti-vac
# Etc. At the end, queue a notification popup
ts zenity --info --text "All done!"
```

# Installing ts

- No ubuntu package, old school install:
  - Download tarball from <http://vicerveza.homeunix.net/~viric/soft/ts/>
  - Uncompress and compile:

```
tar -zxvf ts-0.7.4.tar.gz
cd ts-0.7.4
make
sudo make install
```

# Using ts

- View the tasks status with `ts -l`
- Mistake? Remove the last spooled job with `ts -r`
- Multiple jobs can be run in parallel in a different spool runner “slot”.
- Default is 1 slot. Change it with option `-S`, e.g., `ts -S 4`
- Do `man ts` for more options.

Questions?

















