GNU Parallel



The problem

- Run the same command on lot of files or tasks
- Do it in parallel
 - But do not run them all in parallel simultaneously as it will slow down the computer
 - Run only N at the same time

Solution using GNU Parallel

- Run gzip on the files in current dir
 - parallel gzip ::: *
- Recompress .gz to .bz2
 - parallel "zcat {} | bzip2 >{.}.bz2" ::: *.gz
- GNU Parallel is OK with less quoting
 - parallel zcat {} "|" bzip2 ">"{.}.bz2 ::: *.gz

jduckles: GNU parallel looks neat!

Reading input

- Take arguments from command line parallel gzip ::: *
- Take arguments from stdin (standard input) find . -type f | parallel gzip
- Take arguments from a file parallel -a filelist gzip

jzawodn: very handy looking tool

Reading input - 2

- Reading a table. --colsep gives numbered args {#}
 cat filelist.tsv | parallel --colsep '\t' diff {1} {2} ">" {3}
- Reading from multiple files

```
parallel -a col1.txt -a col2.txt -a col3.txt
diff {1} {2} ">" {3}
parallel -a old_names -a new_names mv {1} {2}
parallel -a <(seq 6) -a <(seq 6 -1 1) echo
```

rusty_conover: Big up to GNU Parallel!

Building the command to run

- -X for multiple args on a line
 ls *.gz | parallel mv {} archive
 - Is *.gz | parallel -X mv {} archive
- With context
 - seq 1 100 | parallel -X mv {}.gz archive
- No command means run the line:
 - (echo Is; echo grep root /etc/passwd) | parallel

gubatron: GNU parallel - Awesome new tool, will be using less and less xargs

Controlling the output

```
    Keep the order
    parallel -k echo {}';' sleep {} ::: 3 2 1
```

Ungroup

parallel -u traceroute ::: foss.org.my debian.org freenetproject.org

parallel traceroute ::: foss.org.my debian.org freenetproject.org

Redirection (> and 2>) works as well. Quoted or not.

tashbarg: It's a hell of a tool, extremely versatile and useful.

Execution of the jobs

yetibear012: just wow

```
Run one job per CPU core
        parallel -j100% gzip ::: *
Run two jobs per CPU core
        parallel -j200% gzip ::: *
Run only one job at a time
        parallel -j1 gzip ::: *
Adjustable while running by using a file
         parallel -j /tmp/number_of_jobs gzip ::: *
```

Remote computers

 Run a jobs on yourserver.example.com and local computer (:)

parallel --sshlogin yourserver.example.com,: hostname';' echo {} ::: 1 2 3

 Use ~/.parallel/sshloginfile for the list of hosts parallel -S .. hostname';' echo {} ::: 1 2 3

frenzart: GNU Parallel ... awesome:)

Remote transfer

 Transfer file to remote computer. Return the result. Cleanup on remote. Except on local (:)

```
find logs/ -name '*.gz' | \
parallel --sshlogin server,server2,: \
--trc {.}.bz2 "zcat {} | bzip2 -9 >{.}.bz2"
```

tonymamacos: Sometimes, I'm just amazed what you can do with computers... Gnu Parallel

Semaphore

 Run 10 jobs in background. Block when the limit is reached until a job has finished:

```
for i in `ls *.log`; do
echo $i
sem -j10 gzip $i ";" echo done
done
sem --wait
```

gcarothers: GNU Parallel Most awesome UNIX tool EVER

Mutex



Block if one jobs is already running

seq 13 | \

parallel sem --id mymutex sed -i -e 'i{}' myfile

squeed: GNU Parallel! Where have you been all my life!?

Arguments for interactive programs

- Start emacs and load all the files cat filelist | parallel -uXj1 emacs
- Or vi:

cat filelist | parallel -uXj1 vi

theryanwalker: Holy crap! GNU Parallel is freaking awesome.

Poor man's job queue system

- Start the "daemon":
 - echo >jobqueue; tail -f jobqueue | parallel
- To submit your jobs to the queue:

echo my_command my_arg >> jobqueue

Anonymous: In my view, this is a cool tool for launching distributed password cracking, DDoS attacks etc. Try it out today!

Dir monitor job queue

- Task: Process all new files put into a dir inotifywait -q -m -r -e CLOSE_WRITE \ --format %w%f my_dir | \
- To submit your jobs just save the file to my dir.
- To distribute processing use -S:

parallel -u echo

parallel -S .. -u echo

taximer: コマンドラインから並列処理させる GNU Parallel

--pipe

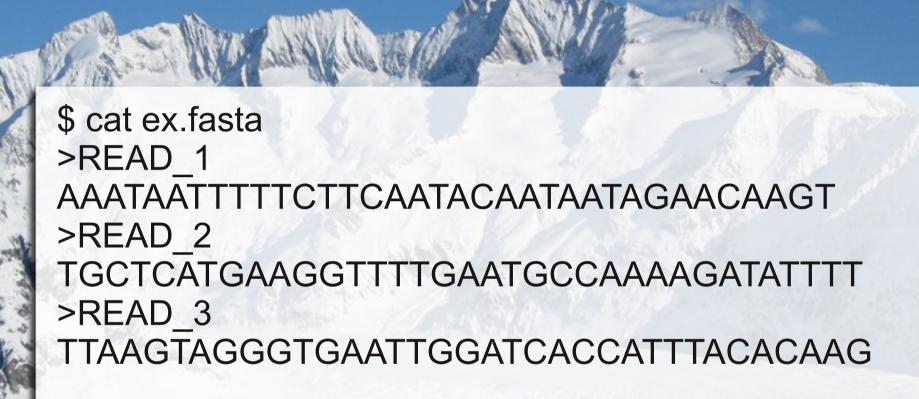


- --pipe pipes a block of records to the same command in parallel
- --recstart and --recend

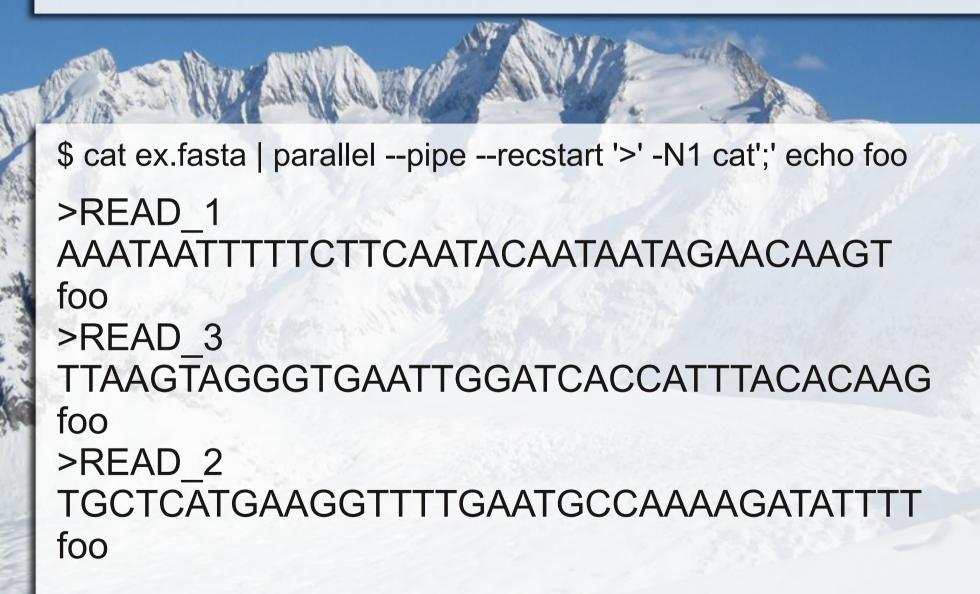
--recend

```
$ seq 1 5 | parallel --pipe --recend '\n' -N1 cat';' echo foo
foo
foo
5
foo
3
foo
4
foo
```

--recstart - 1



--recstart – 2



Both --recstart --recend

\$ cat /var/spool/mail/parallel | parallel --pipe --recend '\n\n' --recstart 'From ' -N1 cat';' echo foo

From root@alpha.tange.dk Mon Apr 23 10:20:38 2007 [...headers...]

The body of the email

foo

From tange@alpha.tange.dk Mon Apr 23 14:52:18 2007 Return-path: <tange@alpha.tange.dk>

Number of records

```
$ seq 1 5 | parallel --pipe -N3 cat';' echo foo
foo
4
5
foo
```

Blocksize - 1

```
$ cat /usr/share/dict/words | parallel --pipe --blocksize 500k wc
 39214 39214 511994
 45341 45341 511994
 38227 38227 512001
 41070 41070 512011
 41514 41514 512000
 24773 24773 314775
 40289 40289 511991
 39643 39643 511997
```

Blocksize – 2

\$ cat /usr/share/dict/words | parallel --pipe wc 81800 81800 1048572

86196 86196 1048570

83538 83538 1048584

58537 58537 753037

Just a bunch of bytes (no records)

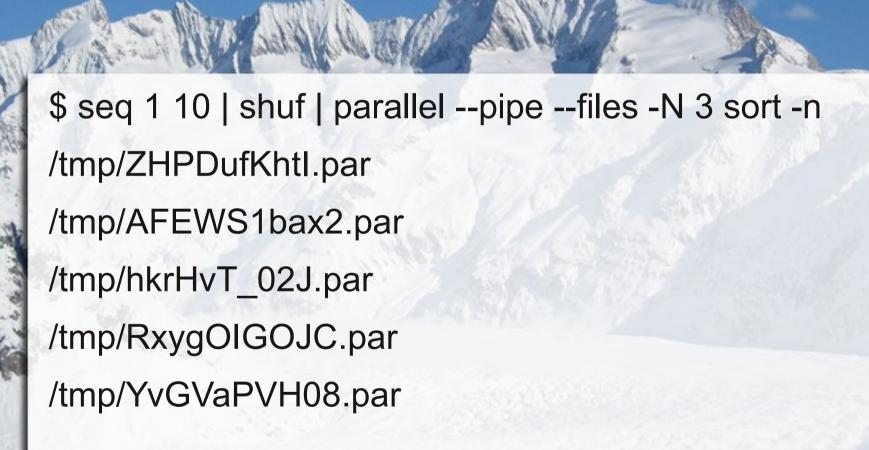
cat lucene.tar | parallel --pipe --recend " -k gzip > lucene.tar.gz

gzip magic: append two gzip-files and get one.

Output as files - 1

```
$ seq 1 10 | shuf | parallel --pipe -N 3 sort -n
49
5
6
```

Output as files – 2



Output as files – 3

```
$ seq 1 10 | shuf | parallel --pipe --files -N 3 sort -n |
  parallel -mj1 sort -nm
5
6
8
```

Output as files – 4

```
$ seq 1 10 | shuf | parallel --pipe --files -N 3 sort -n |
  parallel -mj1 sort -nm {} ";"rm {}
5
6
8
```

How do you feel?



- Any questions?
- Sign my GnuPG key

 Watch the YouTube video if you missed something: Search for "GNU Parallel"

karma4cowards: I feel like the guy in the Intel commercial being shown wireless for the first time. Is this the coolest shit ever