

mfork

Simple command parallelization

Problem

Regularly sync 3 or 5 gigs around the world to 6 sites

Changes?	rdist	serial rsync	parallel rsync
Yes	25 hours	31 mins	4 mins
No	?	2.5 mins	38 secs

Serial rsync

```
for f in foo/bar bar/baz baa; do  
  rsync -a /$SRC/$f/ news:/$DST/$f  
done
```

```
rsync -a /src/foo/bar/ news:/dst/foo/bar  
rsync -a /src/bar/baz/ news:/dst/bar/baz  
rsync -a /src/baa/ news:/dst/baa
```

Parallelize rsync

```
for f in foo/bar bar/baz baa; do  
  echo rsync -a /$SRC/$f/ news:/$DST/$f  
done
```

```
rsync -a /src/foo/bar/ news:/dst/foo/bar  
rsync -a /src/bar/baz/ news:/dst/bar/baz  
rsync -a /src/baa/ news:/dst/baa
```

Parallelize rsync

```
for f in foo/bar bar/baz baa; do  
  echo rsync -a /$SRC/$f/ news:/$DST/$f  
done | mfork
```

Why not use make?

```
SUBDIRS = foo/bar bar/baz baa
```

```
.PHONY: subdirs $(SUBDIRS)
```

```
subdirs: $(SUBDIRS)
```

```
$(SUBDIRS):
```

```
    rsync -a /$(SRC)/$@/ news:/$(DST)/$@
```

Why not use make?

- Need to learn quirky make tricks
- Need GNU make
- Difficult to pipe commands into

More examples

```
$ for f in homer no work beer well without; do  
    echo "echo -n '$f ' "  
done | ./mfork ; echo ""  
homer no work without well beer
```


Recursive calls

```
i=1;
while [ $i -lt 100 ]; do
    echo 'for f in homer no work beer well without; do
        echo echo -n \"$f \"; done | ./mfork; echo "" ';
    i=`expr $i + 1`;
done | ./mfork
```

Wrong!

beer homer no well without work homer homer beer
beer no no well beer well homer without well without
work no work without work no work
beer
homer homer no homer well work without without
homer

Save results!

```
i=1; while [ $i -lt 100 ]; do
  echo 'S=`for f in homer no work beer well without; do
    echo echo -n \"$f \";
  done | ./mfork; echo ""`;
  echo $S';
  i=`expr $i + 1`;
done | ./mfork
```

no beer work well without homer
no work homer without well beer
homer no work without well beer
beer well homer work without no
work homer without no beer well
work no homer well without beer
beer without well no homer work
work well beer no homer without
without beer homer well no work
without work homer no beer well
homer no work beer without well
no work homer without beer well
work without homer beer well no

Options

- c continue on error (default: stop forking)
- d debug output
- h help text
- i maxn set maximum number of instances (default: 10)
- q quiet mode
- v verbose output

Future

- # of forks = # of CPUs/cores/threads ?
- \ line continuations
- better signal handling
- let me know!

Help

- Sourceforge: <http://sourceforge.net/projects/mfork>
- <mailto:maarten@cisco.com>
- <http://employees.org/~mthibaut/LISA07/index.html>

no beer work well without homer
no work homer without well beer
homer no work without well beer
beer well homer work without no
work homer without no beer well
work no homer well without beer
beer without well no homer work
work well beer no homer without
without beer homer well no work
without work homer no beer well
homer no work beer without well
no work homer without beer well
work without homer beer well no