

Poly

rapid development of
parallelizable applications

Michael M. Hoffman

EMBL-European Bioinformatics Institute
and University of Cambridge

22 July 2004

What does Poly allow you to do?

- ◆ Program for one processor, run on one or hundreds
- ◆ Still use a simple stdin/stdout input/output model if you want
- ◆ Worry less about some of the hassles of distributed computing
- ◆ Avoid crashing your server farm

A short Python example

```
print "filename0 filename1 identity_fraction"

for pair in fasta_pairs():
    a = dna1.align(pair)
    identity_fraction = (a.matches /
                        (a.matches+a.mismatches))
    print pair[0], pair[1], identity_fraction
```

```
filename0 filename1 identity_fraction
00.fasta 01.fasta 0.595054095827
00.fasta 02.fasta 0.548993288591
...
```

The same example, Polyfied

```
if poly.firstjob:
    print "filename0 filename1 identity_fraction"

for pair in poly.chunk(fasta_pairs()):
    a = dna1.align(pair)
    identity_fraction = (a.matches /
                        (a.matches+a.mismatches))
    print pair[0], pair[1], identity_fraction
```

Running the example

◆ Standalone

```
$ alignall.py
```

◆ Platform LSF

```
$ polysub -J "alignall[1-99]" -- alignall.py
```

◆ Other systems?

```
filename0 filename1 identity_fraction  
00.fasta 01.fasta 0.595054095827  
00.fasta 02.fasta 0.548993288591  
...
```

Convenience features

- ◆ Gradual job throttling
- ◆ Makes local copies of data on NFS using a load-balanced, throttled copy program
- ◆ Cleans up these and other temporary files

Shell interface

◆ Similar to xargs

```
$ find . *.fasta | polyxargs -J "x[1-99]" -- \
    RepeatMasker
```

Availability

<http://www.ebi.ac.uk/~hoffman/software/poly/>

License: GNU GPL

Acknowledgements

Ewan Birney

Tim Cutts

Guy Coates

Marshall Aid Commemoration Commission