Illuminating OpenMP + MPI Performance

Beau Paisley, bpaisley@allinea.com
HPC means being able to work productively on remote machines

![Image of a software interface](image-url)
Calculating pi with 1 thread, 1 process

```
bpaisley@beau:~/ownCloud/bpaisley/src/pi$ time cpi
initializing...
calculating pi...
pi is approximately 3.1415926535897416, Error is 0.0000000000000515
writing to disk...
finished

real    1m15.971s
user    1m13.306s
sys     0m0.236s
bpaisley@beau:~/ownCloud/bpaisley/src/pi$
```
Profile with 1 process, 1 thread

```
bpaisley@beau:~/ownCloud/bpaisley/src/pl
bpaisley@beau:~/ownCloud/bpaisley/src/pl$ map --profile .\cpl
Allinea Forge 5.0.1 - Allinea MAP
Profiling    : /home/bpaisley/ownCloud/bpaisley/src/pl/cpl
Allinea sampler : preload
MPI enabled  : No

initializing...
calculating pi...
pi is approximately 3.1415926535897416, Error is 0.00000000000000515
writing to disk...
finished

MAP analysing program...
MAP gathering samples...
MAP generated /home/bpaisley/ownCloud/bpaisley/src/pl/cpl_1p_2015-04-28_15-47.map
bpaisley@beau:~/ownCloud/bpaisley/src/pl$```
Using MAP to analyze our 1 process, 1 thread run
Zooming in on the IO portion of our run

```c
for(i=0; i<N; ++i)
{
    f = fopen("cpi.tmp", "w");
    fwrite(buffer, len, sizeof(char), f);
    fclose(f);
}
unlink("cpi.tmp");
```
Always be thinking about vectorization
While still connected to the server we can switch to the debugger.
Adding some OpenMP pragmas to multithread
Adding MPI support and submitting to an HPC batch system
Analyzing our OpenMP + MPI results
Why all this synchronization?
The workflow of pi from serial to HPC