
Programming Assignment 1

Department of Electrical Engineering
Stanford University

<http://eeclass.stanford.edu/ee282>

Programming Assignment 1

- Handout available on the website
 - Contains much of what we'll go over today
- What are you going to be doing?
 - Optimize matrix multiply
 - Write a short (3-4 page) report
 - Performance measurements
 - Qualitative description of your optimizations
 - Due October 29th @ 5PM PST
- Groups of 3. Contact Jason/Dawson if you are still looking for partners!

Logging into the Cyclades Cluster

- Do this soon to make sure your account works
- SSH into **soda.stanford.edu**
 - Not on the public network.
 - If you work off-campus, first log into a Leland node (myth, elaine, bramble, etc.) with your SUNet ID, or use the VPN
 - <https://www.stanford.edu/services/vpn/>
- SSH clients for Windows users
 - <http://www.stanford.edu/dept/its/support/ess/pc/apps/sCRT52inst.exe>
 - <http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Logging into the Cyclades Cluster (cont)

- SSH into **soda.stanford.edu** (also known as hotbox-1)
 - Username: SUNet ID
 - Password: SUNet Password

Submitting Jobs

- ```
% uptime
23:19:32 up 119 days, 9:55, 72 users, load average: 0.18, 0.21, 0.22
```
- This is a busy cluster. Don't run intensive jobs on **soda**, dispatch them to a compute node.
    - Corollary: Let me know if you see users abusing it
  - TORQUE resource manager, MAUI cluster scheduler (if you care)
  - “**jsub -- command**” submits jobs
    - By default, runs the job *asynchronously*
    - “**-I**” flag tells jsub to watch the job synchronously
    - The “**--**” is required...

## jsub example #1

- ```
% jsub -- hostname
140288.cyclades-master.tendot.stanford.edu
%
```
- Returns immediately. The output is your job's “name”.
 - On completion, creates 2 files, where XXXXX is the job name:
 - **<command>.oXXXXX**
 - **<command>.eXXXXX**
- ```
% cat hostname.o140288
cyclades-5.Stanford.EDU
```

## jsub example #2

- Often useful to synchronously wait for the job, especially short jobs

```
% jsub -I -- hostname
qsub: waiting for job 140292.cyclades-master.stanford.edu to start
qsub: job 140292.cyclades-master.stanford.edu ready

cyclades-5.Stanford.EDU

qsub: job 140292.cyclades-master.stanford.edu completed
%
```

- Doesn't create the output files

## Getting Started

- Log into **soda**
- Download <http://ee282.stanford.edu/pa1.tar.gz> using **wget**

```
% wget http://ee282.stanford.edu/pa1.tar.gz
--23:51:19-- http://ee282.stanford.edu/pa1.tar.gz
=> 'pa1.tar.gz'
Resolving ee282.stanford.edu... 171.67.22.49
Connecting to ee282.stanford.edu|171.67.22.49|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3,786 (3.7K) [application/x-tar]

100%[=====] 3,786 ---K/s

23:51:24 (180.53 MB/s) - 'pa1.tar.gz' saved [3786/3786]
```

- Unpack it into your home directory

```
% tar zxvf pa1.tar.gz
pa1/
pa1/driver.c
pa1/Makefile
pa1/matmul.c
pa1/utills.c
pa1/utills.h
```

## Compiling

- Just type “make”

```
% cd pal
% make
gcc -O3 -DPAPI -c -o driver.o driver.c
gcc -O3 -DPAPI -c -o matmul.o matmul.c
gcc -O3 -DPAPI -c -o utils.o utils.c
gcc -O3 -DPAPI -lperfctr -lpapi driver.o matmul.o
utils.o -o matmul
```

- Edit the Makefile to change the compiler to use and what flags to pass to it.

## Running matmul on the cluster

- Don't run matmul on **cyclades-master**, submit to cluster

```
$ jsub -I -- ./matmul
qsub: waiting for job 140294.cyclades-master.stanford.edu to start
qsub: job 140294.cyclades-master.stanford.edu ready
```

Each measurement is average per iteration. Runtime is given in milliseconds.  
MFLOPS is estimated assuming a naive matmul().

| Dim. | MFLOPS  | Runtime   | Tot. Instr. | Tot. Cycles | L1 D-Misses | L2 D-Misses |
|------|---------|-----------|-------------|-------------|-------------|-------------|
| 2    | 393.762 | 0.0000    | 163         | 73          | 0           | 0           |
| 4    | 343.697 | 0.0004    | 905         | 664         | 0           | 0           |
| 8    | 385.434 | 0.0027    | 6301        | 4746        | 0           | 0           |
| 16   | 256.883 | 0.0319    | 47525       | 56926       | 0           | 0           |
| 32   | 273.346 | 0.2398    | 369974      | 428052      | 1           | 0           |
| 64   | 280.860 | 1.8667    | 2921048     | 3359393     | 3376        | 0           |
| 128  | 189.171 | 22.1721   | 23217328    | 39971880    | 2133749     | 779         |
| 256  | 170.931 | 196.3035  | 185141743   | 353638713   | 16918626    | 33157       |
| 512  | 66.031  | 4065.2820 | 1478765072  | 7289412032  | 134582570   | 134486470   |

```
qsub: job 140294.cyclades-master.stanford.edu completed
```

## Performance Counters

- Run “papi\_avail” and “papi\_native\_avail”
- Edit “events[]” in driver.c
- Example: PAPI\_L1\_DCM

| Dim. | MFLOPS  | Runtime    | L1 DCM     | Tot. Cycles  |
|------|---------|------------|------------|--------------|
| 2    | 156.641 | 0.0001     | 0          | 237          |
| 4    | 184.022 | 0.0007     | 0          | 1616         |
| 8    | 192.903 | 0.0053     | 0          | 12330        |
| 16   | 192.600 | 0.0425     | 0          | 98736        |
| 32   | 197.671 | 0.3315     | 27         | 770563       |
| 64   | 191.257 | 2.7413     | 41255      | 6371170      |
| 128  | 167.211 | 25.0839    | 2080498    | 58257602     |
| 256  | 156.333 | 214.6341   | 16736654   | 498615042    |
| 512  | 145.122 | 1849.7188  | 134932400  | 4298516727   |
| 1024 | 43.794  | 49036.5450 | 1088940602 | 113979676129 |

- What size L1 cache does this machine have?

## Example: Change loop order

```
% emacs matmul.c
```

### Original matmul ()

```
void matmul (int N, const double* A, const double* B, double* C) {
 int i, j, k;

 for (i = 0; i < N; i++) {
 for (j = 0; j < N; j++) {
 for (k = 0; k < N; k++) {
 C[i*N + j] += A[i*N + k] * B[k*N + j];
 }
 }
 }
}
```

### Reordered matmul ()

```
void matmul (int N, const double* A, const double* B, double* C) {
 int i, j, k;

 for (i = 0; i < N; i++) {
 for (k = 0; k < N; k++) {
 for (j = 0; j < N; j++) {
 C[i*N + j] += A[i*N + k] * B[k*N + j];
 }
 }
 }
}
```

## Example: Change loop order (cont.)

```
% make
% jsub -I -- ./matmul
```

| Dim. | MFLOPS  | Runtime  | Tot. Instr. | Tot. Cycles | L1 D-Misses | L2 D-Misses |
|------|---------|----------|-------------|-------------|-------------|-------------|
| 2    | 450.519 | 0.0000   | 167         | 64          | 0           | 0           |
| 4    | 529.185 | 0.0002   | 921         | 436         | 0           | 0           |
| 8    | 740.889 | 0.0014   | 6365        | 2489        | 0           | 0           |
| 16   | 613.839 | 0.0133   | 47781       | 23981       | 0           | 0           |
| 32   | 696.654 | 0.0941   | 370997      | 169378      | 0           | 0           |
| 64   | 779.573 | 0.6725   | 2925142     | 1207955     | 1344        | 0           |
| 128  | 569.314 | 7.3673   | 23233698    | 13299151    | 266380      | 0           |
| 256  | 578.613 | 57.9911  | 185207140   | 104502846   | 2113877     | 1064        |
| 512  | 410.326 | 654.2006 | 1479021923  | 1173806219  | 16844796    | 1093520     |

Vs.

| Dim. | MFLOPS  | Runtime   | Tot. Instr. | Tot. Cycles | L1 D-Misses | L2 D-Misses |
|------|---------|-----------|-------------|-------------|-------------|-------------|
| 128  | 189.171 | 22.1721   | 23217328    | 39971880    | 2133749     | 779         |
| 256  | 170.931 | 196.3035  | 185141743   | 353638713   | 16918626    | 33157       |
| 512  | 66.031  | 4065.2820 | 1478765072  | 7289412032  | 134582570   | 134486470   |

## Performance Counters

- Run “`papi_avail`”
- Edit “`events[]`” in `driver.c`
- Example: `PAPI_FP_OPS`

| Dim. | MFLOPS  | Runtime | FPU Instr. | Tot. Cycles | L1 D-Misses | L2 D-Misses |
|------|---------|---------|------------|-------------|-------------|-------------|
| 128  | 187.981 | 22.3124 | 4209242    | 39819708    | 2133725     | 48          |

$$- 4209242 / 39819708 * 1800 = 190.273 \text{ MFLOPS}$$

- What’s the resolution of Runtime?
  - $4209242 / 23 / 10^3 = 183.010 \text{ MFLOPS}$
  - $4209242 / 22.3124 / 10^3 = 188.650 \text{ MFLOPS}$
  - $4209242 / 21 / 10^3 = 200.440 \text{ MFLOPS}$

## CVS (Concurrent Versions System)

- You are going to need to copy your files back and forth between Cyclades and Leland.
- One option is to use `scp`
  - Copy from `soda` to `myth`  
`soda% scp -r pal myth:`
  - Copy from `myth` to `soda`  
`soda% scp -r myth:pal .`
  - Be careful!

## CVS (cont.)

- Second option is to use CVS
  - <http://ximbiot.com/cvs/manual/>
- Brief instructions:
  - Put this in your Leland `.cshrc`:  
`setenv CVSROOT $HOME/CVSROOT`
  - Log out and back in, then

```
% cvs init
% wget http://ee282.stanford.edu/pal.tar.gz
% tar xzvf pal.tar.gz
% cd pal
% cvs import pal x y
% cd ..
% cvs co pal
```

## CVS (cont.)

---

- Now let's check-out that CVS module on **hotbox-1**
- Put this in your **hotbox-1** `.bashrc`:

```
export CVS_RSH="ssh"
export CVSROOT=":ext:leverich@myth:/afs/ir/users/l/e/leverich/CVSROOT"
```

- You will need to adjust the login name (i.e. not leverich) and the path to CVSROOT (type "pwd" on Leland).

- Log out and log back in, then:

```
% cvs co pal
Password: [leland password]
cvs checkout: Updating pal
U pal/Makefile
U pal/driver.c
U pal/matmul.c
U pal/utils.c
U pal/utils.h
```

## Using CVS

---

```
% cd pal
% cvs up
% emacs matmul.c
% cvs commit
```

- “cvs up” updates your local copy from the repository
- “cvs commit” sends any local updates to the repository

- You can even use this to collaborate with your group members.