

## Contents

<b>1</b>	<b>RECAP OF WORK YOU HAVE TO DO!!</b>	<b>1</b>
1.1	MOST IMPORTANT . . . . .	1
1.2	ASSIGNMENT 1 . . . . .	2
<b>2</b>	<b>RECAP OF WHAT WE DISCUSSED DURING L1</b>	<b>2</b>
2.1	Basics . . . . .	2
2.1.1	Power dissipation of a chip . . . . .	2
2.1.2	Counting the number of interleavings . . . . .	3
2.1.3	How can a spider rock a locomotive one inch to/fro every second? . . . . .	3
2.2	Amdahl's law in the multi-core era . . . . .	4
<b>3</b>	<b>LECTURE 2</b>	<b>4</b>

## 1 RECAP OF WORK YOU HAVE TO DO!!

### 1.1 MOST IMPORTANT

1. Make sure that ALL OF YOU you belong to `cs5966@eng.utah.edu`.
2. IN ADDITION, please also send `sriram@cs.utah.edu` your preferred email address.
3. Please join the Google group `cs5966`. The TA will initiate an invitation.
4. Please test that you can access the class wiki.
5. Please fill and mail the class survey electronically to `teach-cs5966` at `eng.utah.edu`.  
The file is called `SURVEY` and is in the class lectures notes for `Week1`.

First, decide if you really belong in this class. I like you and I want to teach you! But I don't want to disappoint you!

- If you have not taken an UG-level Operating Systems class or a basic computer hardware class, consider taking this class ONLY if you can study a ton on your own.
- Ideally you should also have taken a compiler class - however, I will assume that you have familiarity with compilers and how they work.
- Most of you are either seniors or MS students - which is ideal. One or two are PhD students or juniors - that is great.

- In my assignments, I'll put a (\*) against some questions. Even if you don't have sufficient background, you must make a modest attempt at these questions (with my help). However, I'll adjust the grading to fit your background.
- If you are an advanced student: try to contribute to my lectures through discussions, and do a more ambitious project.

## 1.2 ASSIGNMENT 1

The problems are clearly described on the class webpages. They all consist of discussions you have to do on the Google group. Ask me if anything is unclear.

I would like your *extremely enthusiastic* participation in class as well as discussions. I'll be allocating the 15% of discussion points based on the 'enthu level' I observe.

Also the assignments are variable in difficult. Some will clearly need a ton of time. Please begin early so that we can work out difficulties. An ideal approach is: get going Monday evening, ask questions in class on Wed. I'll strive to assign all assignments on Monday.

I would like to add another problem to get you going on parallel computing. Compile and run the program `arl.c` on `nabokov.eng.utah.edu` (an 8-way SMP, quad dual-Opteron) or other machine, as I shall show in class. Here are the compilation commands:

```
gcc -lpthread -lrt arl8.c -o arl8
```

```
arl8 && arl8 && arl8 && arl8 && arl8 && arl8 && arl8 arl8
```

(the above runs the command 8 times..)

(test for different parameters and have a Google email discussion on your findings)

You can find an excellent Pthread tutorial at

<https://computing.llnl.gov/tutorials/threads/>

I have kept a ton of examples (including `arl8.c`) at

<http://www.eng.utah.edu/~cs5966/LECTURES/Week1/pthread-exs/>

## 2 RECAP OF WHAT WE DISCUSSED DURING L1

### 2.1 Basics

#### 2.1.1 Power dissipation of a chip

In Watts, a microprocessor chip dissipates anywhere from 1W to 100W (more or less is easily possible).

The bulk of the power is dissipated through charging and discharging of the node capacitances. The formula to use is  $ACV^2f$  where  $A$  is the activity factor,  $C$  the capacitance,  $V$  the voltage, and  $f$  the frequency.

Assuming 0.0001pF per node, 1 volt, 1B nodes, and 3GHz frequency, and  $A = 0.3$ , we get  $10^9 \times 0.1 \times 10^{-12} \times 1^2 \times 3 \times 10^9 \times 0.3 = 100W!$  BTW an external node to a chip can have 1pF capacitance (see “spider” below)!

In modern chips, a lot of the power is wasted through leakage also. This is relatively constant, but highly temperature dependent (can have “runaway leakage” also).

What’s wrong with dissipating energy? First, it trashes the world (10% of nation’s electricity towards computing, as estimated). Second, it jacks up packaging cost which is a huge percentage of chip costs. Third, you need a fan (noise), reduce battery life, etc., all of which reduces product market.

What are some typical numbers? Anything beyond 100W for a desktop is insane! Computing the power of a supercomputer: 100K CPU units, at 30W each = 3MW. One kWh costs 10 cents? Thus 3MWh is \$30. For 6000 hours of operation a year, that comes \$180K.

### 2.1.2 Counting the number of interleavings

If there are  $n$  threads with  $k$  instructions each, the total possible interleavings are

$$(nk)! / (k!)^n.$$

This grows fast. For  $n = k = 5$ , we get over  $10^{10}$ .

As another test of understanding, calculate the interleavings in this actual MPI program run by two threads. (Don’t worry if you don’t follow all details - I’ll tell you what you need to know.)

Process 1	Process 2
0: MPI_Init	0: MPI_Init
1: MPI_Win_lock	1: MPI_Win_lock
2: MPI_Accumulate	2: MPI_Accumulate
3: MPI_Win_unlock	3: MPI_Win_unlock
4: MPI_Barrier	4: MPI_Barrier
5: MPI_Finalize	5: MPI_Finalize

There are 504 interleavings. How did we get it?

Well, there is a slight twist here!

The MPI\_Barrier is an alignment point. So calculate the number of interleavings till the barrier is reached. This is indeed given by “our friend”  $(10! / (5!)^2)$ .

Thereafter there are only two ways in which MPI\_Finalize can be called - either process 1 calls it first, or process 2 does so.

Thus we get  $2 \times (10! / (5!)^2)$ .

Clearly we cannot get this calculation done always!

Also the number of interleavings alone do not matter. WHAT each step does also matters.

For example, in the following program, **more than just the interleavings matter!**

thread 1	thread 2
0: x++	0: (x = (x : x++ : x--));

```

1: 0: (y = (y : y++ : y--));           1: x++
2: x++                                 2: (x = (x : x++ : x--));
3: 0: (y = (y : y++ : y--));           3: x++

```

### 2.1.3 How can a spider rock a locomotive one inch to/fro every second?

Unlike what Archimedes said, clearly not using a giant crowbar, because that conserves energy (so to increase mechanical advantage you need to reduce velocity ratio). However, you can hook the spider leg thru a cob-web to a light modulator (a black thru gray to white color film), hook a phototransistor, then an amplifier chain, finally a giant servo – yes you can!

This is how a transistor within a chip feels like when it contemplates the prospects of driving the chip pad !! Moral: going off-chip is EXPENSIVE!

## 2.2 Amdahl’s law in the multi-core era

Hope you got the gist. Ask if any questions.

Dynamic multicores: people asked “how many cores”. That depends on  $R$ .

The parallel time goes as

$$1/(\text{sqrt}(R).(N/R))$$

Thus, setting  $R = 1$  gives the best speedup.

## 3 LECTURE 2

Today’s lecture will go through Engblom’s very nice paper and examine the issues he raises.