Introduction to Computer Architecture

Kaushik Patra
(kaushik.patra@sjsu.edu)

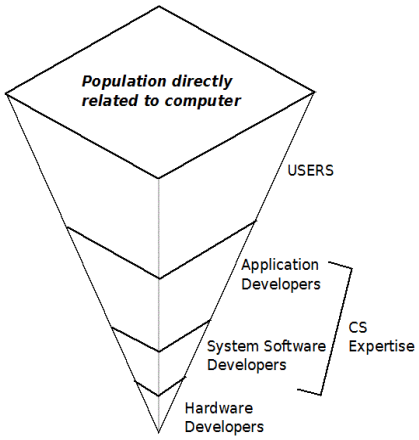
CS147 Objective

- To understand computer organization and architecture.
- To understand how the abstract concept of architecture are implemented.
- To understand handshake between hardware and software.
- To review some advanced architecture for performance optimization.

2

- The subject of computer architecture does not differ much from the classical 'Architecture' subject. The traditional architecture subject conceptualize how to fill up a given space with architectural objects (such as buildings, bridges, etc.) of need. Beautification is one of the important aspect of classical architecture.
- Computer architecture conceptualize how to put together digital architectural objects (CPU, Memory, IOs, etc) to build a machine which can be programed to solve mathematical problems (Anything software we build can be mapped into a mathematical problem). Unlike classical architecture, performance and capacity are the most important aspect for computer architecture.

Why Computer Architecture?



- It is an inverted pyramid – less number of people knows the base and larger number knows the top.
- As CS personnel, one will probably land in application development area or system software area.
- Why would someone in CS care about hardware?

3

- Hardware is the base of any computation and application platform. Only few people have knowledge in this area. As we go up the abstraction level, number of people knowledgeable in the that level grows. Hence representation results in an inverted pyramid.
- The diagram shows a coarse grain classification. There can be many subdivision within and also intermingled professions. [e.g. Electronic Design Automation (EDA) is a application development profession to automate hardware design and implementation – this area needs both CS and CMPE/EE background]

Why Computer Architecture?

- We have a program to multiply integers by 4 in an array. An implementation is a following. Can we improve the performance?

```
boolean cs147_multiply_by_2(float_array arr) {
        for(int i=0; i<arr.size(); i++) {
                arr[i] = arr[i] * 4.0;
        }
        re
}
```

```
boolean cs147_multiply_by_2(int_array arr) {
        for(int i=0; i<arr.size(); i++) {
                arr[i] = arr[i] * 4;
        }
        retu
}
```

```
boolean cs147_multiply_by_2(int_array arr) {
        for(int i=0; i<arr.size(); i++) {
                arr[i] = arr[i] << 2;
        }
        return true;
}
```

4

- Assigning float type for integers is wastage of space. For a 32 bit machine, floating point can represent integers only by 24 bit, where as integer type data can use all 32 bits.
- Floating point operation is much slower than integer operation.
- We can use left shift operation for multiplying any integer by 4, instead of multiplication, which is faster than multiplication.

Why Computer Architecture?

- Software development without knowledge of hardware is similar to driving a car with manual transmission capability in 'cruise control' mode.
- Programs can be tuned for faster performance with smaller memory footprint and minimized execution error with prior knowledge of the target hardware.
- Debugging of a complex software often requires knowledge of target hardware.
- Often, business decision needs prior knowledge of target hardware.

5

- Operating system, compilers, etc. present auto mode of program optimization for run time and memory. A badly implemented algorithm can be executed with fairly acceptable performance level.
- Understanding of underlying hardware can be used to implement faster execution.
  - (a << 2) gives faster performance than (a*4)
  - Multiple Boolean flags can be implemented using individual bits of single integer to compact memory usage.
  - Use of 3D graphic accelerator in math computation.
- Software powerhouses implements platform specific code to gain highest performance on targeted hardware platform.
- Debugging, especially for parallel programming, needs deeper understanding of the target hardware platform.
- Buy or build decision for target hardware platform for large organization depends on the type of computation needed for the organization.

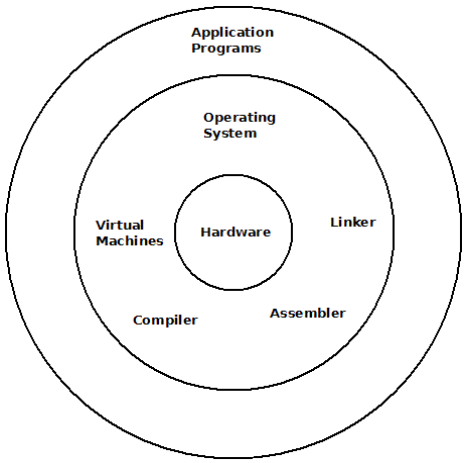
Why Computer Architecture?

- As a subject, understanding of hardware needs background in electrical engineering.
  - For CS study, which can be bit overwhelming and unnecessary.
- However, it is possible abstract out details of the hardware components and operations, which defines:
  - Interface and operational characteristics of individual components.
  - Integration of components into overall system.
  - Implication to program implementation.

Why Computer Architecture?

- Essentially architectural definition of a hardware platform is conceptually equivalent to 'blueprint' used in other engineering field.
- Computer architecture is essentially a handshake between software developers and hardware developers.
- Since computer architecture is a common understanding between software and hardware developers on the target computing platform, it is very much important for a CS personnel to have concept of such common understanding.

What to expect?



- As subject, computer science is related directly to application program space and system software space.
- When we write program to solve target mathematical problem, we are cushioned by system software from complicated nature of underlying hardware platform.
- In this course, these cushions will be removed and we'll be exposed to the intricate details of the hardware.

8

What to expect?

- We'll have more hands-on approach to understand this subject in details.
  - Lab activities and project assignments are very important.
- We'll review basic architecture components and integration.
- We'll review some advanced techniques in computer architecture.

The Green Sheet

- The green sheet is available on Canvas system.
- Let's review it …

Introduction to Computer Architecture

Kaushik Patra
(kaushik.patra@sjsu.edu)

