Ôn tập NT533 - Hệ tính toán phân bố

Week 1 + 2: Course Introduction, Fundamentals

Some keywords from the 5th Computer System Generation

- *The network is the computer*
- Distributed system: **Cluster, Cloud, Grid, P2P-Computing**
- Multicore processors and **parallel applications**
- Virtualization: **VMware, XEN, KVM (Kernel-based Virtual Machine), Docker**
- Open Source: **Linux**, BSD

Client - Server

A system includes

- One or more **client** which use the services of the server and accesses data, stored at the server (⇒ consumers)
- A **Server** which provides services and/or data (⇒ producer)

The connection establishment is initiated by the clients

Communication works according to a protocol

- A client sends a request to the server
- The server responds with a reply

The client-server architecture consists of 2 layers and is called two-tier model (tier = layer)

Tasks in Client - Server Model

- Display (graphical) user interface
- Calculation of the (graphical) user interface
- Data processing
- Data management
- Data storage

Types of Clients

- Text-Terminals, X-Terminals
  - Only display the (graphical) user interface and transfer the user interaction to the server
  - Calculation of the (graphical) user interface, data processing and data storage, data management are tasks of the server
- Thin/Zero Clients
  - Calculate and display the graphical user interface
- Applet Clients
  - Calculate and display the graphical user interface and do a part of the data processing
  - The clients process the applications (applets) themselves
- Fat Clients
  - Only data management and data storage are located on the (file or database) server

Advantages of Thin Clients

- Low acquisition cost
- Reduced power consumption => reduced operating costs
- Little space consumption
- Reduced noise
- Central storage of data is more efficient and more secure
- Virtualization on the server
- Reduced effort for administration

Drawback of Thin Clients

- No 3D graphics performance
- Limited extensibility
- Users fear storing their data outside of their PC
- Server is a single point of failure and eventually a bottleneck

Moore's Law

Published in 1965 by Gordon Moore

It is based of empirical observation

Moore originally meant the electronic components on of integrated circuit double every 12 months

Since the late 1970s, the packing density only doubles every 24 months

Von Neumann Bottleneck

The data and control bus is increasingly becoming a bottleneck between the CPU and memory

The main memory and the bus system are key factors for the performance of a computer

The Von Neumann Architecture describes the structure of the general-purpose computer, which is not limited to a fixed program and has input and output devices

Main difference to modern systems: A single Bus to connect I/O devices directly with the CPU, is impossible today

Main memory is usually DRAM (Dynamic Random Access Memory)

The access time (or cycle time)

- DDR-400 SDRAM is 5 ns: 200 MHz
- DDR3-2400 SDRAM is 0.833 ns: 1200 MHz
- DDR4-4800 SDRAM is 0.417 ns: 2400

> *Note: 1 Hz =* $\frac{1}{seconds}$

To reduce bottleneck impact

- Caches is SRAM (Static Random Access Memory), its access speed is close to the CPU speed, and it can reduce the bottleneck impact

To increase bottleneck impact

- If multiple CPUs (or cores) share the main memory and thus share the memory bus

A program can never be fully executed in parallel

Formula: $S = \frac{1}{1-P+\frac{P}{N}}$

> Source: [Computer Organization | Amdahl's law and its proof - GeeksforGeeks](#)

- S: speedup of the system
- P: proportion of the system that can be improved
- N: number of processors in the system

(Example): If a system has a single bottleneck that occupies 20% of the total execution time, and we add 4 more processors to the system, the speedup would be:

- $S = \frac{1}{1-0.2+\frac{0.2}{4+1}}$
- S = 1.19
- This means that the overall performance of the system would improve by about 19% with the addition of the 4 processors.

Transferred to parallel computers, this means that with a growing number of CPUs, the problem size should grow too

The problem needs to scale with the number of CPUs

Gustafson's Law

Amdahl's Law

Published in 1967, named after Gene Myron Amdahl

Calculates the maximum expected acceleration of programs by parallel execution on multiple CPUs

According to Amdahl, the performance gain is limited mainly by the sequential part of the problem

Week 3 + 4: Cluster Computing, FaaS, Container, Function as a Service

Week 5 + 6: Cloud Computing, Services and Concepts

Week 7 + 8: Infrastructure Services (IaaS), Platform Services (PaaS)

Week 9 - 10: Hadoop - MapReduce - HBase