Паралелно и дистрибуирано процесирање

Домашна 1

Стефан Милев | 206055

1.1

a)

High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation. It is interested in computational operations per second.

b)

High-throughput computing (HTC) is a computer science term to describe the use of many computing resources over long periods of time to accomplish a computational task. The performance goal thus shifts to measure high throughput or the number of tasks completed per unit of time.

c)

Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application.

d)

Cluster:

- Network of compute nodes interconnected by SAN, LAN, or WAN hierarchically.
- Homogeneous nodes with distributed control, running UNIX or Linux.
- High-performance computing, search engines, and web services, etc.

Grid:

- Heterogeneous clusters interconnected by high-speed network links over selected resource sites.
- Centralized control, server oriented with authenticated security.
- Distributed supercomputing, global problem solving, and data center services.

e)

Service-oriented architecture (SOA) is a style of software design where services are provided to the other components by application components, through a communication protocol over a network. A SOA service is a discrete unit of functionality that can be accessed remotely and acted upon and updated independently.

f)

Pervasive computing refers to computing with pervasive devices at any place and time using wired or wireless communication (Internet of things). The term Internet computing is even broader and covers all computing paradigms over the Internet.

g)

Virtual machines (VMs) offer novel solutions to underutilized resources, application inflexibility, software manageability, and security concerns in existing physical machines. Virtual infrastructure is what connects resources to distributed applications. It is a dynamic mapping of system resources to specific applications.

h)

Cloud services are considered "public" when they are delivered over the public Internet, and they may be offered as a paid subscription, or free of charge. Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third party, and hosted either internally or externally. Architecturally, there are few differences between public- and private-cloud services, but security concerns increase substantially when services (applications, storage, and other resources) are shared by multiple customers.

i)

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. An RFID tag consists of a tiny radio transponder; a radio receiver and transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device, the tag transmits digital data, usually an identifying inventory number, back to the reader.

j)

The Global Positioning System (GPS) is a satellite-based radionavigation system owned by the United States government and operated by the United States Space Force. It is one of the global navigation satellite systems (GNSS) that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.

k)

Wireless sensor network (WSN) refers to a group of spatially dispersed and dedicated sensors for monitoring and recording the physical conditions of the environment and organizing the collected data at a central location. WSNs measure environmental conditions like temperature, sound, pollution levels, humidity, wind, and so on.

l)

The Internet of Things (IoT) is a networked connection of everyday objects including computers, sensors, humans, etc. The IoT is supported by Internet clouds to achieve ubiquitous computing with any object at any place and time.

m)

A cyber-physical system (CPS) is the result of interaction between computational processes and the physical world. A CPS integrates “cyber” (heterogeneous, asynchronous) with “physical” (concurrent and information-dense) objects. A CPS merges the “3C” technologies of computation, communication, and control into an intelligent closed feedback system between the physical world and the information world, a concept which is actively explored in the United States.

1.2

1)

Answer: c

2)

Answer: b

1.3

a)

- Virtualized cluster of servers over data centers via SLA
- Dynamic resource provisioning of servers, storage, and networks
- Upgraded web search, utility computing, and outsourced computing services
- Google App Engine, IBM Bluecloud, AWS, and Microsoft Azure

b)

- Distributed computing (cluster, Grid Computing)
- Internet technologies (Service-oriented architecture, Web 3.0, etc.)
- Hardware Technologies (Multi-core chips, Virtualizations, etc.)
- System Management technologies (Automatics computing)

c)

- Freemium model: This business model seeks to lower barriers to entry for online offerings by offering some core services at no charge, then charging a premium if the customer wants to upgrade to something more sophisticated.
- Consumption model: This is the classic, metered, “pay--as--you--go” offering seen with many cloud services.
- Tiered pricing: This is the most common model for enterprise SaaS. The pricing tiers are typically tied into a metric, such as number of seats, modules, data volumes, and Servers.
- Perpetual license model: This is probably as close as it gets to the old model for buying software -- pay one huge sum up front and get it for life. A perpetual license typically bundles in maintenance, support and other professional services. "Traditional perpetual license models are structured as an upfront payment with 18--25% for 'maintenance and support' plus professional services. "Deeter and Jung note that such plans have been superseded by subscription models in recent years.

1.4

- Globus (d)
- BitTorrent (f)
- Gnutella (f)
- EC2 (i)
- TeraGrid (h)
- EGEE (h)
- Hadoop (a)
- SETI@home (j)
- Napster (b)
- BigTable (e)

1.5

a)

Core A and D take 32 units of time to square 32 elements. Core B takes 64 units of time to square 128 elements. Core C takes 21.33 units of time to square 64 elements, i.e. 64/3. So the total execution time that this 4 core processor takes to square all 256 elements in parallel is the worst time of all of the cores, which is core B at 64 units of time.

b)

(32 + 64 + 21.33 + 32) / (64 * 4) = 0.58

Rate = 58%

1.8

a)

Because cloud platforms are already built on top of datacenters, the hardware resources of cluster/grid could be a part of a larger computing cloud. Cloud usually needs some kind of web service, like an app or a web browser. Also, the components of a cloud could be sparsely distributed, which needs a wider network support.

b)

Cloud resources are dynamically provisioned by data centers upon user demand, while clusters aren't. Also, cloud systems provide computing power, storage and flexible platforms for upgraded web-scale application services.

c)

Cloud computing heavily relies on virtualization of all sorts of resources, and it needs stronger protection than cluster/grid.

d)

Cloud is more cost-effective than cluster/grid. Besides the advantages that come with virtualization, the customers of cloud do not have to buy infrastructure before using it. Also, QoS and SLA add extra requirements on the operational and cost models.
In the future, the two computing paradigms can be converged. For example, there could be no so-called cluster. Everything is based on a comprehensive cloud. In this comprehensive cloud, a customer could request compute, storage and network resources as much as a traditional cluster. Thus, clusters are included in the cloud.

1.9

a)

The general computing trend has been to leverage more shared resources over the internet. On the side of HPC we have seen that for the need to share computing resources, the cooperative clusters have replaced homogenous supercomputers. On the side of PCs there is an interest in moving desktop computing to service oriented computing using server clusters and huge databases and data centers over the public internet through powerful network interfaces. Rather than revolutionary, HPC and PC have taken an evolutionary path over the last 30 years.

b)

Sudden changes in processor architecture is not favorable because when the processor architecture changes drastically it will require the instruction set and therefore software for the processor to also change. That will make software and hardware built for the previous processor incompatible with the new architecture design processors. So it will hamper backward compatibility.
The issue of the memory wall has appeared because the improvement of the CPU has exceeded that of the DRAM memory speed.An important reason for this difference is the limited communication bandwidth beyond chip boundaries. The memory wall is stopping the advances in scalable performance because the advantages of higher clock speeds are negated by memory latency, because memory access times haven't advanced as much as clock frequencies.

c)

x86 is the name of a processor instruction set, a collection of mathematical and logical operations that a processor is able to perform. Almost all processors today maintain compatibility with the x86 instruction set. Reasons why x86 instruction set processors are still dominating the PC and HPC markets today are:

- Backwards compatibility with previous architectures. Meaning x86 processors support older applications built for older architectures
- The x86 processors are cheap and readily available
- The x86 architecture can be virtualized
- x86 has matured to multi-core, high clock speeds processors, which would need time for a new architecture to achieve

1.10

a)

They are both parallel architectures. However, GPUs have a more specialized architecture set which is designed to do highly parallelized complex algorithms for video graphics. While CPUs have a more general-purpose architecture set which is harder to parallelize, but is able to be used for a vast larger set of problems

b)

Processor technology and speed follow Moore’s law and advances rapidly at an exponential rate. Which makes parallel programming redundant most of the time because it cannot match the progress of processor technology. Also not all programs can be parallelized or they are split into a serial and a parallel part, which means there is a maximum benefit in parallelizing a program. And lastly, parallelizing a program comes with extra communication overhead which decreases the benefits.

c)

One way would be to use thread-level parallelism. With core scaling we could increase the number of cores on a chip by a lot. Which would support a large number of concurrent threads. Also, if we cannot increase core usage and program parallelization, we could at least use DVFS to save energy power.

d)

SSDs use solid-state memory and non moving parts, they store memory in non-volatile memory chips which makes them a lot faster. While traditional HDDs use spinning magnet disks to store data which is orders of magnitude slower. This speed benefit of SSDs would deliver better speedups for HPC and HTC systems.

e)

At its current technology, INfiniBand and Ethernet can both achieve bandwidths of 10Gbps, and are cost-effective. Which for today's requirements of HPC systems is satisfactory. For ethernet there is already an IEEE standard for 40/100 Gbps. Which is sufficient for the near future of HPC systems. Therefore both of them will continue dominating the HPC market, until higher speeds are required.

1.11

Single-threaded superscalar:

- Characteristics: instructions are issued from a sequential instruction stream; the CPU dynamically checks for data dependencies between instructions at run time

(vs. software checking at compile time); the CPU can execute multiple
instructions per clock cycle

- Advantages: backward compatibility; can execute any instruction set; branch prediction, out-of-order execution and speculative execution can easily find parallelism over multiple basic blocks and multiple loop iterations

o Disadvantages: must perform to check what instructions that can be executed are not free; transistors are needed to implement them, those checks can also consume a considerable amount of power, becoming less used in small embedded systems; speculative execution opens new side-channels for sidechannel attacks (SPECTRE family of attacks, 2018)

- Representative processors: Seymour Cray's CDC 6600 (1966), the Motorola MC88100 (1988), Intel i960CA (1989), AMD 29000-series (1990), P5 Pentium, Nx586, P6 Pentium Pro, AMD K5

Fine-grain multithreading:

- Characteristics: issues instructions for different threads after every cycle such that no instructions from the thread are in the pipeline concurrently, but it can use the entire issue width of the processor; that hides all sources of vertical waste, but does not hide horizontal waste; it is the only model that does not feature simultaneous multithreading
- Advantages: pipeline utilization; no need for dependency checking between instructions (only one instruction in pipeline from a single thread); no need for branch prediction logic; otherwise-bubble cycles used for executing useful instructions from different threads; improved system throughput, latency tolerance, utilization
- Disadvantages: complicated hardware; costly; reduced single thread performance (one instruction fetched every N cycles); resource contention between threads in caches and memory; dependency checking logic between threads remains (load/store)
- Representative processors: Sun Microsystems' UltraSPARC T1, Seymour Cray's CDC 6600, Tera MTA

Coarse-grain multithreading:

- Characteristics: switches to issue instructions from another thread when the current executing thread causes some long latency events (i.e. page faults); more common for less context switch between threads
- Advantages: sacrifice very little individual thread performance
- Disadvantages: need short in-order pipeline to gain performance; cannot tolerate short latency
- Representative processors: Intel's Montecito, Northstar, IBM's Pulsar Power PC

Simultaneous multithreading (SMT):

- Characteristics: fine-grained, dynamically share the pipeline; can multithread and out-of-order processor; dispatch instructions from multiple threads in the same cycle (to keep multiple execution units utilized); the processor must be

superscalar to do so
- Advantages: tolerate all latencies; higher utilization; sacrifice some individual threads' performance
- Disadvantages: depends on the design and architecture of the processor, SMT can decrease performance if any of the shared resources are bottlenecks for performance
- Representative processors: Intel Pentium 4, IBM's Blue Gene/Q, IBM POWER5, IBM z13, Sun Microsystems' UltraSPARC T1, Oracle Corporation SPARC T3, Fujitsu SPARC64, Intel Itanium Montecito, Intel Xeon Phi, AMD Bulldozer microarchitecture FlexFPU, AMD Zen microarchitecture

Multicore chip multiprocessor (CMP):
- Characteristics: integrates two or more processors into one chip, each executing threads independently
- Advantages: can complete more work than single-core processor; works great for multi-threading apps; can complete simultaneous work as low frequency; can process more data than single-core processor; can complete more work while consuming low energy as compared to the single-core processor; users can do complex works like scanning of the virus by antivirus and watching a movie at the same time; as both cores of processors are on single-chip so computer cache takes advantage and data has not to travel longer; printed circuit board (PCB) need less space in case of using multicore processors
- Disadvantages: difficult to manage as compared to SCP; costly; their speed is not twice than the normal processor; the performance depends upon how the user uses the computer; consumes more electricity; becomes hot while doing more work; if some process needs linear/sequential processing then the multicore processor needs to wait longer
- Representative processors: AMD Ryzen, IBM Xenon, IBM z13, IBM z14, Intel Core 2 Duo, Intel Core i3,i5,i7,i9, Intel Xeon, Intel Xeon Phi, Sun Microsystems' UltraSPARC T1,T2,T3

1.12

a)

This is because virtual machines and virtual clusters offer novel solutions to several serious issues in cloud computing systems, such as underutilized resources, application inflexibility, software manageability and security concerns. Also, with resources virtualized, it would be more cost-effective.

b)

These areas include the virtualization of low-cost machines to provide high overall performance, virtualization techniques to better deal with data deluge, and virtualization that could provide better security.

c)

Cloud platforms can have a significant impact on how HPC and HTC designs, creates, and delivers applications and systems to the customers. For example, with SLA, HPC and HTC systems would face bigger challenges to design the system with minimal delay and maximum throughput. Larger storage space and better reliability are also desired.

1.13

a)

This model put together infrastructures demanded by users, namely servers, storage, networks, and datacenter fabric. The user can deploy and run on multiple VMs running guest OSes on specific applications. THe user does not manage or control the underlying cloud infrastructure, but can specify when to request and release the needed resources. The best examples are AWS, GoGrid, and Rackspace.

b)

This model provides the user to deploy user-built applications onto a virtualized cloud platform. PaaS includes middleware, database, development tools, and some runtime support like Web 2.0 and Java, etc. The platform includes both hardware and software Integrated with specific programming interfaces. THe provider supplies the API and software tools (eg., Java, python, Web2.0, .Net). THe user is freed from managing the cloud infrastructure. The best examples are Google AppEngine, Windows Azure, and IBM BlueCloud.

c)

This refers to browser-initiated application software over thousands of paid cloud customers. The SaaS model applies to business processes, industry applications, CRM (consumer relationship management), ERP (enterprise resources planning), HR (human resources) and collaborative applications. On the customer side, there is no upfront investment in servers or software licensing. On the provider side, costs are rather low, compared with conventional hosting of user applications. The best examples are the Salesforce.com, Google Docs, and Microsoft Dynamix CRM services.

1.17

The main difference between CPU and GPU architecture is that a CPU is designed to handle a wide-range of tasks quickly (as measured by CPU clock speed), but are limited in the concurrency of tasks that can be running. A GPU is designed to quickly render high-resolution images and video concurrently.
Because GPUs can perform parallel operations on multiple sets of data, they are also commonly used for non-graphical tasks such as machine learning and scientific computation. Designed with thousands of processor cores running simultaneously, GPUs enable massive parallelism where each core is focused on making efficient calculations.
While GPUs can process data several orders of magnitude faster than a CPU due to massive parallelism, GPUs are not as versatile as CPUs. CPUs have large and broad instruction sets, managing every input and output of a computer, which a GPU cannot do. In a server environment, there might be 24 to 48 very fast CPU cores. Adding 4 to 8 GPUs to this same server can provide as many as 40,000 additional cores. While individual CPU cores are faster (as measured by CPU clock speed) and smarter than individual GPU cores (as measured by available instruction sets), the sheer number of GPU cores and the massive amount of parallelism that they offer more than make up the single-core clock speed difference and limited instruction sets.
GPUs are best suited for repetitive and highly-parallel computing tasks. Beyond video rendering, GPUs excel in machine learning, financial simulations and risk modeling, and many other types of scientific computations. While in years past, GPUs were used for mining cryptocurrencies such as Bitcoin or Ethereum, GPUs are generally no longer utilized at scale, giving way to specialized hardware such as Field-Programmable Grid Arrays (FPGA) and then Application Specific Integrated Circuits (ASIC).

1.18

Amoeba:

- Written in C and tested in the European community; modified version 5.2 released in 1996
- Still in slightly experimental state, but ready to use
- Fireball Software Distribution (FSD) Amoeba with the current version 2002A, bug-fixed Amoeba-5.3 successor distribution with many new programs and features for the i386 architecture
- Microkernel-based and location-transparent, uses many servers to handle files, directories, replications, run, boot, and TCP/IP services
- A special microkernel that handles low-level process, memory, I/O, and communication functions
- Uses a network-layer FLIP protocol and RPC to implement point-to-point and group communication

DCE:

- Built as a user extension on top of UNIX, VMS, Windows, OS/2
- Middleware OS providing a platform for running distributed apps
- The system supports RPC, security, and threads
- DCE packages handle file, time, directory, security services, RPC, authentication at middleware or user space
- RPC supports authenticated communication and other security services in user programs

MOSIX:

- Developed since 1977, now called MOSIX2 used in HPC Linux and GPU clusters
- A distributed OS with resource discovery, process migration, runtime support, load balancing, flood control, configuration
- MOSIX2 runs with Linux 2.6
- Extensions for use in multiple clusters and clouds provisioned VMs
- Using PVM, MPI in collective communications, priority process control, and queueing services

Each of these Oss have done their job developing the newest clusters, grids and/or Clouds

They will never be commercial systems, because of the lack of versatility

They are made for experimental purposes, not for the end-users

References:

1. Book
2. https://insidehpc.com/hpc-basic-training/what-is-hpc/
3. https://en.wikipedia.org/wiki/High-throughput_computing
4. https://en.wikipedia.org/wiki/Peer-to-peer
5. https://en.wikipedia.org/wiki/Ubiquitous_computing
6. https://en.wikipedia.org/wiki/Cloud_computing
7. https://en.wikipedia.org/wiki/Radio-frequency_identification
8. https://en.wikipedia.org/wiki/Global_Positioning_System
9. https://en.wikipedia.org/wiki/Wireless_sensor_network
10. https://intellipaat.com/community/51251/which-are-the-cloud-computing-key-enabling-technologies
11. https://www.forbes.com/sites/joemckendrick/2013/09/12/the-four-ways-vendors-make-money-from-cloud-computing/#4f97b2684e47
12. https://www.globus.org/what-we-do
13. https://www.globus.org/our-story
14. https://en.wikipedia.org/wiki/Gnutella
15. https://aws.amazon.com/ec2/
16. https://en.wikipedia.org/wiki/TeraGrid
17. https://en.wikipedia.org/wiki/European_Grid_Infrastructure
18. https://en.wikipedia.org/wiki/Apache_Hadoop
19. https://setiathome.berkeley.edu/sah_about.php
20. https://cloud.google.com/bigtable
21. https://www.omnisci.com/technical-glossary/cpu-vs-gpu
22. https://en.wikipedia.org/wiki/Amoeba_(operating_system)
23. https://en.wikipedia.org/wiki/Distributed_Computing_Environment
24. https://en.wikipedia.org/wiki/MOSIX