



Универзитет „Св. Кирил и Методиј“ во Скопје  
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И  
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

## Домашна задача бр. 1

по предметот Паралелно и дистрибуирано процесирање

---

студент: Александар Ивановски  
бр. индекс: 186063  
група: Компјутерски науки

Струмица  
октомври 2020 год.

# Содржина

<b>Проблеми</b>	<b>1</b>
Problem 1.1	1
Problem 1.2	4
Problem 1.3	4
Problem 1.4	5
Problem 1.5	5
Problem 1.8	6
Problem 1.9	7
Problem 1.10	8
Problem 1.11	9
Problem 1.12	12
Problem 1.13	12
Problem 1.17	13
Problem 1.18	14
<b>Заклучок</b>	<b>15</b>
<b>Референци</b>	

# Проблеми

## Problem 1.1

Briefly define the following basic techniques and technologies that represent recent related advances in computer architecture, parallel processing, distributed computing, Internet technology, and information services:

- a. **High-performance computing (HPC)** system, претставуваат компјутерски системи кои нудат на располагање многу моќни пресметковни перформанси и имаат моќ да решат огромни и комплексни пресметковни проблеми (пр. Огромни симулации, обратботка на сеизмички податоци, обработка на сателитски снимки кои доаѓаат во реално време).

Пример за употреба на HPC систем е проектот LIVING HEART PROJECT: A CYBER-HEART, каде големата пресметковна моќ е искористена за креирање на виртуелно човечко срце. (линк: <https://builtin.com/hardware/high-performance-computing-applications>)

- b. **High-throughput computing (HTC)** system, претставуваат компјутерски системи кои имаат способност да ги искористат своите капацитети на подолг временски период за решавање пресметковни проблеми. Пример за HTC систем претставува HTCondor, развиен од University of Wisconsin–Madison.

Клучна разлика меѓу HPC и HTC е тоа што намената на HPC системите е за решавање на пресметковни проблеми во пократок временски период, односно од редот на секунди, минути, часови - пример (пресметки во секунда, илирендерираани рамки во минута, итн..) додека кај HTC фокусот е во извршување на многу пресметки во подолги периоди, од редот на ден, седмица, месец, па дури и години.

- c. **Peer-to-peer (P2P) network**, претставува дистрибуирана мрежа која ги дели задачите и ресурсите помеѓу сите припадници од неа, наречени peers. Секој peer е со еднакви привилегии, и нема централен авторитет во мрежата. Некои примери за P2P мрежи претставуваат Bitcoin - којшто претставува P2P платежна мрежа, BitTorrent - којшто претставува P2P мрежа за размена на ресурси.

- d. **Computer cluster versus computational grid**

Компјутерски кластер претставува мрежа од компјутери од ист тип, чија што задача е да работат заедно како една хомогена целина. Таа целина се користи за извршување на задачи кои имаат големи ресурсни или мемориски побарувања. Сите членови на кластерот се лоцирани на исто географско место и се на мала физичка оддалеченост и се поврзани меѓусебно со голема пропусна моќ.

Пресметковна мрежа претставува мрежа од компјутери кои што се од ист или од различен тип, кои што нудат рамка за извршување различни задачи, секој компјутер може да работи

посебно, но може и повеќе компјутери да работат заедно за решавање на одредена задача. Пресметковните мрежи може да се сочинети од компјутери кои се на различни географски места и со огромна физичка оддалеченост и притоа да се поврзани меѓусебно користејки ја глобалната мрежа - интернетот.

#### e. Service-oriented architecture (SOA)

Сервисно-ориентирана архитектура претставува архитектура за дизајн на софтвер која се состои од повеќе независни и самостојни сервиси кои нудат услуги на различни апликации преку мрежа, користејќи точно определен и однапред дефиниран протокол.

Таков пример претставува сервисот на банките за автентикација на дебитна/ кредитна картичка со нејзниот пин, тоа претставува сервис кој за дадена картичка и пин код проверува дали соодветствуваат, и истиот е користен од повеќе апликации, и тоа: апликацијата за наплата во ПОС терминалите, апликацијата за верификација во ATM, влезот во делот за депоненти во филијалите на банки, и сл..

#### f. Pervasive computing versus Internet computing

Терминот Internet computing ги покрива сите пресметковни парадигми кои на некаков начин се поврзани со, или се одвиваат на интернетот, додека pervasive computing претставува извршување на пресметковни задачи од уреди кои се секојдневно во нашата околина и на некаков начин се поврзани со интернет, пример паметниот часовник цело време врши пресметки на сензорските податоци од жiroskopot вграден во него за да одери колку чекори изминува корисникот, и тие пресметки ги праќа до некој сервер или до некој сервис од дадената апликација, па тие податоци потоа се достапни до крајниот корисник.

#### g. Virtual machine versus virtual infrastructure

Виртуелна инфраструктура претставува инфраструктурата која што е хостирана на одредена физичка инфраструктура и се дистрибуира како сервис на други лица, познат модел за искористување на виртуелна инфраструктура е моделот IaaS, каде лице (физичко или правно) може да закупи комплетно виртуелна инфраструктура за неговите потреби, без да врши никакво нејзино менацирање, и притоа имајќи ја можноста практично бесконечно да ја скалира истата. Компании кои нудат инфраструктура како сервис (IaaS) се salesforce и amazon.

#### h. Public cloud versus private cloud

Public cloud и private cloud претставуваат компјутерски системи и услуги кои се во “cloud” односно се користат по потреба од најразлични корисници од различни физички локации за најразлични цели, private cloud е менациран од страна на компанијата или организацијата која што ја користи, и истата има тотална автономија и контрола во неа, додека public cloud е нуден како услуга од трето лице, и крајниот корисник може да бира од понудените конфигурации и сервиси и нема никаква автономија и контрола за менаџмент врз истите.

**i. Radio-frequency identifier (RFID)**

RFID е технологија која овозможува да се искористат радио брановите за да се прочита информација која е складирана на физички предмет, така наречен таг, со што се нуди уникатен идентификатор за објектот на којшто е закачен таг-от.

**j. Global positioning system (GPS)**

GPS претставува систем за навигација којшто е базиран врз сателитска инфраструктура, каде што секаде на земјата каде што има консталација од повеќе од 4 сателити е возможно со користење на методологии од сателитска геодезија да се пресметат агловни координати, кои потоа можат да бидат користени за најразлични цели, или од тие агловни координати да се пресметат координатите од локалниот координатен систем, користејќи најразлични проекции.

**k. Sensor network**, претставува група од повеќе сензори кои собираат мерења од околината во која оперираат и најчесто ги праќаат до централно место - сервер, којшто е одговорен за нивниот менаџмент.

**l. Internet of Things (IoT)**, претставува систем од објекти кои се заеднички обединети и поврзани со интернет и имаат способност да собираат, обработуваат и пренесуваат податоци, без интеракција од човек.

**m. Cyber-physical system (CPS)**, претставува компјутерски систем во кој одреден механизам е контролиран или набљудуван од страна на компјутерски алгоритам.

## Problem 1.2

Circle only one correct answer in each of the following two questions:

1. In the 2009 Top 500 list of the fastest computer systems, which architecture dominates?

- a. Symmetric shared-memory multiprocessor systems.
- b. Centralized massively parallel processor (MPP) systems.**
- c. Clusters of cooperative computers.

2. In a cloud formed by a cluster of servers, all servers must be selected as follows:

- a. All cloud machines must be built on physical servers.**
- b. All cloud machines must be built with virtual servers.
- c. The cloud machines can be either physical or virtual servers.

## Problem 1.3

An increasing number of organizations in industry and business sectors adopt cloud systems. Answer the following questions regarding cloud computing:

- a. List and describe the main characteristics of cloud computing systems.  
Услуги по потреба на корисниците (On-demand service), голема мрежна пристапност (broad network access), повеќе трајни корисници и ефикасен менаџмент со ресурси (multi-tenancy)

and resource pooling), отпорни на пад и спротивување со него (fail-safe), многу рапидно хоризонтално и вертикално скалирање (rapid horizontal and vertical scalability), еластичност (elasticity), прецисно мерење на услугите (precise measurement of services), бидејќи тоа е неопходно за наплатата од корисниците.

**b.** Discuss key enabling technologies in cloud computing systems.

Дистрибуирано процесирање - за нудење големи пресметковни перформанси  
Интернет технологии за поголема достапност бидејќи огромен дел до апликациите денес, во 2020 работат на веб.

Хардвер - за зголемување на перформансите, но и овозможување на скалирањето.

Системи за менаџмент - За успешен менаџмент од страна на сопственикот на системот.

**c.** Discuss different ways for cloud service providers to maximize their revenues.

Како еден начин за зголемување на добивките е оптимизација на складирањето на податоци, со што може повеќе податоци да се складираат на досегашните ресурси, а тоа значи повеќе може да се понуди на пазарот, со што ќе резултира со зголемување на профит, или од друга страна може досегашната цена на услуга за складирање да се намали, па да се привлечат поголем број на корисници.

Подобрување на менаџментот со споделени ресурси, например во DBaaS (database as a service) честите корисчки барања да се кешираат, со што кога повеќе корисници го побараат истото нешто да им се прати кешираната верзија, одколку од ново да се прави пребарување.

## Problem 1.4

Match 10 abbreviated terms and system models on the left with their descriptions on the right. Enter the description label (a, b, c, ..., j) in the underlined blanks in front of the terms.

<b>d</b> <u>Globus</u>	(a) A scalable software platform promoted by Apache for web users to write and run applications over vast amounts of distributed data
<b>f</b> <u>BitTorrent</u>	(b) A P2P network for MP3 music delivery with a centralized directory server
<b>b</b> <u>Gnutella</u>	(c) The programming model and associated implementation by Google for distributed mapping and reduction of very large data sets
<b>i</b> <u>EC2</u>	(d) A middleware library jointly developed by USC/ISI and Argonne National Lab for grid resource management and job scheduling
<b>h</b> <u>TeraGrid</u>	(e) A distributed storage program by Google for managing structured data that can scale to very large sizes
<b>h</b> <u>EGEE</u>	(f) A P2P file-sharing network using multiple file index trackers
<b>a</b> <u>Hadoop</u>	(g) A critical design goal of clusters of computers to tolerate nodal faults or recovery from host failures
<b>j</b> <u>SETI@home</u>	(h) The service architecture specification as an open grid standard
<b>b</b> <u>Napster</u>	(i) An elastic and flexible computing environment that allows web application developers to acquire cloud resources effectively
<b>e</b> <u>BigTable</u>	(j) A P2P grid over 3 million desktops for distributed signal processing in search of extraterrestrial intelligence

Табела 1: Поврзување на термини со ниви описи

## Problem 1.5

Consider a multicore processor with four heterogeneous cores labeled A, B, C, and D. Assume cores A and D have the same speed. Core B runs twice as fast as core A, and core C runs three times faster than core A. Assume that all four cores start executing the following application at the same time and no cache misses are encountered in all core operations. Suppose an application needs to compute the square of each element of an array of 256 elements. Assume 1 unit time for core A or D to compute the square of an element. Thus, core B takes 1/2 unit time and core C takes 1/3 unit time to compute the square of an element. Given the following division of labor in four cores:

Core A	32 elements
Core B	128 elements
Core C	64 elements
Core D	32 elements

- a. Compute the total execution time (in time units) for using the four-core processor to compute the squares of 256 elements in parallel. The four cores have different speeds. Some faster cores finish the job and may become idle, while others are still busy computing until all squares are computed.

Најслабото јадро ќе биде bottleneck за пресметката, па истата ќе се сведе на:

$$\max(1 \times 32 + 1/2 \times 64 + 1/3 \times 64 + 32) = 64$$

Значи потребни ќе бидат 64 временски единици за да се изврши пресметката.

- b. Calculate the processor utilization rate, which is the total amount of time the cores are busy (not idle) divided by the total execution time they are using all cores in the processor to execute the above application.

За да се пресмета utilization rate, потребно е за секое јадро да се пресмета поодделно така што времето потрено на секое јадро ќе го поделим со вкупното време на извршување на task-от добиено под a. односно со 64 временски единици, и потоа да се направи аритметичка средина, па имаме:

$$\text{Utilization(Core A)} = 32/64 = \frac{1}{2}$$

$$\text{Utilization(Core B)} = 64/64 = 1$$

$$\text{Utilization(Core C)} = 64/3/64 = \frac{1}{3}$$

$$\text{Utilization(Core D)} = 32/64 = \frac{1}{2}$$

Па вкупниот utilization е

$$[\text{Utilization(Core A)} + \text{Utilization(Core B)} + \text{Utilization(Core C)} + \text{Utilization(Core D)}] / 4 = \\ (\frac{1}{2} + 1 + \frac{1}{3} + \frac{1}{2}) / 4 = 0.584$$

Односно utilization rate на процесорот изнесува **58.4%**

## Problem 1.8

Compare the similarities and differences between traditional computing clusters/grids and the computing clouds launched in recent years. Consider all technical and economic aspects as listed below. Answer the following questions against real example systems or platforms built in recent years. Also discuss the possible convergence of the two computing paradigms in the future.

**a. Hardware, software, and networking support**

Cloud платформите се дел од податочни центри, и за нив е потребно интерфејс за пристап за крајните корисници најчесто преку веб, и софтвер за управување преку веб, додека традиционалните кластери и гридови работат на ниско ниво, односно директно изворниот код или задачата може да се извршува на нив, бидејќи најчесто тие се во целосна сопственост на корисниците.

**b. Resource allocation and provisioning methods**

Алокацијата на ресурси во cloud се одвива динамички од страна на податочните центри кои користат софтвери и опрема за контрола на оптоварувањето, додека во традиционалните кластери ресурсите се тие, и ако треба повеќе ресурси, корисникот треба физички да ги овозможи, или да ги побара од тој што е одговорен.

**c. Infrastructure management and protection**

Менаџментот на инфраструктурата и заштита на cloud-от се прави во дата центарот и бидејќи има виртуелизација и споделени ресурси потребно е да се стави особено внимание врз заштита на податоците и споделените ресурси, додека во традиционалните класти и гридови тој процес е по едноставен.

**d. Support of utility computing services**

Cloud-от претставува може да се користи за најразлични помошни пресметки, и тоа без никакви посебни адаптации, додека кластер или грид којшто е оптимизиран за конкретна цел пример нумерички пресметки не така лесно може да се искористи за масовни графички операции како рендерирање или тренирање на невронски мрежи.

**e. Operational and cost models applied**

Во главно помодерните cloud провидери како google cloud platform, за крајните корисници нудат pay-as-you go модел на наплата, односно во зависност од потребите и искористените ресурси се наплаќа, додека традиционалниот грид има најчесто фиксни трошоци за оджување без разлика од работата која што е извршена во тековниот период. Пример ако некоја компанија за своите пресметки користи традиционален кластер или грид управуван од самата компанија, трошоците се фиксни, може цел месец за ништо да не биде искористен кластерот или гридот, но трошоците ќе се акумулираат, додека ако истата компанија користи cloud ќе плаќа толку колку што користи. Истата аналогија може да се направи и од телекомуникациските провајдери односно со моделот на фиксна претплата, и моделот на претплата според потрошено.

## Problem 1.9

Answer the following questions regarding PC and HPC systems:

**a. Explain why PCs and HPCs were evolutionary rather than revolutionary in the past 30 years.**

Пресметковните моќи кои што ги имаме денес може да ни делуваат како практично бесконечни, но тоа не е било така во минатото, првите тогашни flagship компјутери биле со многу слаби перформанси гледано од денешна перспектива, но од тогашна перспектива тие

били огромни перформанси, и од година на година според Муровиот закон се зголемувале и практично ако во една година максимумот перформанс на компјутер или супер-компјутер во банкарска околина било обработка на 100 трансакцији во минута, наредната година тие станувале 200 трансакции, па после две години 300 трансакции, и така со текот на времето евалуирало.

- b.** Discuss the drawbacks in disruptive changes in processor architecture. Why is the memory wall a major problem in achieving scalable changes in performance?

Бидејќи скалирањето постепено се доближи кон физичката горна граница, односно не така лесно може да важи Муровиот закон.

- c.** Explain why x-86 processors are still dominating the PC and HPC markets.

Поради цената на поновите процесори, но и според цената на адаптација на тековниот софтвер којшто работи на тие архитектури. Поевтино претставува да се користи сегашниот софтвер на постара архитектура, одколку да се купи нова архитектура која што ќе побара и изработка на нов софтвер.

## Problem 1.10

Multicore and many-core processors have appeared in widespread use in both desktop computers and HPC systems. Answer the following questions regarding advanced processors, memory devices, and system interconnects:

- a.** What are the differences between multicore CPUs and GPUs in terms of architecture and usage?

CPU е дизајниран да се справува со голем број на задачи брзо, но поради ограничениот број на јадра и начинот на којшто се користи споделената меморија не може да постигне големи нивоа на паралелизација и конкурентност на нитките, додека GPU е дизајниран на начин што има многу повеќе јадра и споделена меморија, а со тоа може да постигне големи нивоа на паралелизација и конкурентност.

- b.** Explain why parallel programming cannot match the progress of processor technology.

Не секој код или програм е лесен да се паралелизира, едноставно некои процеси мора да се извршуваат серијски, односно се каузално зависни еден од друг, но со примена на концепти како MapReduce, може да се подобри паралелизација на некои процеси кои навидум делвуваат како тешки за паралелизација.

- c.** Suggest ideas and defend your argument with some plausible solutions to this mismatch problem between core scaling and effective programming and use of multicores.

Многу инженери се фокусирани на развој на повеќе јадра, наместо уште јадра, може да се искористи сегашниот број на јадра, но да се искористат техники за нивно максимално искористување, тоа може да биде MapReduce, Service oriented architecture каде всушност сервисите се разделени на атомично ниво каде еден сервис прави само една задача и со тоа се намалува непотребниот middleware во кодот за управување со различни зависни (coupled) компоненти од традиционално програмирање.

- d. Explain why flash memory SSD can deliver better speedups in some HPC or HTC applications.

Флеш меморијата и ССД дисковите е познато дека нудат огромни брзини за читање и запишување, па на пример доколку некој систем за пресметки, било НРС или НТС треба да обработи огромно податочно множество, тоа множество мора прво физички да биде достапно за тој систем, па доколку има бавни дискови или магнетни траки како медиум за трајно складирање на податоците може повеќе време да биде потребно да се внесат податоците во системот одколку да се обработат, додека флеш меморијата и ССД дисковите нудат забрзување на тој процес од неколку стотици до илјадници пати.

- e. Justify the prediction that InfiniBand and Ethernet will continue to dominate the HPC market.  
Скалирањето на InfiniBand и Ethernet е практично без лимити, секогаш би можело сегашната оптика да се замени со нова, или да се спроведе уште една линија, и уште стотици линии доколку е потребно.

## Problem 1.11

In Figure 1.7, you studied five categories of modern processors. Characterize in Table 1.9 five micro-architectures for designing these processors. Comment on their advantages/shortcomings and identify the names of two example commercial processors that are built in each processor category. Assume a single core in the superscalar processor and the three multithreaded processors. The last processor category is a multicore CMP and each core is assumed to handle one thread at a time.

### **Single-threaded Superscalar**

#### Architecture Characteristics

Single-threaded Superscalar е процесор кој имплементира форма на паралелизација на инструкциско ниво во еден процесор. Super-scalar може да извршува повеќе од една инструкција во текот на еден циклус на начин што ги распоредува инструкциите до различни извршувачки единици на процесорот.

#### Advantages/ Shortcomings

Предност е тоа што овозможува повеќе инструкции да се извршат во еден такт циклус, со што се намалува CPI - clocks per instruction.

Недостатоци, се тоа што бара комплексна и скапа логика за менацирање на повеќето инструкции кои ги извршува во моментот, односно треба постојано да има логика за промена на иминњата на регистрите, бидејќи во текот на еден циклус ист регистар се користи од различни инструкции за различни цели.

#### Representative processors

Претставници на оваа архитектура се: Nx586, P6 Pentium Pro, AMD K5, Motorola MC88100, Intel i960CA.

## Fine-grain Multithreading

### Architecture Characteristics

Fine-grain Multithreading е варијација на архитектурата Simultaneous multithreading, која вкупност претставува начин за подобрување на superscalar процесорите. Fine-grained multithreading извршува инструкции за различни нитки после секој циклус.

### Advantages/ Shortcomings

Предност е тоа што се избегнува изгладнувањето бидејќи сите нитки рамномерно го користат процесорското време.

Недостатоци се тоа што може да се зголеми работното чекање односно некоја блокирана инструкција на празно да троши процесорско време.

### Representative processors

Претставници на оваа архитектура се : IBM's Blue Gene/Q, IBM z13, UltraSPARC T1 (познат и како "Niagara").

## Coarse-grain Multithreading

### Architecture Characteristics

Coarse-grain Multithreading е варијација на архитектурата Simultaneous multithreading, која вкупност претставува начин за подобрување на superscalar процесорите. Coarse-grain multithreading почнува да извршува инструкции од друга нитка, кога тековната нитка од која се извршуваат инструкции ќе предизвика подолго чекање - на пример page fault или слично.

### Advantages/ Shortcomings

Предност е тоа што ефикасно се справува со чекањето, односно го намалува работното чекање.

Недостатоци се тоа што може да се јави проблемот на изгладнување, односно некоја нитка да го зазема целото процесорско време, додека останатите нитки чекаат на ред.

### Representative processors

Претставници на оваа архитектура се: Fujitsu SPARC64 VI, Intel Itanium Montecito, Intel Xeon Phi.

## Simultaneous Multithreading (SMT)

### Architecture Characteristics

Simultaneous multithreading е архитектура која претставува начин за подобрување на superscalar процесорите, така што извршува повеќе независни нитки, за подобро да ги искористи ресурсите и да го подобри целокупниот перформанс на системот.

### Advantages/ Shortcomings

Предности се тоа што се намалува изгладнувањето на нитките, и се подобрува целокупниот системски перформанс

Недостатоци се тоа што голем дел од претставниците на оваа архитектура немаат добри API за комуникација со оперативните системи, исто така од криптографски експерти се потенцирани големи безбедносни ризици во процесот на промена на нитките.

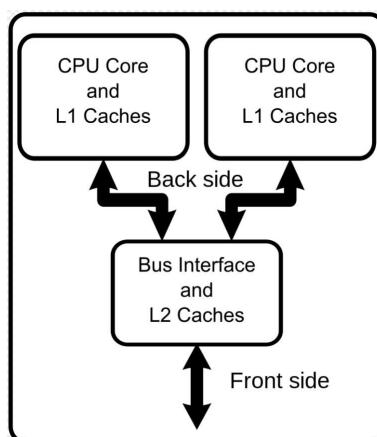
#### Representative processors

Представници на оваа архитектура се: AMD Bulldozer, AMD Zen, како и сите представници на fine-grain multithreading i coarse-grain multithreading кои се варијации на оваа архитектура.

## Multicore Chip Multiprocessor (CMP)

#### Architecture Characteristics

CMP претставува архитектура на процесори, така што во еден процесор има две или повеќе независни процесирачки единици наречени јадра, кои се способни да извршуваат повеќе инструкции независно едни од други, и се ствара ефектот како компјутерот да има повеќе процесори.



Слика1: Пример за процесор со две јадра

#### Advantages/ Shortcomings

Предности се тоа што оваа архитектура овозможува големи нивоа на паралелизација со што се подобрува целокупниот перформанс на системот, овозможува процесори кои работат на поголема феквенција, односно имаат повеќе такт циклуси, се повеќе енергетски ефикасни за разлика од неколку едно-јадредни процесори во ист систем.

Недостатоци се тоа што мора да се направат модификации на софтверите и самите оперативни системи кои не се наменети за повеќе јадра, и исто така при развојот на софтверот треба да се обезбеди поголемо ниво на паралелизација, за да би можеле да се искористат придобивките кои ги нуди оваа архитектура.

#### Representative processors

Представници на оваа архитектура се:

IBM:

POWER4, a dual-core PowerPC processor, released in 2001.

POWER5, a dual-core PowerPC processor, released in 2004.

POWER6, a dual-core PowerPC processor, released in 2007.

POWER7, a 4,6,8-core PowerPC processor, released in 2010.

POWER8, a 12-core PowerPC processor, released in 2013.

POWER9, a 12 or 24-core PowerPC processor, released in 2017.

PowerPC 970MP, a dual-core PowerPC processor, used in the Apple Power Mac G5.

Xenon, a triple-core, SMT-capable, PowerPC microprocessor used in the Microsoft Xbox 360 game console.

z10, a quad-core z/Architecture processor, released in 2008.

z196, a quad-core z/Architecture processor, released in 2010.

zEC12, a six-core z/Architecture processor, released in 2012.

z13, an eight-core z/Architecture processor, released in 2015.

z14, a ten-core z/Architecture processor, released in 2017.

Intel:

Atom, single, dual-core, quad-core, 8-, 12-, and 16-core processors for netbooks, nettops, embedded applications, and mobile internet devices (MIDs).

Atom SoC (system on a chip), single-core, dual-core, and quad-core processors for smartphones and tablets.

Celeron, the first dual-core (and, later, quad-core) processor for the budget/entry-level market.

Core Duo, a dual-core processor.

Core 2 Duo, a dual-core processor.

Core 2 Quad, 2 dual-core dies packaged in a multi-chip module

Core i3, Core i5, Core i7 and Core i9, a family of dual-, quad-, 6-, 8-, 10-, 12-, 14-, 16-, and 18-core processors, and the successor of the Core 2 Duo and the Core 2 Quad.

Itanium, single, dual-core, quad-core, and 8-core processors.

Pentium, single, dual-core, and quad-core processors for the entry-level market.

Teraflops Research Chip (Polaris), a 3.16 GHz, 80-core processor prototype, which the company originally stated would be released by 2011.

Xeon dual-, quad-, 6-, 8-, 10-, 12-, 14-, 15-, 16-, 18-, 20-, 22-, 24-, 26-, 28-, 32-, 48-, and 56-core processors.

Xeon Phi 57-, 60-, 61-, 64-, 68-, and 72-core processors.

Како и многу други, кои се достапни [овде](#).

Еминентно е дека оваа архитектура го доминира пазарот.

## Problem 1.12

Discuss the major advantages and disadvantages in the following areas:

a. Why are virtual machines and virtual clusters suggested in cloud computing systems?

Заради подобрување на скалирањето, подобрување на достапноста до крајните клиенти.

b. What breakthroughs are required to build virtualized cloud systems cost-effectively?

Адаптација на успешен модел за наплата, пример pay-as-you-go, па така како што доаѓаат

нови клиенти и се зголемуваат барањата, ќе се скалира инфраструктурата, outsourcing на сите активности кои не се клучни за провайдерот, без долгочни договори, бидејќи нештата во cloud светот се менуваат многу брзо, например ако сега за 10\$ може да се добијат 1TB дисков простор, и тоа со некој клиент ако е фиксен договор, може цената драстично да падне па клиентот едноставно да го раскине договорот, или ако ние направиме промени за да се адаптираме кон променетата бизнис околина, тогаш ќе се побунат досегашните клиенти заради не еднаквоста на цената на чинење на услугите за сите клиенти.

**c. What are the impacts of cloud platforms on the future of the HPC and HTC industry?**

Големи перформанси и пресметковна моќ е достапна дури и за индивидуалци и мали компании, со тоа се зголемува бројот на успешни стартапи.

## Problem 1.13

Characterize the following three cloud computing models:

**a. What is an IaaS (Infrastructure-as-a-Service) cloud? Give one example system.**

IaaS нуди cloud базирани сервиси, и pay as you go модел на наплата на сервиси за складирање, мрежа и виртуелизација. Примери се: DigitalOcean, Linode, Rackspace, Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE).

**b. What is a PaaS (Platform-as-a-Service) cloud? Give one example system.**

PaaS нуди софтвер и хардвер којшто е достапен преку интернет и cloud. Примери се, AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos, OpenShift.

**c. What is a SaaS (Software-as-a-Service) cloud? Give one example system**

SaaS нуди софтвер којшто е достапен преку интернет, а е направен и менациран од трето лице. Примери се Google Workspace, Dropbox, Salesforce, Cisco WebEx, Concur, GoToMeeting.



Слика 2: Поделба на менацирањето кај различните cloud computing модели

## Problem 1.17

Compare GPU and CPU chips in terms of their strengths and weaknesses. In particular, discuss the trade-offs between power efficiency, programmability, and performance. Also compare various MPP architectures in processor selection, performance target, efficiency, and packaging constraints.

GPU може да процесира многу побрзо податоци кои бараат паралализам, додека CPU има големо инструкциско множество и може да го контролира секој аспект од компјутерскиот систем, за што графичкиот процесор не е способен. Гледано од аспект на инструкции помалкуте јадра на CPU се генерално побрзи и поефикасни од многуте јадра на GPU, но од аспект на паралелизам еминетно е дека GPU е со многу подобри перформанси, но не секој код може да се паралелизира и да ги искористи придобивките од GPU, па затоа потребно е и добри и брзи CPU.

## Problem 1.18

Compare three distributed operating systems: Amoeba, DCE, and MOSIX. Research their recent developments and their impact on applications in clusters, grids, and clouds. Discuss the suitability of each system in its commercial or experimental distributed applications. Also discuss each system's limitations and explain why they were not successful as commercial systems.

Amoeba претставува дистрибуиран оперативен систем развиен од страна на Andrew S. Tanenbaum, негова основна цел е комплетна апстракција на многуте кластерите во еден компјутерски систем. Мрежниот корисник има чувство како да работи на еден систем, за разлика од DCE е направена конкретна имплементација, но за разлика од MOSIX неговата намена е многу специфична и ограничена.

DCE претставува концептуален дистрибуиран оперативен систем развиен од страна на фондацијата за отворен код во 1990 год. Тој слично како OSI моделот претставува подлога за изградба на други имплементации на дистрибуирани оперативни системи. Неговиот неуспех како конкретна имплементација е во големата доза на апстракција.

MOSIX претставува дистрибуиран оперативен систем развиен од Prof. Amnon Barak, своите почетоци ги бележи уште во 1977 година. Негови главни карактеристки се тоа што корисниците можат да се најват на било која единица и притоа не треба да се свесни каде се извршуваат нивните програми, нема потреба од линкување на апликациите со посебни библиотеки, нема потреба да се копираат податоци за оддалечени единици. Миграцијата во подобри единици е многу едноставна и не бара никаков ангажман од одржувачите, има многу добар load balancer. Негова последна верзија е 4.4.4 од 2017 година.

## Заклучок

Оваа домашна задача ми помогна да ги продлабочам моите знањеа од областа на компјутери со голема пресметковна моќ, основните аспекти на дистрибуираното и паралелно процесирање како и предизвиците кои произлегуваат конкрето од делот на програмирањето, основите на cluster и grid системите, основите на cloud computing како и предизвиците кои произлегуваат во делот за имплементација, менаџмент па и до бизнис моделот за крајните корисници. Домашната ме воведе во процесорските архитектури и основите за нив, во виртуализација, во подетално истражување за дистрибуираните оперативни системи.

## Референци

- Cinar Kilcioglu, Costis Maglaras, - *Revenue Maximization for Cloud Computing Services* [\[link\]](#)
- Omnisci.com - *CPU vs GPU* [\[link\]](#)
- Microsoft Azure Overview - *What is IaaS?* [\[link\]](#)
- Stephen Watts, Muhammad Raza - *SaaS vs PaaS vs IaaS: What's The Difference & How To Choose.* [\[link\]](#)
- Moxix - *cluster management system* - *Information from the official website* [\[link\]](#)
- Wikipedia - *Simultaneous multithreading* [\[link\]](#)
- Wikipedia - *Superscalar processor* [\[link\]](#)
- Wikipedia - *Multicore processor* [\[link\]](#)