

Паралелно и дистрибуирано процесирање

Домашна задача 2

Петар Атанасовски - 216052

1. Што претставува дистрибуиран систем?

Дистрибуиран систем е мрежа од меѓусебно поврзани компјутери кои работат заедно за да постигнат заедничка цел, иако се физички одвоени. Овие компјутери комуницираат и едни со други преку споделување информации и ресурси, овозможувајќи им да ги извршуваат задачите поефикасно и посигурно од една изолирана машина. Дистрибуираните системи се користат во различни апликации, од cloud компјутери и социјални мрежи до обработка на податоци од големи размери и интернет услуги.

2. Наведете кои се архитектури за дистрибуирани системи и кои се нивните предности и недостатоци.

а) Client-Server архитектура:

○ Предности:

Јасно раздвојување на проблемите. Клиентите бараат услуги од сервери кои управуваат со ресурсите, а ова овозможува централизирана контрола, полесно одржување и лесно додавање на повеќе сервери.

○ Недостатоци:

Единствена точка на неуспех (серверот); потенцијални ограничувања на приспособливост бидејќи серверот може да стане тесно грло со премногу барања од клиентот.

б) Архитектура Peer-to-Peer (P2P):

○ Предности:

Нема единствена точка на неуспех; секој јазол има еднаков статус и може директно да споделува ресурси. Децентрализирано, што може да ја подобри толеранцијата на грешки и приспособливост.

○ Недостатоци:

Посложена комуникација и координација меѓу врсниците; може да биде помалку ефикасен за одредени задачи за кои е потребна централизација.

в) Хибридни архитектури:

Некои системи комбинираат елементи на архитектурата **Client-Server** и **P2P** за да ги балансираат предностите и недостатоците врз основа на специфични барања.

3. Опишете го workstation-server моделот. Дали постојат познати имплементации на овој модел?

Workstation-server моделот е архитектонски пристап каде што моќни работни станици (машини на клиенти) се потпираат на централен сервер за извршување на сложени задачи или за пристап до споделени ресурси како бази на податоци, датотеки или апликации. Серверот се справува со обработка и складирање на податоци, додека работните станици се фокусираат на интеракцијата со корисниците и интерфејсот.

Познати имплементации на овој модел вклучуваат Citrix XenApp (порано MetaFrame) и Microsoft's Remote Desktop Services (порано познати како Terminal Services). Овие системи им овозможуваат на повеќе корисници да извршуваат апликации на сервер комуницирајќи преку нивните работни станици, со што се оптимизира искористувањето на ресурсите и обезбедува централизирана контрола.

4. Разгледајте го processor-pool моделот. Обидете се да најдете реални системи каде овој модел се користи и наведете ги.

Processor-pool моделот вклучува група процесори кои работат заедно за да се справат со дојдовните задачи. Овој пристап се користи во различни реални системи и веб-сервери како Apache HTTP Server и Nginx, како и во cloud-компјутерски платформи како Amazon EC2 и Google Cloud's Compute Engine. Овие системи ги дистрибуираат дојдовните барања низ pool(базен) на процесори за ефикасно да управуваат со обемот на работа и да обезбедат ефикасност.

5. Кои се основните карактеристики на Distributed Operating System (DOS)? Побарајте и наведете имплементации.

Distributed Operating System (DOS) се карактеризираат со транспарентност, истовременост, доверливост и приспособливост. Тие обезбедуваат унифициран приказ на ресурсите низ мрежата, поддржуваат истовремени процеси, нудат толеранција на грешки и можат да бидат скалирани за да се приспособат на зголемен број корисници или задачи.

Имплементациите на дистрибуирани оперативни системи вклучуваат:

- **Amoeba**: Истражувачки проект кој имаше за цел да создаде целосно дистрибуиран и отворен оперативен систем.
- **Chrome OS на Google**: Примарно фокусиран на веб-апликации, користи дистрибуирани елементи за задачи.
- **Mach**: Ја вовеле архитектурата на Microkernel на која беа изградени други DOS. Тој не е строго дистрибуиран оперативен систем.
- **VKernel**: Дистрибуиран оперативен систем развиен за архитектурата Intel iAPX432.

6. За најдените имплементации од Прашање 4 наведете каков пристап за кернел користат (пр. monolithic kernel, microkernel, hybrid?).

- **Amoeba**: Microkernel Approach
- **Google's Chrome OS**: Modified Linux Kernel (Monolithic Kernel)
- **Mach**: Microkernel Approach

- **VKernel**: Monolithic Kernel

7. Разгледајте имплементации на оперативни системи кои користат microkernel и наведете за кои намени се користат.

- **QNX**: Примарно се користи во вградени системи, особено во автомобилската индустрија, здравството и воздушната индустрија.
- **MINIX**: Првично дизајниран за едукативни цели, исто така се користи во вградени системи, особено во безбедносно критични апликации.
- **Микрокернаели L4** (на пр., Fiasco.OC, seL4): Најчесто се користат во системи во реално време, високобезбедни средини и апликации каде доверливоста и безбедноста се најважни.
- **LynxSecure**: Се користи во виртуелизација и безбедно партиционирање на вградени системи, особено во одбранбената и воздушната индустрија.

8. Кои се најголемите предизвици во креирање на Distributed Operating System (DOS)?

- **Конзистентност и синхронизација**: Обезбедување дека податоците остануваат конзистентни низ јазлите.
- **Толеранција на грешки**: Механизми за справување со неуспеси, обезбедување сигурност на системот во случај на прекин.
- **Трошоци за комуникација**: Ефикасно управување со комуникацијата помеѓу јазлите за да се избегнат тесни грла.
- **Безбедност и автентикација**: Спроведување силни безбедносни мерки за заштита на податоците и ресурсите во дистрибуирана околина.
- **Управување со ресурси**: ефикасно распределување и управување со ресурси.
- **Транспарентност**: Што поголема транспарентност во дистрибуираниот систем за да изгледа како единствен ентитет.
- **Приспособливост**: Сигурност дека системот може беспрекорно да се скалира за да се приспособи на растечкиот број корисници.
- **Дебагирање и тестирање**: Сложено е дебагирањето во дистрибуирана околина.
- **Ракување со грешки**: Грешки и исклучоци можат да произлезат од различни јазли.

Референци:

- <https://www.javatpoint.com/distributed-operating-system>
- <https://www.geeksforgeeks.org/what-is-a-distributed-operating-system/>
- https://en.wikipedia.org/wiki/Distributed_operating_system
- <https://www.spiceworks.com/tech/cloud/articles/what-is-distributed-computing/>
- <https://www.quora.com/p/2108/explain-distributed-computing-models-with-diagram-1/>
- <https://jrtrivedi.blogspot.com/2016/03/processor-pool-model-in-distributed-24.html>
- <https://somedudesays.com/2020/02/monolithic-kernels-microkernels-and-everything-in-between/>
- <https://www.baeldung.com/linux/monolithic-kernel#:~:text=Monolithic%20kernels%20house%20all%20components,balance%20between%20performance%20and%20modularity.>
- <https://lemp.io/what-are-the-examples-of-distributed-operating-system/>