

17

Модели на разпределени софтуерни арх. Среди и протоколи за разпределени приложения.

1. Гаранитри на паралелна обработка. Метрика, методи за анализ

- "паралелна обработка" и "разпределена обработка"

- нямат ясно определена разлика помежду си

- паралелната може да се съмнява за форма на разпределената, но то също съвръзка

- Груп критерии за разлика

- паралелна - всички процесари ^{имат} имат достъп до общата памет, за да делят инфо помежду си

- разпределена - всички процеси има своя памет

делят инфо чрез сообщения между процеси

- Гарантичен алгоритъм - абстрактно представяне на изчислителни проблем като набор от процеси за едновременно изпълнение.

Характеристики:

- други процесари и логическата им организация

- разпределение на данните

- точки на синхронизация

- модел на междупроцесорния обмен

Сложност:

- при последователен алг. - оценява се спрямо разшира на всички данни

- не зависи от архитектура

- при паралелни алг. - функция на архитектурата и средата на паралелна

- Закон на Amdahl - всеки процес, подложен на паралелна обработка, се състои от част, която може да се изпълни паралелно, и от друга, която трябва да се изпълни последователно.

Блокиродимостта не може да надвиши живото, определено от наименото на последователно изпълняваната част от програмата.

Метрики и методи за анализ:

- Степен на паралелизъм Р-арх. величина, макс брой операции, които могат да са паралелни (изпълнени) паралелно

- Ускорение - $S_p = \frac{T_1}{T_p} \rightarrow$ последователно време за изпълнение

- Ефективност - $E_p = \frac{S_p}{p} < 100\%$ $p = \text{бр. изп. машини (процесори)}$

- Цена на обработката - критерий за макс. брой операции, които могат да се изпълнят за времето на обработката

$$C_p = p \cdot T_p$$

- Конфигурация на използване - описание на действителните инициативни операции - $R_p = \frac{O_p}{C_p} = \frac{O_p}{r \cdot T_p} \rightarrow O_p - \text{бр. на операциите с редица}$
- План на обработка - предоставя със идентични скани и изборът на скани завинти от типа МА (паралелен алг.)
- Изчисник - $R_p = \frac{O_p}{O_1} \rightarrow O_1 - \text{бр. операции с 1 процесор за последователна обработка}$

2. Модели на РСА. Структури, организация, компоненти, приложение
Модел на дадена СА - описва декомпозицията на компоненти, тяхната функционална композиция, приложението арх. стила и качествените (нефункционални) атрибути на услугата.

2.1. Процедурни

- Грифер: - отдалечено извикване на процедура (Remote Procedure Call)
- извикване на процедура от един компютър, която да се изпълни в различно адресно пространство
- извикването търси да съществува като извикват на локална процедура

2.2. Обектни

- основна концепция на ООДизайнер
- класове и обекти
- Ч приципа на ООМ - да се опишат от шеста място на обекти и модели в системата
- паджет с когато трябва уточняване и extensibility

2.3. Гётмаки

+ важни компоненти модификацията на данни в изходни

- последователност от трансформации във последователност от данни
- Серия от операции, логично синхронизирани
- системата се декомпозира на функ. модули, интерфейсът между които е под формата на потоци, обратни, канали и др.
- приложени при автоматизирани процеси
- видове в зависимост от механизма на свързване между модулите

Пакетна обработка (Batch Sequential)

- пакетът данните са в пакети като обратни
- модулите сканират входен поток и представят изходен

Филтриращи канали (Pipe & Filter)

- данните са FIFO потоци

- декомпозира се до - източник, обработка, канали, консуматори

- обработките - трансформират данни и придават им канали

- активни и пасивни обработки

- Контролни процеси (Process Control) - при брандери системи
- извикване на данни от сензори и управление им с цели действия

2.4. Контекстни

- данните са в централизиране достъпно от всички компютри
- декомпозиция - модул за упр. на данни и операции, извршващи операции
- два модела на ~~изход~~ за модула за упр. на данни:
 - Хранилище (Repository)
 - агентите са активни
 - хранил. е СУБД, обработка
 - Черна дъска
 - модулът за данни е активен, агентите се ~~автоматизират~~ за ~~редовидни~~
 - publish/subscribe механизм

2.5 Щерархични

- фунд. на системата на щерархия на няколко нива по управление
- модулите от различни нива си комуникират с един общица
- едно ниво се явява услуга за това над него, използва се чрез интерфейс
- Технологии - ОС, ТБР/ТР
 - базови услуги, междинен слой
 - щерархия с подуслугами
 - слоевти архитектури

2.6. Асинхронни

- ~~Базират се на неявен обмен.~~ ^(свободни) обменници между обработвателни процеси
- декомпозиция:
 - генератори на събития
 - event-listeners
 - регистратори на събития - регистрират и разпределят събития
- 2 вида асинхр. обмен:
 - свергдан/online - и двата процеса трябва да са активни
 - неизвестни/offline - с посредник на обмена на свидетелствата
 - процес - дъждител, ако приемателите са неактивни
- Технологии за сладо свръзани системи с неявен обмен на свидетелства
- висока склонност към заменяването на компонентите

2.7. Интерактивни

- поддържат ~~интелигентни~~ потребителски интерфейси.
- декомпозиция
 - модул на предоставяне - графичен интерфейс
 - модул на данни - съхранение данни и операции с тях
 - модул на управление - предава операциите от потребител
- сладо свръзан

2 категории:

MVC - архитектуре са равнопоставени

- подходящ за приложения с активно повър. взаимодействие и динамично представяне на данните

PAC - (Presentation - Abstraction - Control)

- иерархично и разпределено управление
- поддържа архитектура базирана на взаимодействията
- за интерактивни системи от концептуализации са адекватни

3. Организация на разпределение

3.1. Клиент-сервер, двуслойна, промеждна и n-слойна модели

① Клиент-сервер - процеси са подадени в 2 групи:

- Сервер - процес, който имплементира конкретна услуга
- Клиент - процес, който изиска услуга от сервера, изпраща заявка и така отв.

Комуникация:

- протоколи без изграждане на връзка - UDP
- протоколи, които установяват връзка - TCP

② Двуслойна арх. - клиентът и серверът са разделени

- клиент - подредителски интерфейс
- сервер - СУБД

③ Тройслойна арх.

(1) Подредителски интерфейс

(2) Слон с буфер логика - логика за обработка на инфо

(3) Слон с данни - съхранава инфо - най-често СУБД

В този случай (2) се намира на отделен сервер, който играе ролята и на клиент и като взаимодейства със сервера с данни.

④ N-слойна арх.

- множество слоев - всички са специфични по роля и отговорност
- модулност, скапризност и лесна поддръжка
- обикновено най-горен - интерфейс
най-долен - СУБД

3.2 РЗР - реалътът

- мрежова арх. с участваници - всички отдава част от своите ресурси
- всеки участник е единствено "клиент" и "подредител" на ресурси
- в контраст с клиент-сервер
- няма нужда от централна мрежова инстанция
- fault tolerant

3.3. Сервъри за приложения и Web-сервъри

① Сервъри за приложения - сервъри специализирани в изпълнението на определени приложения. Съвкупност от компоненти, които могат да бъдат достъпени чрез API.

② Web-сервъри - специализират варианта на сервърите за приложения

- доставят HTML, изображения, мултимедия и др.
- предоставят допълнителни системни функции, като контрол на трафик, каше/декаше на съдържанието, вирт. хостинг

3.4. Метасистеми и гриз

Разпределени системи, използват се за изпълнение на сложни приложения

Категории:

- Клиенти - ин-бо подобни на компютри свързани помежду си чрез съюзник
- Гриз системи - изградени от различни напр. съф. бл. в разл. административни области и с различни пакети

3.5. Сервизно-, модуло-, аспектно-ориентирани архитектури

① Сервизно

SOA - представява съвкупност от принципи за изграждане и поддръжка на близки процеси в голям разп. системи

Характ.:

- Услуги - съфт. компоненти с определен функционален смисъл
- Сервизна инфраструктура - инфраструктура, която позволява съвместна работа между разп. системи (с разл. платформи и технологии)
- Сладо свързване - намаляване на връзките в системата

② Модуло

- базира се на принципите на модулното програмиране
- се съди съда за раздели дигитата от конкр. имплементация

③ Аспектно

- основна цел - подобрена структ. на приложението
 - да се улесни добавянето на нови функции.
- разделят се отдельните гризкивания в отделни модули
- препрограмиращи се функции. (В начало модула се разглеждват като аспекти)

3.6. Софтуерни агенти

Автономни ^{програм} агенти, които могат да изпълняват наказва задачи
задно с други агенти. (евентуално отдалечени)

Основни свойства:

- Автономност - извриване на задачи/възможен на решени ~~да~~ ^{на} ~~човека~~ ^{намеса}
- Комуникативност - могат да взаимодействят с други агенти
- Реактивност - възприемат средата, в която са и могат да реагират на ~~прекъсн~~
- Гурактивност - могат да променят параметрите на средата

Некардинални св-ва за всички софт. агенти:

- Модулност - миграция и/у ≠ възможни
- Адаптивност - възможност за приспособяване към средата
- Непрекъснато - многократно изпълнение на един и същи функции