

Distributed System architecture

2020년 6월 13일 토요일 오전 1:04

Distributed system architecture

분산시스템특징:

장점및 특징:

리소스 공유

openess개방성(다양한 공급자로부터 하드웨어 소프트웨어포함

concurrency병행성(서로다른 컴퓨터 다양한 프로세스처리 및 통신가능

확장성(Scalability) (기능추가가능)

결함내성(tolerance)(정보의 중복으로인해 결함내성)

단점:설계 복잡성,보안,관리힘듦,예측불가능

쓰레드 시스템에서 디버깅어려움

1)complexity 2)Security 3)Manageability 4) Unpredictability

설계이슈: 자원 식별, 통신, 서비스의 질, 소프트웨어 아키텍처

멀티프로세서 아키텍처:분산시스템아키텍처의 가장간단한 예
시스템의 프로세스 단위마다 다른 하드웨어 프로세서가 실행됨
하나의 컴퓨터보다 여러개의 컴퓨터를 쓴다.

Real-time-systems 예흔히쓰음

System types

Personal systems:개인컴퓨터

Embedded systems: 단일 프로세서또는 통합 프로세서 그룹에서 실행

Distributed systems: 네트워크로 link된 프로세서들의 느슨한 통합에쓰임

Middleware

다른 Framework간 데이터 주고받는데 있어서 middleware가 중간에서 manage해줌

Transaction processing monitors

Data converters

8 Distributed systems architectures 종류

Client-server architecture : 서비스를 제공하는 서버는 서비스를 이용하는 클라이언트와는 다르게 취급된다.

서버 클라이언트알아야함

클라 몰라도됨

논리적 프로세스와 프로세스가 1대1 매칭될필요없음

Distributed object architecture: 딱히 클라이언트와 서버의 차이가 존재하지 않는다.

서비스 주기도하고 요청하기도 함

장점: 사용자들의 결정을 늦게해도 일이 진행될수 있다.

기존시스템에 새로운기능을 쉽게 추가 할 수 있음.

단점: (Flexible) 더복잡함

Client-server architecture

15 Layered application architecture(Client-server architectures)

1)Presentation layer: 연산결과를 보여지는것

2)Application processing layer: 실제 연산하는것이랑 연관

3)Data management layer: 데이터관리

17 Thin and fat clients

Thin client model: application data managemet가 서버측에서 client단순하게
presetaion보여지는것만됨

Fat client model: datamanage만 서버 나머지 client

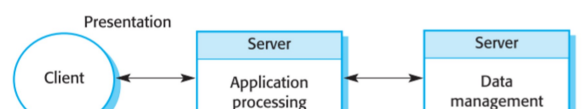
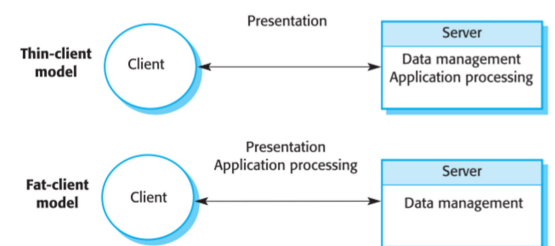
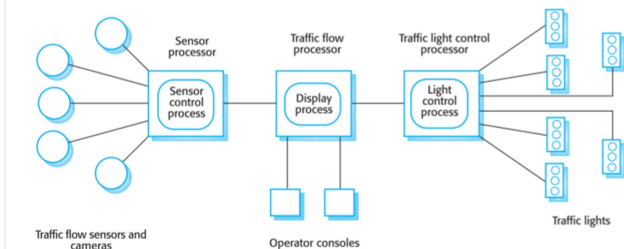
19 Thin CL

기존 만들어진 시스템을 클라이언트 서버아키텍처에 가져올때

장점: 보안안전

단점: 서버가 감당해야하는 프로세싱 로드가 너무크다.

20 Fat client



기존 만들어진 시스템을 클라이언트 서버아키텍처에 가져올때

장점: 보안안전

단점: 서버가 감당해야하는 프로세싱 로드가 너무크다.

20 Fat CL

빈번한 업데이트 힘들수있음

22 Three tier architectures

각각 application layer가 각각의 프로세서(컴퓨터)에서 구동됨

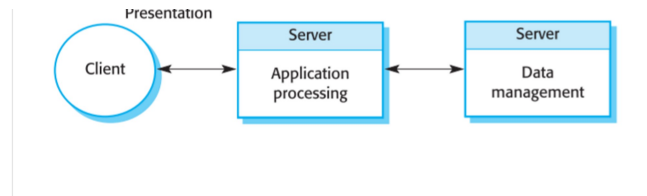
Fat client보다 단순하고 관리하기 쉬움

Thin보다 퍼포먼스 좋음 ,사용 어플리케이션에 따라 다름

게임의경우 Three보다 Application processing가 클라이언트에서 일어나야함

24 internet banking system =three tier

25 visualization 해야 할때는 fat을써야함



26 Distributed Object architectures

Object에서는 클라 서버 정확한 구분존재X

서비스를 받거나 제공할수 있음.

오브젝트리퀘스트브로커라는 미드웨어 사용가능

그러나 더 복잡하게 만들어짐

28 장점: 사용자들의 결정을 늦게해도 일이 진행될수 있다.

기존시스템에 새로운기능을 쉽게 추가 할 수 있음.(Flexible)

29.시험내기 난해한부분 참고용으로만

30 a data mining system

새로운데이터들을 만들어냄 새로운 유형의 관계를 이끌어냄

단점:복잡함

31~ CORBA(시험안냄) //미드웨어에대한표준

40 표준으로되어있으면 남들이 해놓은걸쓸수있음

단점:간단한 시스템을 만들때에도 표준을 이해해야함

42 inter-organisational computing (보안수준의이유로 높은기업수준에서 관리됨

43 peer to peer architectures

P2p는 중앙이없이 1대1로 주고받음semi도존재

오버헤드가 엄청날 수 있음

Semi 가존재 처음연결때 서버가 알려줌

47 service oriented architectures(생략)

Webservice

50 service and distributed objects

독립적이고 코드관점에서 단순해지고

서비스에 지불해야한다.

반응적이다