Väliohjelmistojen tyypit

Tyypit
- Tapahtumaorientoitu
- Viestiperustainen
- Proseduuripohjainen
- Objekti- ja komponenttipohjainen

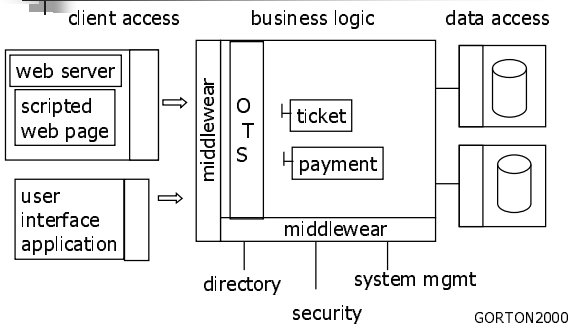
Teemat
- Arkkitehtuurimallit
- Palvelut
- Sisärakenteet
- Laajennustrendit
- Realisaatioita

Tapahtumaorientoidut väliohjelmistot

Teemat

- Järjestelmämallin tarpeet
- Palvelut ja rajapinnat
- Sisärakenteet
  - networking and interoperability
  - server control
  - system administration infrastructure
- Sovellustilanteen arkkitehtuurimallikkeita
- Realisaatioita
- Laajennustrendit

Enterprise application – anatomy



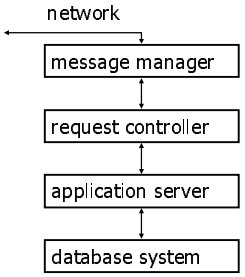
Enterprise application – anatomy

- requirements
  - high availability (99,9-99.999 %)
    - 24 * 7 * 365, downtime 8 h – 5 min /year
  - data integrity and consistency
  - scalability
  - security
- applications
  - customer billing
  - airport management system
  - loan processing system

Enterprise application – anatomy

- services
  - middlewear – message transport, ”plumbing”
  - OTS = object transaction service
  - security serivce for access control and data encryption
  - directory service for locating services and applications in the enterprise system
  - management service: auditing, logging, configuration, alert handling

Traditional mainframe processing



collects transaction input
constructs standard msg
sends transaction output

starts and commits transaction
determines type of request
routes request to proper appl.

executes requested application

Manages shared data

BERNSTEIN198x?

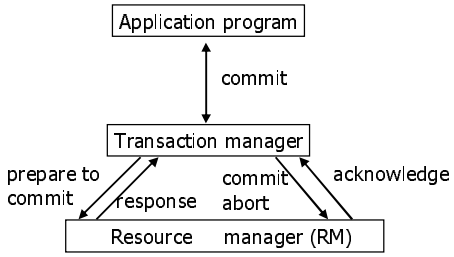
Anatomia

- tietokantaan kohdistuvia kyselyjä ja pyyntöjä
  - stabiili, ennustettava, korkea kuorma
  - muutamia sovellustyyppejä
  - reaaliaikaisuus, dataintensiivisyys
  - transaktioiden perusominaisuudet ACID
- **Services**
  - syncronous, asyncronous, store-and-forward, conversational communication
  - homogeneous cross-platform naming
  - scalability/high performance
  - high availability (recovery, rerouting)
  - workload management by priority management
  - access control

Palvelut: Tapahtumamonitori

- Transaction processing monitor
- Tukee ACID-ominaisuuksia
  - atomicity – kokonaan tai ei lainkaan
  - consistency – tieto vain hyvinmääritellyissä tiloissa
  - isolation – välitilanteet eivät näkyvissä, lukitus
  - durability – tapahtumien vaikutusten pysyvyys
- Hoidetaan
  - toipuvilla prosesseilla (recoverable processes)
  - 2PC – kaksivaiheinen sitoutumisprotokolla
- Palveluaspektit
  - sovelluskehitys
  - suoritusympäristö
  - hallintarajapinnat
  - yhteydet erilaisiin tietokantoihin

Kaksivaiheinen sitoutumisprotokolla Two-phase commit



Rajapinta?

- TM
  - begin
  - commit
  - rollback
  - open
  - close
  - info
  - set-commit-return
  - set-transaction-timeout
- RM
  - start
  - end
  - prepare
  - commit
  - rollback
  - recover

API

```
interface Current{
    void begin()
        raises(SubtrUnavail);
    void commit (in boolean
        report_heuristics)
        raises (notrans,
        heuristicmixed,
        heuristichasard);
    void rollback()
        raises (notrans);
    Status get_status();
    string get_transaction_name();
    void set_timeout (in long secs);
    Control get_control();
    control suspend();
    void resume (in Control wihch)
        raises (invalidcontrol);
};
```

”Pitkät” transaktiot

- tilanteissa joissa
  - käytetään verkon takana olevia hitaasti saatavia resursseja
  - suoritetaan loppukäsittelyä (tulostus tms)
- ei halua/tarvetta odottaa transaktion todellista päättymistä
  - ero esim pitkiin CAD-istuntoihin, joissa peruutusriski
  - ACID-ominaisuudet osittain vaarassa kun prosessointia jätetään suoritettavaksi transaktion ulkopuolella
- menettely
  - transaktio kirjoittaa varmistettuun jonoon tulokset kuten tietokantaan ikään
  - transaktio päättyy (commit), tulokset jonossa
  - erillinen sovellus käy purkamassa jonon (toivon mukaan transaktionaalisesti jonoa käsitellen)
- jononkäsittelyominaisuuksia
  - prioriteettijärjestetyt jonot, klusteroidut jonot kuormantasausta varten
  - non-fifo –käsittely, jonon selailu

Lukitseminen

- lukulukot ja kirjoituslukot
- lukkojen konfliktitaulukot
- lukkojen gradunariteetti -> skaalautuvuus, tiedon eheys
- 2PC-protokolla käsittelee myös lukot

Toipuvat prosessit

- TM ja RM ovat toipuvia: kykenevät palauttamaan oman tilansa häiriön jälkeen
  - TM kerää lokia ”transaction log”
  - RM kirjaa transaktion tilan ja datamuutokset, joita transaktio sai aikaan
  - toipumisprosessissa toistetaan lokien tapahtumat, ”replay”
  - optimointimahdollisuuksia …

Suosituksia ja toteutuksia

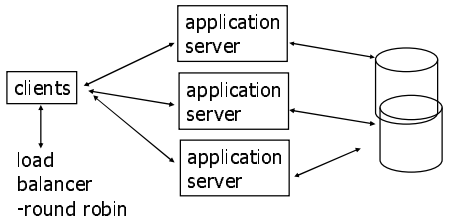
Tuotteita – nopeasti vanhenevaa asiaa

- OTMs
  - BEA WebLogic Enterprise
  - IBM Transarc Encina++
  - Iona OrbixOTM
  - Inprise Visibroker ITS
  - …
- MTMs
  - samoilta ja miksei muiltakin
  - MOM & MQ
- RPC-varianttejakin …

Suosituksia ja kehikoita – nämäkin vielä liikkuvia maaleja

- CORBA transaction service
- J2EE transactions
- .NET transactions
- TIP (Transactions over Internet protocols)?

Optimointia: kuormantasaus



load balancer
-round robin
-by client id
-by type …

fault tolerance, scalability

Optimointia: tilallinen vai tilaton?

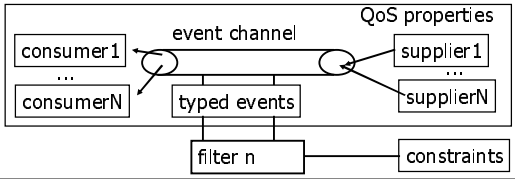
- mid-tier services
- stateless
  - easy to recover, program, scalable
- stateful
  - state not in client
  - clients eacy to recover, program etc
  - no persistency services available in mid-tier -> clumsy
  - cache only read-only information

Optimointia: tulosten reititys

- monissa yrityssovelluksissa on lähetettävä tietoa suurelle joukolle vastaanottajia
- dynaamiset tapahtumien valvonnan ja odotuksen tarpeet
- tapahtumailmoituksen lähettäjän ja vastaanottajien ei tarvitse tuntea toisiaan
- julkaise-tilaa-arkkitehtuuri (publish-subscribe)
- vrt. message-queue-malli

CORBA Events & Notifications

- Event namespace (names and attributes)
- Typed events (header+body; fixed + other)
- Consumer event filtering, event batching, event priority, event expiration, logging, internationalization, flow control mechanism

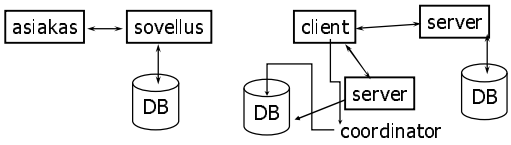


Publish-subscribe

- viestilaatikko yhteinen, kaikki voivat lähettää, tilaajilla suotimet vastaanotettaville viesteille
- taataan että kaikki tilaajat saavat kaikki asiaankuuluvat viestit
- mallit
  - Otsakepohjainen
  - Aiheperustainen
- Ongelmat
  - Skaalautuvuus: viestien vertailu tilausehtoihin
  - Viestien järjestys

Ongelmat ja laajennustrendit

- ongelmat:
  - heterogeeniset tietomallit
  - usein liian raskas sitoutumisprotokolla (pitkäkestoiset tai omnipotentit tapahtumat)
  - puutteellinen siirtosyntaksin tuki
- trendit
  - nested transactions
  - mukaantulo objektiväliohjelmistoihin



Viestipohjaiset väliohjelmistot

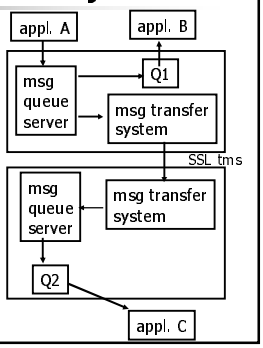
Teemat

- Järjestelmämallin tarpeet
- Palvelut ja rajapinnat
- Sisärakenteet
  - networking and interoperability
  - server control
  - system administration infrastructure
- Sovellustilanteen arkkitehtuurimallikkeita
- Realisaatioita
- Laajennustrendit

Viestipohjaiset väliohjelmistot

Arkkitehtuurimalli

- asyncronous messages
  - reliable, fault-tolerant
  - no loss, duplication, permutation, cluttering
- persistent subscriptions
- models supported
  - message queue
  - request-response
  - multicast
  - publish-subscribe



Viestipohjaiset väliohjelmistot

- MOM = message oriented middleware
- asyncronous messaging natural, syncronous communication cumbersome
- message queues support reliability of message transport
- violates access transparency, no support for data heterogenity unless in programming language mapping, no support for transactions
- suitable for event notifications, publish/subscribe-based architectures
- persistent message queues support fault tolearance

Viestipohjaiset väliohjelmistot

- Perustila: client-server-malli, putki
  - vrt pistoketason ohjelmointi
- Topics for variation and development
  - persistent/transient msgs
  - FIFO/priority queues
  - translations of msgs
  - abstractions on msg ordering
  - multithreading, automatic load balancing
  - msg routing (source, cost, changes in topology etc)
  - secure transfer of msgs (at least between msg servers)

Viestipohjaiset väliohjelmistot

- Services (MQSeries)
  - 4 types of messages
    - Datagram: one way message, no reply
    - RequestMessage: query initiating message
    - ReplyMessage: response to RequestMessage
    - ReportMessage – "exeption report"
  - API: 11 calls
    - Connect/disconnect queue manager, open/close queue, inquire/set (queue) attributes, get/put message from/to queue, put1 (open, put, close), commit/rollback

Viestiperustaiset väliohjelmistot

- Konsortioita
  - MOMA – message oriented middleware association www.moma-inc.org ???
  - BQM – business quality messaging ???
- Välinekuvauksia
  - IBM WebSphere MQ (MQSeries), Sun JMS, Microsoft MSMQ, BEA MessageQ, TIB
  - http://dsonline.computer.org/middleware/projects_MOM.html
  - http://www.ibm.com/websphere/developer/techjournal

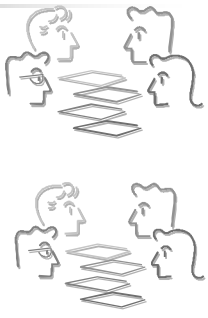
Rajaavampi määritelmä

- The term "messaging" most often refers to electronic mail systems. MOM message movers, however, differ radically from mail systems, because they are high-speed, generally connectionless and are usually deployed for concurrently executing applications with a non-blocking sender. Message-queuing middleware (MQM) is a type of MOM that combines a high-speed message mover and a queuing (message storage) service.

Korhosen seminaarityö

Tuumaustauko ...

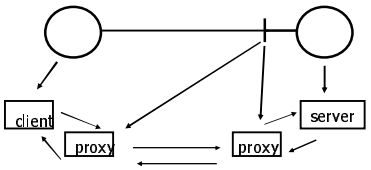
- Onko MOM kuollut?
- Onko malli ehkä näkyvissä uudella tavalla?
  - http ja web? virukset?
  - sulautetut järjestelmät? auton jarrut ja lukitus?
  - miten ydinvoimalan säätö on järjestetty?
- Mitä tiedämme liikuteltavasta tiedosta?
  - MOM vs tapahtumaorient.



Proseduraaliset väliohjelmistot

Teemat

- Järjestelmämallin tarpeet
- Palvelut ja rajapinnat
- Sisärakenteet
  - networking and interoperability
  - server control
  - system administration infrastructure
- Sovellustilanteen arkkitehtuurimallikkeita
- Realisaatioita
- Laajennustrendit

RPC middleware

- Goal of RPC is access-transparent call semantics
  - First class - integrated with language, e.g. Java RMI
  - Second class - language-independent IDL, e.g. CORBA