3. Multi-Computer DBS

Theo Härder
www.haerder.de

Coupling of computers

Spatial distributions

Classification of MC-DBS

Integrated vs. federated architectures

Client/server architectures

Overflow: Decentralized – Self-organizing – Mobile

Main references:

David DeWitt, Jim Gray: Parallel Database Systems: The Future of High Performance Database Systems. Commun. ACM 35:6, 85-98, 1992
Erhard Rahm: Mehrrechner-Datenbanksysteme, Addison-Wesley, 1994, Kap. 3

© 2004 AG DBIS

Technische Universität Kaiserslautern

Classification Aspects

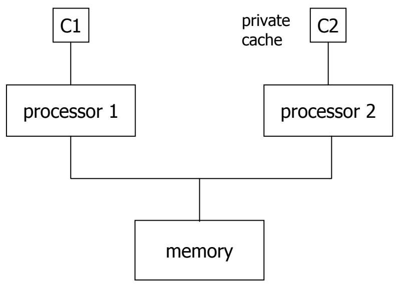
- **Computer coupling**
  tight, loose or close coupling, shared data
- **Spatial distribution**
  - local or spatially distributed
  - communication:
    infrastructure-based or infrastructure-less
- **Attachment of storage**
  shared or partitioned
- **Kind of system cooperation**
  integrated vs. federated MC-DBS
- **Types of system components**
  homogeneous vs. heterogeneous DBMS
- **Functional specialization vs. equalization of processors**
  vertical vs. horizontal distribution

➔ How far does the orthogonality of the classification aspects reach?

© 2005 AG DBIS

Tightly Coupled Systems

- **Properties**
  - shared memory for all processors
  - a single copy of SW-components (OS, DBMS, application, …)
  - widely available (Symmetric Multiprocessing, SMP)



- **Pros**
  - simple realization, few new DB-problems
  - efficient communication via memory
  - load balancing through OS[1]
  - Single System Image
- **Cons**
  - lack of failure isolation (shared memory and SW-components)
  - limited extensibility (N<50, mostly N<16)
  - maintenance of cache coherency
- **Example:**
  DBMS on tightly coupled mainframes

1 But: S/390 "Virtual Image Facility" enables more than 2500 Linux instances to run on a single S/390 server.

© 2005 AG DBIS

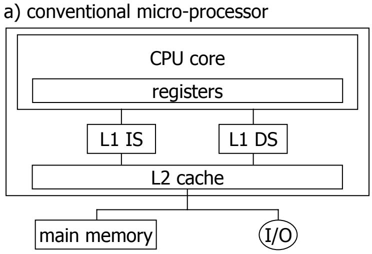
Trends in DBS

Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

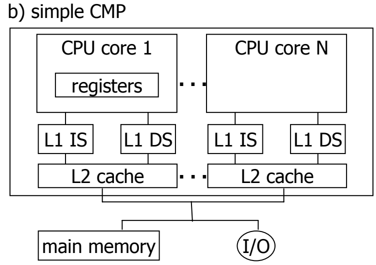
CMP Implementation Options

New option: chip multi-processor (CMP)

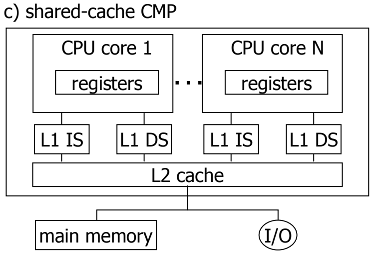
a) conventional micro-processor



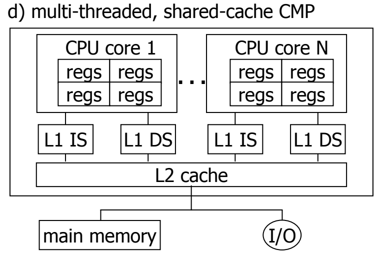
b) simple CMP



c) shared-cache CMP



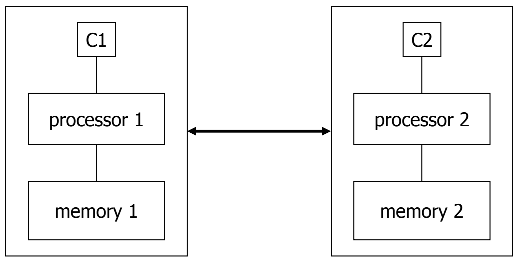
d) multi-threaded, shared-cache CMP



© 2005 AG DBIS

Loosely Coupled Systems

- **Properties**
  - N autonomous computers (separate memory per node, own copy of OS and DB/DC system)
  - each computer can be a CMP
  - communication via message exchange
- **Pros**
  - higher failure isolation / availability
  - improved extensibility
- **Cons**
  - message exchange expensive (communication overhead)
  - no Single System Image



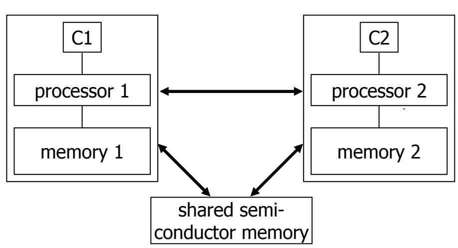
© 2005 AG DBIS

Trends in DBS

Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

Closely Coupled Systems

- **Trade-off between tight and loose coupling**
  goal: more efficient communication as compared to loose coupling while sufficient failure isolation and extensibility is preserved
- **Properties**
  - N autonomous computer nodes (potentially CMPs)[1]
  - communication (partly) via shared semi-conductor memory (GM)
  - prerequisite: local computer placement



1 S/390 Sysplex and successors represent a clustering technology, which enables standard applications such as DB2, CICS, IMS and Unix system services to scale from 2 CPUs to 100 CPUs, with a performance degradation of less than 10 percent. A key element for this purpose is a technology called Coupling Facility, which provides centralized lock management and coherence control of disk caches within a cluster.

© 2005 AG DBIS

Closely Coupled Systems (2)

- **Properties of the GM**
  - fast access (micro-seconds and below) to avoid process switches (*synchronous access*)
  - in general, no instruction-based addressability, non-volatile, if necessary
  - duplication of memory contents, if necessary
- **Further use of close coupling**
  - use of specialized processors, for example, a 'Lock Engine' for global concurrency control

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Spatial Distribution

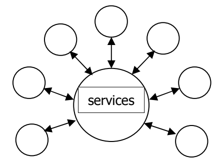
- **Local**
  - fast computer coupling possible (shared memory, high-speed bus, etc.)
  - message exchange more efficient and robust than in Wide-Area Networks (WAN)
    - simpler communication protocols
    - Broadcast, Multicast
  - effective dynamic load distribution possible
  - support of intra-transaction parallelism
  - simpler administration

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

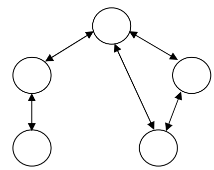
DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Spatial Distribution (2)

- **Distributed**
  - communication
    - "classical": conventional telephone network (Internet)
    - mobile (e.g.: MobileIP): fixed network infrastructure required
    - wireless (IrDA, Bluetooth, WaveLAN): fixed network infrastructure not mandatory



  - future communication
    - via self-organizing infrastructures without mandatory fixed network infrastructure
    - with mobile devices, services and applications



  - support of decentralized forms of organizations
  - prerequisite for fast catastrophe recovery (replicated DBs at remote nodes)

Trends in DBS
Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow
DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Communication Costs

- **Cost for sending/receiving** a message contains three essential components
  - CPU cost for the communication protocol
  - latency (signal propagation delay) for the transfer of the first bit (speed of light ~ $3*10^5$ km/sec; in our model: $2*10^5$ km/sec)
  - duration of transmission for the entire message depending on the existing bandwidth of the communication medium
  - "revolution" in the WAN area: leading-edge technologies even much faster (40 Gbps -1.6 Tbps)

	Shared Memory	Cluster	LAN	MAN	WAN
typical distance	< 10 m	100 m	1 km	100 km	10.000 km
CPU cost per SEND/ RECEIVE	250 instr.	2500 instr.	2500 instr.	25.000 instr.	25.000 instr.
latency	0.05 µs	0.5 µs	5 µs	500 µs	50.000 µs
bandwidth 1990 2006	1 Gbps 100 Gbps	1 Gbps 40 Gbps	10 Mbps 10 Gbps	1 Mbps 1 Gbps	50 Kbps 1 Gbps

Trends in DBS

- Computer coupling
- Spatial distribution
- Classification
- Integrated vs. federated DBS
- Example: ObjectGlobe
- Client/server DBS
- Overall evaluation
- Overflow

Communication Costs

- **Total cost for a message of 10 KByte:**
  - CPU capacity of 10 Mips (1990) resp. 1 Gips (2006)
  - latency remains unchanged

	Shared Memory	Cluster	LAN	WAN
CPU cost 1990	25 µs	250 µs	250 µs	2500 µs
2006	0.25 µs	2.5 µs	2.5 µs	25 µs
latency + transmission 1990	(0.0 + 100) µs	(0.5 + 1000) µs	(5 + 10,000) µs	(50,000 + 2,000,000) µs
2006	(0.0 + 1) µs	(0.5 + 2.5) µs	(5 + 10) µs	(50,000 + 100) µs
sum 1990	125.0 µs	1,250.5 µs	10,255 µs	2,052,500 µs ~ 2,05 sec
sum 2006	1.25 µs	5.5 µs	17.5 µs	50,125 µs ~ 0.05 sec

DBIS Datenbanken und Informationssysteme

© 2005 AG DBIS

Trends in DBS

- Computer coupling
- Spatial distribution
- Classification
- Integrated vs. federated DBS
- Example: ObjectGlobe
- Client/server DBS
- Overall evaluation
- Overflow

Attachment of storage

- **shared**
  - local computer allocation
  - loose or close coupling (*Shared Disk*) resp. tight coupling

    \+ **each processor directly reaches all data** (high potential for load balancing)

    \+ **no partitioning of the database required**

  - new DB problems as to concurrency control, buffer management, logging/recovery, ...

- **partitioned (Shared Nothing)**
  - local or distributed computer allocation
  - loose computer coupling, in general
  - (static) replication of data possible
  - distributed transaction execution, to access and process remote data

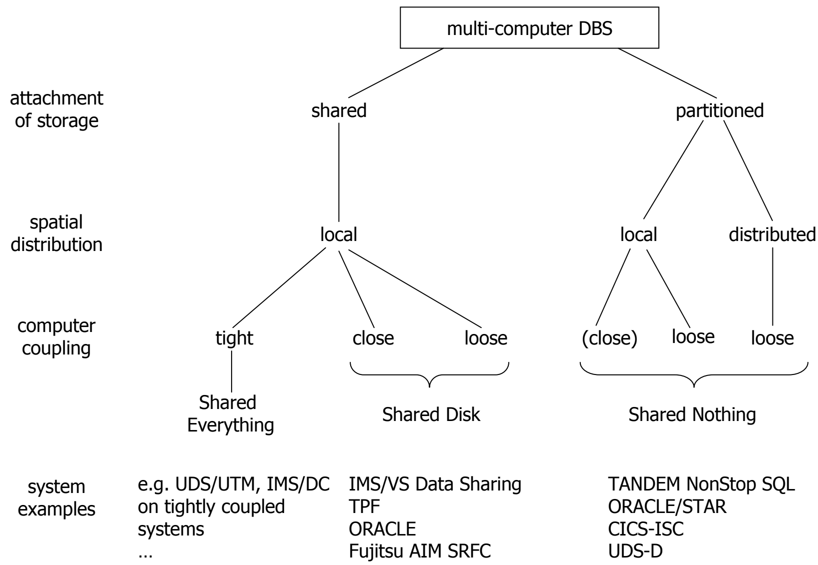
DBIS Datenbanken und Informationssysteme

© 2005 AG DBIS

Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

© 2005 AG DBIS

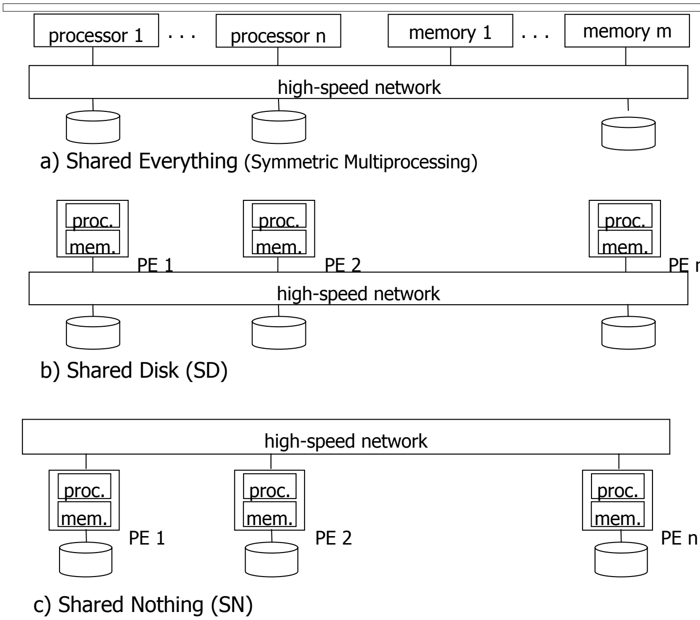
Classification of Multi-Computer DBS



Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

© 2005 AG DBIS

Parallel / Distributed DBS



PDBS Architecture

- **Main alternatives:**
  SE, SD, SN and hybrid forms

- **Different HW platforms possible**
  - **SMP**: Symmetric Multiprocessing[1]
    (all storage accesses happen via the same shared memory bus)
  - **NUMA**: Non-Uniform Memory Access: Sequent, Data General, ...
    (memory regions are reachable via different physical buses: "local and remote" memory)
  - **Cluster**: VMS cluster, NT cluster, Sysplex
    (short connections, efficient inter-process communication, low error rates, homogeneous participants)
  - **MPP** - Massively Parallel Processing: SP2, NCR, NCube, ...

1 auch SMMP: Shared Memory Multi-Processors

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

PDBS Architecture

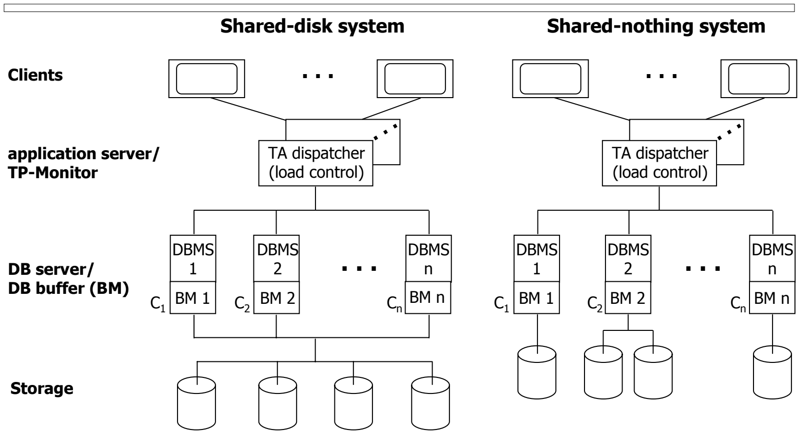
- **The "view" of the DBMS instances is the essential aspect**
  - **SN:** partitioning of the DB among the DBMS instances which determines the resp. query and transaction processing
  - **SE:** direct access to the entire DB for all DBMS instances; centralized data structures (lock table, DB buffer, etc.), data exchange via memory and external storage
  - **SD:** direct access to the entire DB for all DBMS instances; local buffer and lock tables, data exchange via external storage resp. messages

- **Hybrid solutions embodying enhanced complexity**
  - SD- resp. SN-type coupling of SE systems
  - SN-type coupling of SD systems ...

  ➔ "recursive" application of the basic structuring concepts

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Overall Architectures of MC-DBS



Configuration:

- number of computers ?
- broadband communication / local placement
- no "single point of failure"
- (copy of) DB system on each computer

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

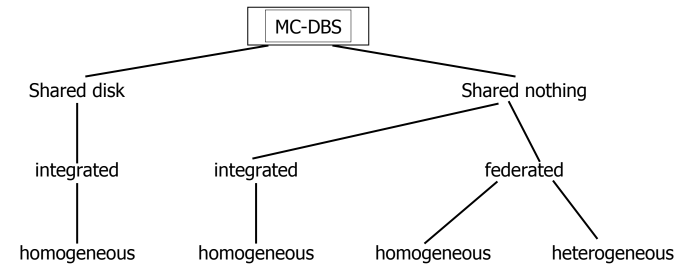
Processor Functionality

- **Functional equalization** ("horizontal distribution")
  - each node has the same functionality w.r.t. DB processing
  - in general, complete DBMS per node
  - replication of functions
- **Functional specialization** ("vertical distribution")
  - partitioning of functions
  - expl 1: DB engines using dedicated processors for certain DB functions (join processor, sort processor, etc.)
  - expl 2: federated DBS
  - expl 3: client/server DBS
  - expl 4: Web information systems (multi-layered architectures) using DB server and DB processing at application server
- **Specialization hampers load balancing, extensibility and fault tolerance**
- **Hybrid forms using horizontal/vertical distribution** (partitioning and replication of DBMS functions)
  - n-server systems: homogeneous or heterogeneous
  - mobile objects are stored where they are currently referenced frequently

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Integrated vs. Federated MC-DBS

- **Integrated MC-DBS**



  - **a single conceptual DB schema**
  - DB access similar to the centralized case (distribution transparency for AP)
  - restricted autonomy for participating DBMS
  - identical DBMS instances (homogeneous MC-DBS), in general
  - example.: Distributed DBS, Shared-disk DBS

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

What are Federated Database Systems?

- Definition
  - A **federated database system** (FDBS) is a collection of cooperating database systems that are autonomous and possibly heterogeneous.
- Wikipedia explanation[1]
  - An **FDBS** is a type of meta-database management system which transparently integrates multiple autonomous database systems into a single federated database. The constituent databases are interconnected via computer network, and may be geographically decentralized. Since the constituent DBS remain autonomous, an FDBS is a contrastable alternative to the (sometimes daunting) task of merging together several disparate databases.
  - Through data abstraction, FDBSs can provide a uniform front-end user interface, enabling users to store and retrieve data in multiple databases with a single query – even if the constituent databases are heterogeneous. To this end, an FDBS must be able to deconstruct the query into subqueries for submission to the relevant constituent DBMSs, after which the system must composite the result sets of the subqueries.
  - Because various DBMSs employ different query languages, FDBSs can apply wrappers to the subqueries to translate them into the appropriate query languages

1) http://en.wikipedia.org/wiki/Federated_database_system

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Integrated vs. Federated MC-DBS (2)

- **Federated MC-DBS**
  - widely independent DBMS having private conceptual DB schemata
  - partial export of schema information for external access
  - heterogeneity possible concerning data models and transaction management
  - strong problems coping with semantic heterogeneity
  - in general, distribution transparency only obtainable in a limited way

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Use of Heterogeneous, Autonomous Data Sets

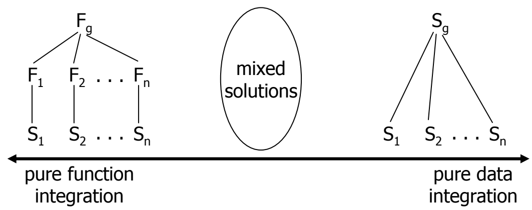
- **Application integration**
  - DB accesses **encapsulated within application functions**
  - access to remote data bases by invocation of **application functions** / Web Services
  - use of standardized communication protocols and message formats (e.g., SOAP / XML)
  - support of distributed transactions by **application server / TP monitor**
  - compared to data integration simpler realization, however, less flexibility and functionality (no table-spanning DB operations like joins, no ad-hoc queries, etc.)

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Use of Heterogeneous, Autonomous Data Sets (2)

- **Data integration**
  - **global view** to distributed data sets
  - **different levels** of integration having partial up to complete transparency
  - federated DBS (FDBS) / mediator systems, VDBS, Data Warehouses
- **Spectrum of integration**

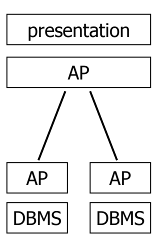
  access via global functions F_g (predefined, composed from F_i)
  vs. access via global schema S_g (using declarative queries, generic)

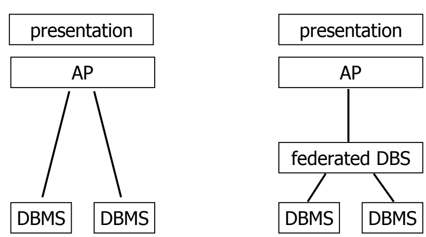


© 2005 AG DBIS

Forms of Integration with Autonomous and Heterogeneous DBS

programmed distribution | distribution of DB operations | federated DBMS





programmed distribution
- DB access via invocation of predefined program fragments resp. of procedures / methods
- Forms of realization:
  - distributed transaction systems (TP monitors)
  - distributed OO systems

distribution of DB operations
- Invocation granularity: DB operation (SQL statement)
- DB schemata must be known to the APs
- Client DBMS can take over role of AP

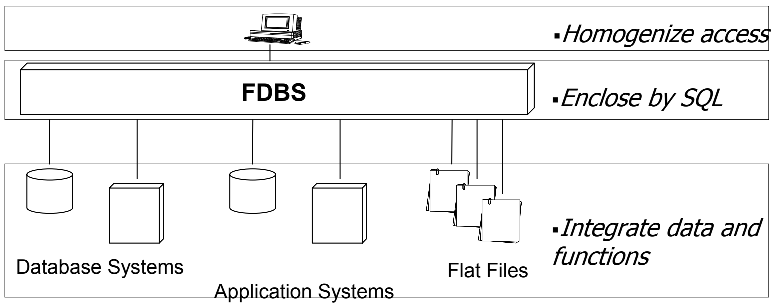
federated DBMS
- FDBS can support unified DB view
- Distributed execution of a DB operation

© 2005 AG DBIS

Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

Function and Data Integration

- **Processing shared data from diverse sources**
  - database systems using generic means of access (SQL)
  - file systems having only simple access operations
  - application systems (APS) encapsulate DB and application
    - access exclusively via API using predefined functions
    - no DB interface available



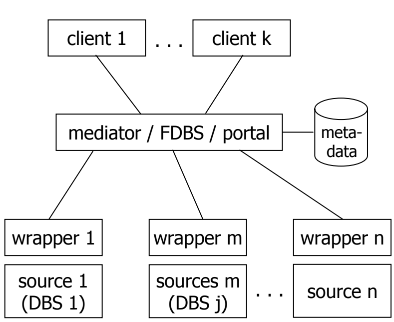
DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Trends in DBS

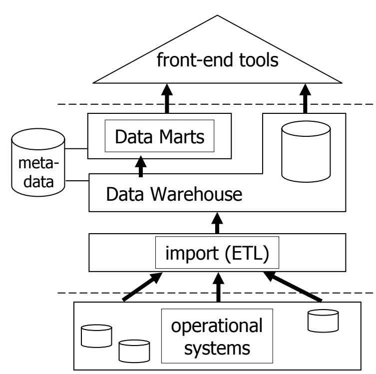
Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

Further Alternatives for Data Integration

Virtual integration (mediator/wrapper architectures, federated DBS)



Physical (pre-) integration (Data Warehousing)



ETL: Extract, Transform, Load

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

What is Grid- and Meta-Computing?

- Interplay of data, functions, and engines (computers)

- **Grid Computing**
  - decomposition of large applications (problems) and their execution (solution) using a large number of geographically distributed computers
  - challenge: dynamic partitioning of data resources
  - typically centralized decomposition, control, and result evaluation in contrast to P2P applications
  - simple appl.: SETI@HOME, factorization of large numbers
  - advanced appl.: sequencing of genomes, exploration of experiments (physics), Einstein Grid to detect gravitation waves, etc.

- **Meta-Computing**
  - context: autonomous computers (in the Web)
  - purpose: dynamic planning and search of resources to spontaneously solve a task
  - principal characteristics: combine
    - arbitrary processing locations (cycle providers), data sources (data providers), application functions (function providers) and
    - solve a problem (composition of a multimedia appl. or query evaluation)

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

ObjectGlobe – Ubiquitous Query Processing in the Internet[1]

- **Goals**
  - provision of an **infrastructure** which allows distribution of mechanisms for query processing (query operators) as simple as publication of data and documents in the Web
  - Clients should be enabled to process complex queries which include execution of operators of **various** providers of **different** Web sites and access to data and documents of **heterogeneous** data sources

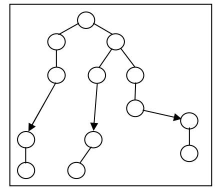
  ➔ query processing by use of query operators (supplied by **functions providers**), of **cycle providers** (computers) and of data sources (delivered by **data providers**); all of them are independent of one another

1 Braumandl, R. et al.: ObjectGlobe: Ubiquitous Query Processing on the Internet, VLDB Journal 10:1, Aug. 2001, pp. 48-71.

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

ObjectGlobe

- **Prerequisites**
  - query operators can be executed in distributed query plans
  - they can be displaced to arbitrary Web sites, e.g, to those **"close"** to the data sources
  - distributed query plans can be composed of **arbitrary query operators**
  - query operators (in Java) observe the secure interfaces of ObjectGlobe



- **Properties**
  - ObjectGlobe ia a **federated system**
  - it embodies a **distributed and open** query processor (query engine) for Internet data sources
  - **meta-data repository** (lookup service) registers all usable data sources, operators and computers
  - optimizer generates a plan to execute a query (QEP: query evaluation plan) which corresponds to the quality demands of the user, if possible

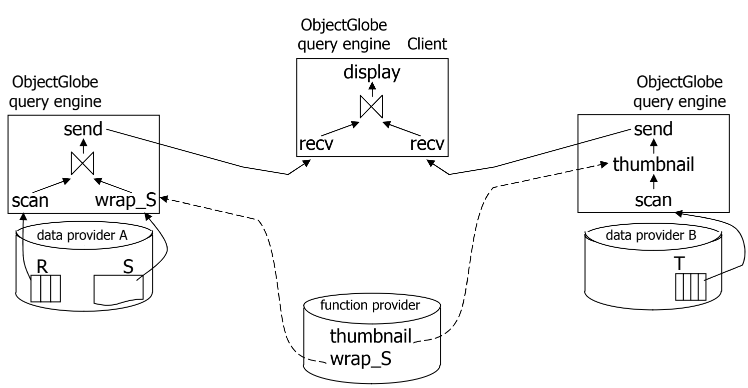
© 2005 AG DBIS

Trends in DBS

Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

ObjectGlobe

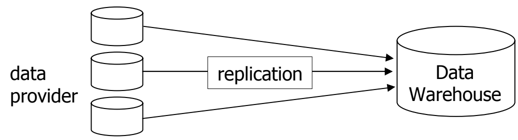
- **Transparent ad-hoc integration** of operators and functions is a challenge for query optimization
- Integration of heterogeneous data is not a core topic. Data are present in a **standard format** (relational, XML) or are encapsulated by wrappers
- **Meta-schema** describes all relevant properties of all services (structure of data, access methods, statistics, cost)



© 2005 AG DBIS

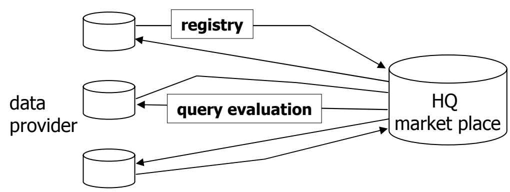
E-Market Place – Application of ObjectGlobe

- **Conventional techniques**
  - replication of data in a centralized Data Warehouse



- problems with schema integration
- restricted operators, …

- **HyperQuery approach**
  - market place is "mediator"



- distribution of query processing along the implicit data allocation scheme
- objects and queries "flow" through the WWW
- scalability

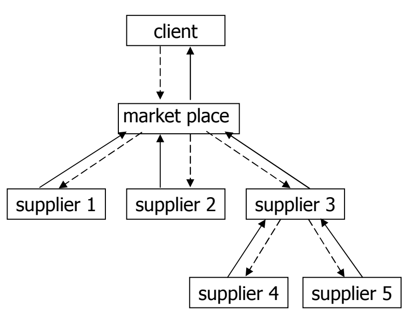
DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Trends in DBS

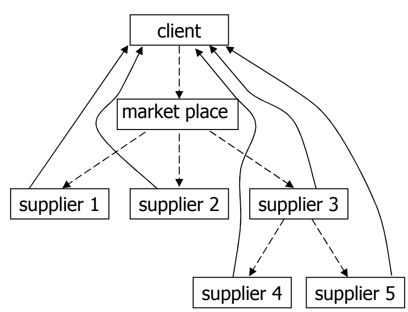
Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

E-Market Place (2)

- **Execution models**
  - hierarchical execution



- broadcast execution



Select	p.ProductDesciption, k.Supplier, k.Price
From	Products p, Catalog@MarketPlace k
Where	p.ProductDescription = k.ProductDescription
Order by	p.ProductDescription
Expires	5/5/2002

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

DBIS Datenbanken und Informationssysteme

© 2005 AG DBIS

E-Market Place (3)

- **Hyperlinks**
  - embedding of hyperlinks as virtual attributes in the DB
  - hyperlinks refer to HyperQueries

product description	supplier	price
battery, 12 V 32 A	supp 1	hq://Supplier1.com/Electric/Price ?ProdId=CB1232
tire, 175/65 TR 14	supp 2	hq://Supplier2.com/Tires/Price ?ProdId=175_65TR14
. . .	. . .	. . .

  - virtual attribute : Price
  - HyperQuery protocol : hq
  - DNS : Supplier1.com
  - HyperQuery ID : Electric/Price
  - object-specific parameter : ?ProdId = CB1232
- **HyperQueries**
  - partial plans to remote engines
  - virtual table HyperQueryInputStream
  - query example: Electric/Price@Supplier1.com as SQL dialect

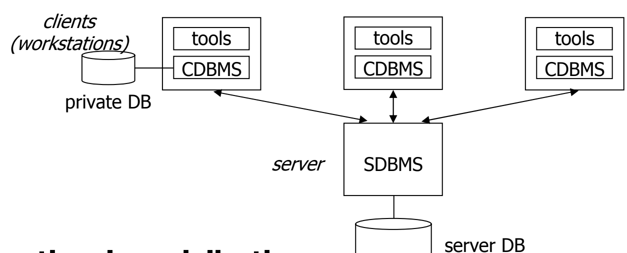
```
Select   h.*, p.Price AS Price
From     HyperQueryInputStream h, Products p
Where    h.ProdId = p.ProdId
```


DBIS Datenbanken und Informationssysteme

© 2005 AG DBIS

Client/Server-DBS

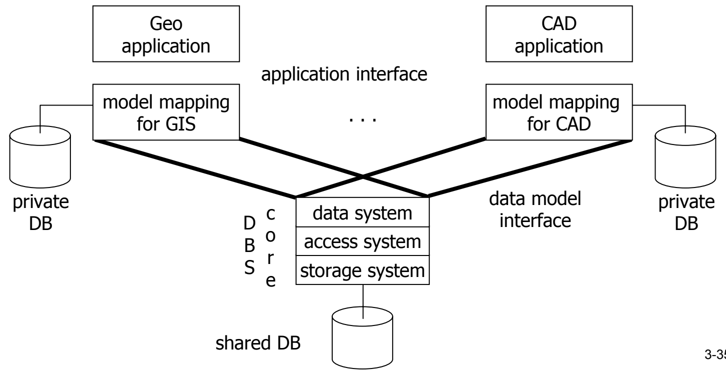
- **OODBS, engineering applications**
  - DB-based processing of large, complex-structured data sets at the client side
  - high re-reference probability for the data
  - long transactions



- **Functional specialization**
  - data management in client **and** server
  - client object buffer: saving of communication requests
  - local execution of queries and methods
  - global tasks at the server: logging, concurrency control, management of external storage

Client/Server DBS (2)

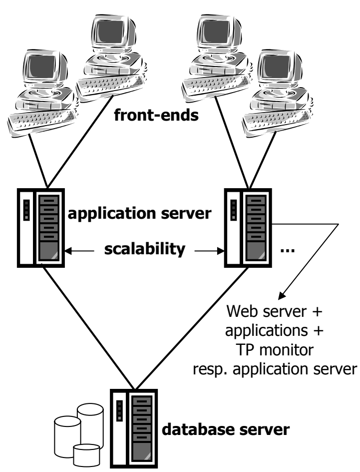
- **DBS core architecture**
  - neutral DBMS core
  - adequate model mapping for a variety of domains
  - redundant modifications in client and server
  - problems of consistency control in long transactions
- **Separation of application-independent and application-dependent processing**



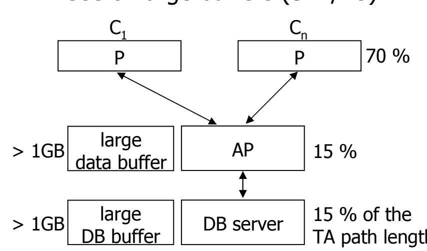
© 2005 AG DBIS

Trends in DBS
Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

Multi-level Client/Server Architectures



- Use of large buffers (SAP/R3)



- **Goals**
  - AP data should be kept close to the application (in AP buffers)
  - concurrency control through AP knowledge at the level of AP servers

  ➔ AP server contains TP-monitor functionality and must reimplement much of the DBMS functionality

© 2005 AG DBIS

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Multi-level Client/Server Architectures (2)

- **Goals**
  - development of data-centric Internet applications
  - reliability, availability, scalability
  - relief of the bottleneck of the DBS processing by memory-resident data management services
- **PDSP based on Shared-Nothing architecture[1]**
  - internally, TimesTen (a copy per node) is used as a memory-resident DBMS for data storage and query processing
  - persistent data storage through disk-based DBMS (like Oracle or DB2)
  - PDSP provides for the applications a location-transparent view (single image) to all distributed DB data

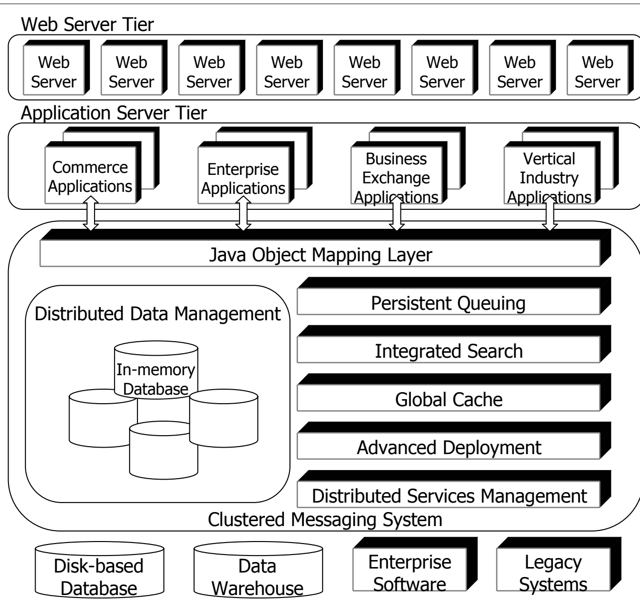
[1] M. Carey et al.: The Propel Distributed Services Platform, Proc. VLDB 2002

Trends in DBS

Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Multi-level Client/Server Architectures (3)



Trends in DBS

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Overall Evaluation of MC-DBS

	Parallel DBS (SD, SN)	Distributed DBS	Federated DBS	Client/ Server DBS
High transaction rates	++	o/+	o	o
Intra-TA Parallelism	++	o/+	-/o	o/+
Extensibility	+	o/+	o	o
Availability	+/o	+	-	o
Distribution transparency	++	+	o	++
Geographical distribution	-	+	+	o
Node autonomy	-	o	+	-
DBMS heterogeneity	-	-	+	-/o
Administration	o	-	-/--	o

Trends in DBS

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Summary

- **Classification aspects:**
  - computer coupling, spatial distribution**,** attachment of disks
  - integrated vs. federated and homogeneous vs. heterogeneous DBS
  - functional specialization vs. equalization
- **Manifold requirements** lead to **different MC-architecture types:**
  - parallel DBS**,** distributed DBS
  - federated DBS, client/server-DBS
- **Parallel DBS**
  - use of I/O and processing parallelism
  - local computer allocation enables
    - efficient communication
    - dynamic load balancing
    - efficient parallelism for processing complex queries
  - main architectures: shared everything, shared disk, shared nothing and hybrids
  - parallelism in general MC systems or dedicated back-end systems (DB server)

Summary (2)

- **Distributed DBS:**
  - goals
    - support of decentralized organizations,
    - high availability,
    - extensibility (new nodes),
    - cost effectiveness
  - Primary approach
    spatially distributed, integrated SN systems (global schema)
- **Federated MC-DBS**
  - loose (virtual) integration of independent ad heterogeneous DBs
  - preservation of a relatively high node autonomy
  - integration of data and functions,
    also possible via the Web (Web services)
- **Alternatives for the support of heterogeneous DBs:**
  - Data Warehouses (physical data integration)
  - application integration
- **MC-DBS having functional specialization**
  - e.g., client/server-DBS for object-oriented DBS
  - multi-level client/server architectures/Web information systems
  - use of dedicated hardware („database machines") widely failed:
    too little cost effectiveness, functionality, and flexibility

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

Further References

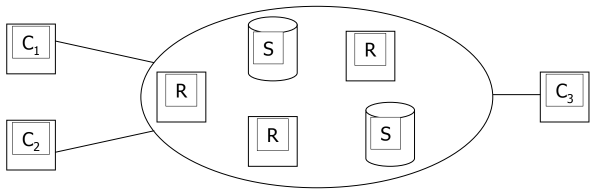
- *Braumandl, R., Keidl, M., Kemper, A., Kossmann, D., Kreutz, A., Seltzsam, S., Stocker, K. :* ObjectGlobe: Ubiquitous Query Processing on the Internet. The VLDB Journal 10:1, 48-71, 2001
- *DeWitt, D., Gray, J.:* Parallel Database Systems: The Future of High Performance Database Systems. Commun. ACM 35:6, 85-98, 1992
- *Özsu, T., Valduriez, P.:* Distributed and Parallel Database Systems. The Computer Science and Engineering Handbook 1997: 1093-1111
- *Özsu, T., Valduriez, P.:* Principles of Distributed Database Systems. Second Edition, Prentice-Hall, 1999
- *Rahm, E.:* Mehrrechner-Datenbanksysteme, Addison-Wesley, 1994
- *Sheth, A.P.; Larson, J.A.*: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Surveys 22:3, 183-236, 1990
- *Spruth, W., Rahm, E.:* Sysplex-Cluster-Technologien für Hochleistungs-Datenbanken. Datenbank-Spektrum 3, 16-26, 2002

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Decentralized – Self-organizing - Mobile

- **Which communication infrastructure** is required for future mobile systems?
- **So far: clear separation between end systems (clients), routers and servers**
  - applications running on clients
  - routers are dedicated systems to forward messages (statically configured)
  - client/server paradigm with servers within and outside of networks

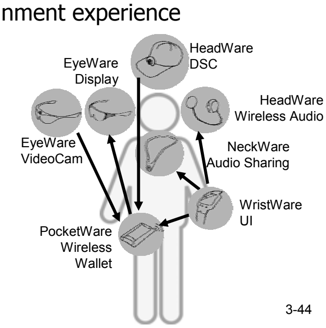


DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Example: Personal infotainment Companion (PiC)

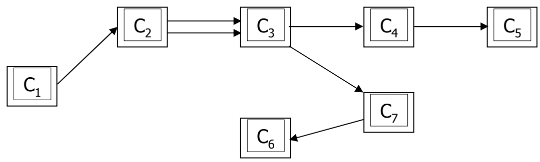
- **Trends**
  - Storage capacity growth
  - Wireless connectivity growth
  - Diversity of interconnected devices (stationary, mobile, sensors)
- **Need**
  - Distributed data management solutions

- **Example:** Personal infotainment Companion (PiC)
  - PiC is my personal audio-visual entertainment experience
  - PiC is my personal data wallet
  - PiC supports networking, cellular and broadcast functions
  - PiC is small, light, robust
  - PiC is flexible and future proof



Decentralized – Self-organizing – Mobile (2)

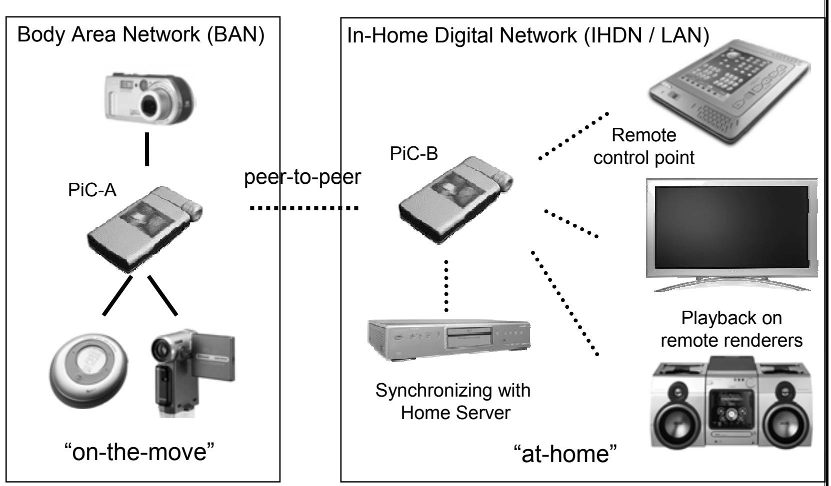
- **Decentralized, self-organizing infrastructures**
  - peer-to-peer paradigm
  - mobile devices integrate functions of the clients, the routers, as well as the servers
  - no permanent infrastructure needed: **spontaneous cross-linking**



- Examples of existing sensor-based DBMS
  - TinyDB defines a query language similar to SQL, with which data present at a sensor node can be queried
  - its implementation consists of a distributed query processor running under TinyOS on each node of the sensor network
  - COUGAR: The network is the database http://www.cs.cornell.edu/database/cougar/

© 2005 AG DBIS

Application Area



Goal: provide user with experience of having all digital content available at any time, in any place, regardless of connection availability

© 2005 AG DBIS

Decentralized – Self-organizing – Mobile (3)

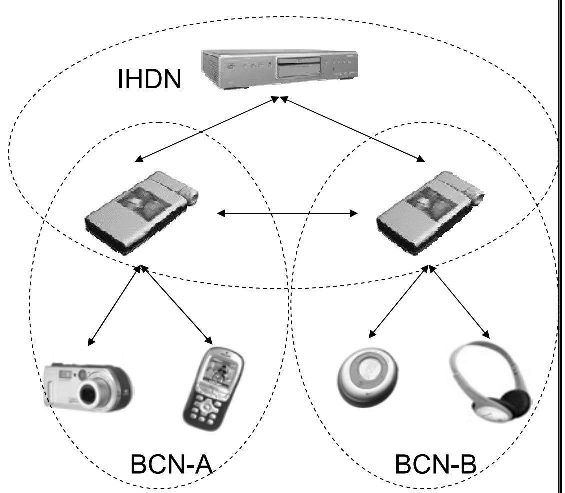
Which characteristics do future systems distinguish?

- **High mobility**
  - devices, services, and applications are mobile
  - significantly higher dynamics as compared to portable (nomadic) systems
  - interactions with conventional telephone network possible, but not presumed
- **Self-organization**
  - no fixed infrastructure (e.g. servers)
  - demand-oriented organization
- **Distributed, decentralized**
  no centralized infrastructure elements required
- **Heterogeneity**
  if needed, wide spectrum of devices and communication options

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

CE Storage Devices

Device Class	**Storage Capacity**
Stationary *(e.g. e-Hub or Home Server)*	100s GB
Smart Mobile *(e.g. Personal Infotainment Companion, Mobile Server)*	10s GB
Mobile Peripheral	10s-100s MB



Storage capacity asymmetry of 1-2 orders in magnitude!

IHDN = In-Home Digital Network
BCN = Body Centric Network

DBIS Datenbanken und Informationssysteme
© 2005 AG DBIS

Decentralized – Self-organizing – Mobile (4)

Which (new) services are required?

- **Base services**
  - application-, context-, service-dependent routing
  - resource management, load control, load distribution
  - advanced forms of communication (Anycast, Multicast)
- **Service platforms and higher-valued services**
  - location- and context-dependent services
  - migration of services
  - context-aware catalog and information services with high update frequencies – which structures are suitable?
- **Semantic interoperability**
  - for highly mobile communication partners
  - by regarding existing infrastructures
  - use of ontologies[1] essential

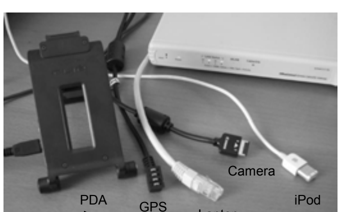
1 Originally, ontology is a philosophical discipline, now a buzzword of computer science. According to Meyer's Encyclopedic Lexicon, ontology is the "Lehre von dem Wesen und den Eigenschaften des Seienden". Within the computer science domain, it is the "formal specification of a distinct universe of discourse in the form of a system of terms".

© 2005 AG DBIS

Trends in DBS
Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

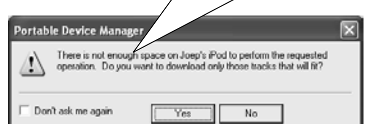
How does Mobile differ from Stationary?

- More severe design constraints
  - *power is considered the most important constraint in (mobile) embedded systems*
- Operates in highly dynamic environment
  - *mobile devices roam between different (networked) environments*
  - *connectivity will not always be available (and hence content is not always accessible)*
- Current synchronization experience

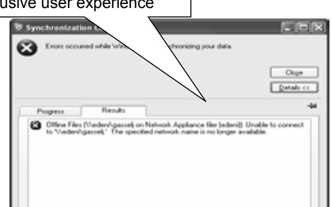


With power supplies it becomes even worse!

Currently Apple already significantly improved on this…



Synchronizing is a rather obtrusive user experience



Today's synchronization user experience is ***clumsy****,* ***obtrusive*** *and* ***not smart*** *enough ("dumb")!!*

© 2005 AG DBIS

Trends in DBS

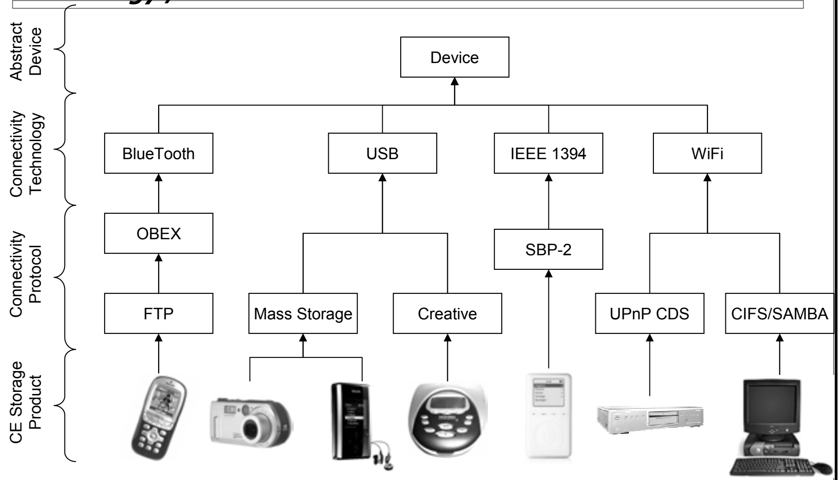
Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

DBIS Datenbanken und Informationssysteme

© 2005 AG DBIS

Proliferation of Storage Devices

Technology / Device Inheritance Tree



- Problem
  - Proliferation of device & technologies dramatically increases software development
- Solution
  - Move from *block/file level* to a common abstract *object-level*
  - Provide *adaptation layers* for different *devices and protocols*
  - Unify different device and service *discovery mechanisms*

Trends in DBS

Computer coupling
Spatial distribution
Classification
Integrated vs. federated DBS
Example: ObjectGlobe
Client/server DBS
Overall evaluation
Overflow

DBIS Datenbanken und Informationssysteme

© 2005 AG DBIS

Decentralized – Self-organizing – Mobile (5)

What are the new application areas?

- **E-Learning**
  - integration of mobile students
  - spontaneous formation of learning teams
- **E-Medicine**
  - higher mobility for invalids and therefore higher quality of life
  - sensors attached to the body (Body Area Networks)
    - control of health conditions
    - notification (alarm) in emergency situations using wireless infrastructure
- **Catastrophe situations**
  - spontaneous cross-linking of field helpers (coordination of helpers)
  - provision of context-related information
- **Traffic telematics**
- **Ubiquitous systems, "Ambient Internet" (4th gen.), ...**