DISTRIBUTED DATA MANAGEMENT

Prof. Fabio A. Schreiber
Dipartimento di Elettronica e Informazione
Politecnico di Milano
COMPANY TELECOMMUNICATIONS

1st STAGE (< ‘80)
• TELEPHONE (POTS), TELEX, DATA TRANSMISSION INDEPENDENT ON SEPARATE NETWORKS

2nd STAGE (‘80 - ‘95)
• DESIGN AND IMPLEMENTATION OF LARGE DIGITAL COMMUNICATION NETWORKS
• DEVELOPMENT OF LOCAL NETWORKS OF PCs AND WORKSTATIONS

3rd STAGE (> 1995)
• INTEGRATION AND MANAGEMENT OF LARGE HETEROGENEOUS WANs AND OF LOCAL NETWORKS
ACCESS TO REMOTE RESOURCES
connection to different computers and computing centers is made possible using the same terminal (e.g. TELNET, FTP, ... PROTOCOLS)

DISTRIBUTED COMPUTING
complex systems are built in which the application process uses several remote computers and/or data sets, through telecommunication networks (e.g. distributed information systems, HPC, ...)

TELEMATIC APPLICATIONS
- electronic mail
- teleconference
- ........
DATA MANAGEMENT ON THE NETWORK
DATA MANAGEMENT ON THE NETWORK: INDEPENDENT DATA BASES
DATA MANAGEMENT ON THE NETWORK: COMMON COMMAND LANGUAGE
DATA MANAGEMENT ON THE NETWORK: DISTRIBUTED DATA BASE
MODERN INFORMATION SYSTEMS

TELECOMMUNICATION NETWORKS BECOME AN ESSENTIAL COMPONENT FOR A GOOD ECONOMICAL AND FUNCTIONAL OPERATION OF THE ORGANIZATION

THE AVAILABILITY OF EFFECTIVE TELECOMMUNICATION SYSTEMS ALLOWS THE DEVELOPMENT OF NEW BUSINESS TYPES
VERTICAL FUNCTION PARTITIONING OF AN I.S.
HORIZONTAL FUNCTION PARTITIONING OF AN I.S.
DATA MANAGEMENT

1\textsuperscript{st} STAGE (< ‘70)
  • SPARSE FILES

2\textsuperscript{nd} STAGE (‘70 - ‘90)
  • LARGE CENTRALIZED DATA BASES

3\textsuperscript{rd} STAGE (> 1990)
  • DISTRIBUTED DATA MANAGEMENT
SYSTEM ARCHITECTURE: A DIALECTIC PROCESS

THE THESIS
A MAINFRAME IS THE BEST!

THE ANTITHESIS
10, 100, 1000 DECENTRALIZED MINICOMPUTERS

THE SYNTHESIS
DISTRIBUTED INFORMATICS FOR MAXIMAL FLEXIBILITY

MARX

HEGEL

EDP MANAGER
DISTRIBUTED DATA MANAGEMENT: FUNCTIONAL GOALS

- AVAILABILITY
- LOAD SHARING
- RESOURCE SHARING
- QUALITY OF SERVICE TO THE USER
FUNCTIONAL GOALS: AVAILABILITY

- REDUNDANT HW/SW RESOURCES CAN BE USED TO OBTAIN AN OVERALL SYSTEM HIGHER AVAILABILITY
  - FAULT TOLERANT SYSTEMS
  - SOFT DEGRADATION SYSTEMS
FUNCTIONAL GOALS: LOAD SHARING

IT ALLOWS A BALANCED RESOURCES DEVELOPMENT
FUNCTIONAL GOALS: LOAD SHARING

IT ALLOWS A BALANCED RESOURCES DEVELOPMENT
DISTRIBUTED DATA MANAGEMENT: FUNCTIONAL GOALS

• **RESOURCE SHARING**
  SPECIALIZED OR UNIQUE RESOURCES CAN BE SHARED AT WHICHEVER NODE

• **QUALITY OF SERVICE IMPROVEMENT**
  – LOCAL PROCESSING CAPABILITIES
  – RESPONSE TIME REDUCTION
  – USER FRIENDLY INTERFACE
DISTRIBUTED DATA MANAGEMENT: ECONOMICAL AND ORGANIZATIONAL FACTORS

• THE PROS
  – LOCAL SYSTEMS EFFECTIVENESS ALLOWS A TIGHTER CONNECTION BETWEEN USERS AND SYSTEM
  – DISTRIBUTION OF THE “POWER” (ORGANIZATIONAL, PSYCHOLOGICAL, POLITICAL, …) ASSOCIATED TO INFORMATION OWNING
  – ORGANIZATIONAL ACTIVITIES INTEGRATION IN GEOGRAPHICALLY DISTRIBUTED COMPANIES
  – COST/BENEFIT ???

• THE CONS
  – THE GENERAL COORDINATION AND COLLABORATION NEED REQUIRES A “CULTURAL ATTITUDE” NOT EASY TO FIND
  – COST/BENEFIT???
DISTRIBUTED SYSTEMS

SYSTEMS IN WHICH MESSAGE TRANSMISSION TIME IS NOT NEGLIGIBLE WITH RESPECT TO THE TIME BETWEEN TWO EVENTS IN A SINGLE PROCESS
# DISTRIBUTED DATA MANAGEMENT: ARCHITECTURES

<table>
<thead>
<tr>
<th>Hw/Sw ARCHITECTURE</th>
<th>DATA MANAGEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>narrow bandwidth loosely coupled systems</td>
<td>Distributed Data Base Management System (DDBMS)</td>
</tr>
<tr>
<td>geographically distributed computer networks</td>
<td></td>
</tr>
<tr>
<td>wide bandwidth loosely coupled systems</td>
<td>back-end processor</td>
</tr>
<tr>
<td>local networks, functionally distributed systems</td>
<td></td>
</tr>
<tr>
<td>wide bandwidth tightly coupled systems</td>
<td>database machine</td>
</tr>
<tr>
<td>multiprocessor systems, associative memories, ...</td>
<td></td>
</tr>
</tbody>
</table>
## DDSS Taxonomy

<table>
<thead>
<tr>
<th>System Type</th>
<th>The Global System Has Access To ...</th>
<th>Typical Local Nodes Are ...</th>
<th>Global DB Functionality</th>
<th>How Global Information Is Dealt With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed Database</td>
<td>DBMS Internal Functions</td>
<td>Homogeneous Data Bases</td>
<td>Y</td>
<td>Name Space and Global Schema</td>
</tr>
<tr>
<td>Federated Database with a Global Schema</td>
<td>DBMS User Interface</td>
<td>Heterogeneous Data Bases</td>
<td>Y</td>
<td>Global Schema</td>
</tr>
<tr>
<td>Federated Database</td>
<td>DBMS User Interface</td>
<td>Heterogeneous Data Bases</td>
<td>Y</td>
<td>Partial Global Schema</td>
</tr>
<tr>
<td>Multidatabase with a System Language</td>
<td>DBMS User Interface</td>
<td>Heterogeneous Data Bases</td>
<td>Y</td>
<td>Access Functions in the Language</td>
</tr>
<tr>
<td>Homogeneous Multidatabase with a System Language</td>
<td>DBMS User Interface and Some Internal Functions</td>
<td>Homogeneous Data Bases</td>
<td>Y</td>
<td>Access Functions in the Language</td>
</tr>
<tr>
<td>Interoperable Systems</td>
<td>Application Programs Above DBMS</td>
<td>Any Data Source Satisfying the Interface Protocol</td>
<td>N</td>
<td>Data Exchange</td>
</tr>
</tbody>
</table>
DISTRIBUTED DATA MANAGEMENT

• CLIENT-SERVER ARCHITECTURE
  – MANY CLIENTS REFER TO A SINGLE SERVER
  – MAINLY USED FOR OLTP ON LOCAL NETWORKS

• DISTRIBUTED DATABASE
  – MANY CLIENTS REFER TO MANY SERVERS
  – MAINLY USED FOR OLTP ON LOCAL AND WIDE AREA NETWORKS

• DATA WAREHOUSE
  – DATA COLLECTION FROM MANY DIFFERENT DATA SOURCES
  – USED IN DSS ON LOCAL AND WIDE AREA NETWORKS
DDSS IN A CLIENT-SERVER ARCHITECTURE

- **THE CLIENT**
  - IS MANAGED BY THE APPLICATION PROGRAMMER
  - SHOWS A FRIENDLY INTERFACE TO THE FINAL USER
  - USES EITHER STATIC OR DYNAMIC SQL

- **THE SERVER**
  - IS MANAGED BY THE DATABASE ADMINISTRATOR
  - ITS DIMENSION DEPENDS ON THE WORKLOAD AND ON THE SERVICES TO BE DELIVERED
  - MANAGES THE OPTIMIZATION PROCEDURES
DDSS IN A CLIENT-SERVER ARCHITECTURE

USING DDBMS PRIMITIVES

CLIENT SITE

APPLICATION
PROGRAM

DATABASE ACCESS
PRIMITIVES

DDBMS₁

RESULTS

DDBMS₂

DATABASE

SERVER SITE
DDSS IN A CLIENT-SERVER ARCHITECTURE

USING AUXILIARY PROGRAMS AND RPC
DDSS IN A CLIENT-SERVER ARCHITECTURE

• USING DDBMS PRIMITIVES
  – THE DDBMS LOCAL COMPONENT ROUTES THE QUERY TO THE SERVER WHICH ACCESSES THE DATABASE AND SENDS BACK THE RESULTS
  – **HIGH DISTRIBUTION TRANSPARENCY** THANKS TO GLOBAL FILE NAMES
  – **LOW EFFICIENCY** SINCE THE ANSWER TRAVELS ONE TUPLE AT A TIME

• USING AUXILIARY PROGRAMS AND RPC
  – THE APPLICATION ASKS THE AUXILIARY PROGRAM TO EXECUTE ON THE SERVER AND TO SEND BACK THE RESULT
  – THE AUXILIARY PROGRAM ASSEMBLES TUPLES INTO RESULT SETS **IMPROVING TRANSMISSION EFFICIENCY**
DATA WAREHOUSE (DW)

• A TECHNIQUE FOR CORRECTLY ASSEMBLING AND MANAGING DATA COMING FROM DIFFERENT SOURCES TO OBTAIN A DETAILED VIEW OF AN ECONOMIC SYSTEM

• IT IS AN
  – INTEGRATED
  – PERMANENT
  – TIME VARIANT
  – TOPIC ORIENTED

  DATA COLLECTION TO SUPPORT MANAGERIAL DECISIONS

• IT IS THE SEPARATION ELEMENT BETWEEN OLTP AND DSS WORKLOADS
DW ARCHITECTURE

DATABASE 1

DATABASE 2

DATABASE LEGACY

SPARSE FILES

WAREHOUSE CONSTRUCTION

REPPLICATION AND PROPAGATION

DATA MART

DATA MART

DATA MART

COMPANY WAREHOUSE

1 + 1 = 3

INFORMATION ACCESS AND MANAGEMENT

KNOWLEDGE DISCOVERY AND DATA MINING

Gabio A. Schreiber

Distributed I.S. 29
MAIN PROBLEMS IN A DW

- VIEW AND METADATA MAINTENANCE
- REPLICATION MANAGEMENT
- CONSISTENCY MANAGEMENT
- APPLICATIONS IMPLEMENTATION
A DISTRIBUTED DATA BASE

IS A SET OF FILES, STORED IN DIFFERENT NODES OF A DISTRIBUTED SYSTEM, WHICH ARE LOGICALLY CORRELATED WITH FUNCTIONAL RELATIONSHIPS OR WHICH ARE REPLICA OF THE SAME FILE, IN SUCH A WAY AS TO CONSTITUTE A SINGLE DATA COLLECTION
A DISTRIBUTED DATA BASE ...

• **IS A DATA BASE**
  – AN INTEGRATED ACCESS MODE TO DATA MUST EXIST
  – IT MUST BE PROTECTED AGAINST INCONSISTENCIES AND FAILURES IN SUCH A WAY AS TO GUARANTEE DATA INTEGRITY

• **IS DISTRIBUTED**
  – PHYSICAL DATA DISTRIBUTION MUST BE TRANSPARENT TO THE END USER
SOME DESIGN PROBLEMS

• GENERAL SYSTEM ARCHITECTURE
  – DESIGN FROM SCRATCH
  – SYSTEM RESTRUCTURING
  – SYSTEM AND DATA HETEROGENEITY

• LOGICAL RELATIONS FRAGMENTATION

• REPLICATION AND ALLOCATION
  – HOW MANY COPIES AND WHERE

• ACCESS TO AND PROCESSING OF RELATIONS

• INTEGRITY AND PRIVACY

• RELIABILITY
DATA INDEPENDENCE

THE NOTION OF DATA INDEPENDENCE MUST BE EXTENDED TO ENCOMPASS THE FOLLOWING CASES

• **LOGICAL**
  THE DB ADMINISTRATOR NEEDS TO RESHAPE THE GLOBAL SCHEMA IN ORDER TO MEET THE REQUIREMENTS OF A VERY LARGE, HETEROGENEOUS AND DYNAMIC SET OF USERS

• **PHYSICAL**
  IMMUNITY TO (DYNAMIC) NETWORK CONFIGURATION CHANGES (SITES CONNECTION/DISCONNECTION)
INTEGRATED DDBMS

DDBMS

LDBMS

SE1

SE2

external schemas

global conceptual schema

fragmentation schema

site independent

location schema

SML 1

SML 2

SML m

SCL 1

SCL 2

SCL m

SIL 1

SIL 2

SIL m

local mapping schemas

local conceptual schemas

local internal schemas
FEDERATED DDSS

- External schemas:
  - SE1
  - SE2
  - SEn

- Global conceptual schema:
  - fragmentation schema
  - location schema

- Local mapping schemas:
  - SML 1
  - SML 2
  - SML m

- Local conceptual schemas:
  - SCL 1
  - SCL 2
  - SCL m

- Local internal schemas:
  - SIL 1
  - SIL 2
  - SIL m
GLOBAL SCHEMA DESIGN

• SIMILAR TO THE **VIEW INTEGRATION** PROBLEM
• STRUCTURAL CONFLICTS (different schemas)
• SEMANTIC CONFLICTS (even with similar schemas)
  – HOMONYMY (same name, different meaning)
  – AMBIGUITY (different name, same meaning)
  – FORMAT (NUMERIC ←→ ALPHANUMERIC, ...)
  – AGGREGATIONS (PART-OF, ...)
  – DATA OWN SEMANTIC (DIFFERENT MEASURE UNIT, DIFFERENT GRANULARITY, DIFFERENT JUDGEMENT, ...)
• UPDATING LIMITED TO **LOCAL ACTIVITY**
SEMANTIC CONFLICTS SOLUTION

- LOCAL DATA BASE RESTRUCTURING (NOT FEASIBLE)

- EXPLICIT ADDITION IN THE SCHEMAS OF SEMANTIC INFORMATION ABOUT DATA TO ALLOW APPLICATION PROGRAMS TO BEHAVE ACCORDINGLY
LOGICAL RELATIONS FRAGMENTATION

• HORIZONTAL
  – ALL THE FRAGMENTS SHARE THE SAME SCHEMA
  – TUPLES BELONG TO FRAGMENTS ACCORDING TO A SELECTION PREDICATE CORRESPONDING TO A DISTRIBUTION CRITERION

• VERTICAL
  – EACH FRAGMENT SCHEMA IS A PROJECTION OF THE GLOBAL RELATION SCHEMA
    • SCHEMAS WITH A NOT EMPTY INTERSECTION
    • DISJOINT SCHEMAS
LOGICAL RELATIONS FRAGMENTATION

HORIZONTAL

VERTICAL

DUPICATION
DATA REPLICATION

• PERMANENCE

– A COPY OF A DATA ELEMENT (RELATION OR FRAGMENT) IS PERMANENT IF IT EVOLVES IN TIME UNDER THE DDBMS MANAGEMENT

– IT IS TEMPORARY IF IT IS CREATED ONLY FOR SOME SPECIFIC OPERATION (IN A WORK AREA) AND THEN IT IS CANCELLED OR REFRESHED, ON DEMAND, FROM THE MASTER COPY
DATA REPLICATION

• CONSISTENCY

  – STRONG CONSISTENCY
    AT EVERY INSTANT EACH COPY OF EACH DATA
    ELEMENT MUST HAVE IDENTICAL VALUES

  – WEAK CONSISTENCY
    UPDATES MADE ON A COPY ARE PROPAGATED TO THE
    OTHER COPIES LATER ON

  – INDEPENDENCE
    UPDATES ON DIFFERENT COPIES ARE UNCORRELATED
    (OFTEN USED WITH TEMPORARY COPIES)
FRAGMENTATION AND REPLICATION

GLOBAL RELATION

FRAGMENTS

PHYSICAL IMAGES

SITE 1

SITE 2

SITE 3

R

R1

R2

R3

R4

R1

R2

R3

R4

R1

R2

R3

R4
TRANSPARENCY LEVELS

• TO FRAGMENTATION
  – APPLICATION PROGRAMS REFER TO GLOBAL RELATIONS AND IGNORE FRAGMENTATION

• TO LOCATION
  – APPLICATION PROGRAMS ARE INDEPENDENT OF REPLICAION AND OF PHYSICAL DATA LOCATION, BUT THEY PERCEIVE THE CHANGES IN THE FRAGMENTATION SCHEMA

• TO LOCAL MAPPING
  – APPLICATION PROGRAMS USE THE OBJECTS (DATA FRAGMENTS OR ACCESS PRIMITIVES) GLOBAL NAMES, BUT THEY MUST SPECIFY THE LOCATION SITE

• NO TRANSPARENCY
  – THE APPLICATION PROGRAMMER MUST WRITE THE ACCESS MODULE FOR EACH DBMS, WHICH ONLY ACTIVATE THE REMOTE MODULES
TRANSPARENCY LEVELS: QUERIES
fragmentation transparency

read (terminal,$SNUM);
select NAME into $NAME
from SUPPLIER
where SNUM=$SNUM;
write (terminal, $SNAME)
TRANSPARENCY LEVELS: QUERIES
location transparency

read (terminal, $SNUM);
select NAME into $NAME
from SUPPLIER_1
where SNUM=$SNUM;
if not #FOUND then
select NAME into $NAME
from SUPPLIER_2
where SNUM=$SNUM;
write (terminal, $SNAME)

©abio A. Schreiber
read (terminal,$SNUM);
  select NAME into $NAME
  from SUPPLIER_1 AT SITE_1
  where SNUM=$SNUM;
if not #FOUND then
  select NAME into $NAME
  from SUPPLIER_2 AT SITE_3
  where SNUM=$SNUM
write (terminal, $SNAME)
TRANSPARENCY LEVELS: QUERIES
no transparency

SUPPINQRY:
read (terminal, $SNUM);
   execute $SUPIMS ($SNUM, $FOUND, $NAME) at site 1;
if not $FOUND then
   execute $SUPCODASYL ($SNUM, $FOUND, $NAME) at site 3;
write (terminal, $NAME)

DDBMS

SUPCODASYL ($SNUM, FOUND, NAME)
find SUPPLIER_RECORD
***************

SUPIMS ($SNUM, FOUND, NAME)
get unique SUPPLIER_SEGMENT
**************

local DBMS (Codasyl)

DATABASE Codasyl

local DBMS (IMS)

DATABASE IMS
TRANSPARENCY LEVELS: UPDATES

• FOR FRAGMENTATION TRANSPARENCY
  – IF AN ATTRIBUTE BELONGING TO AN HORIZONTAL FRAGMENTATION PREDICATE IS UPDATED THE TUPLE SHALL BE MOVED BETWEEN THE FRAGMENTS

• FOR LOCATION AND REPLICATION TRANSPARENCY
  – ALL THE COPIES MUST BE UPDATED SIMULTANEuosly
MEDIATORS

A MEDIATOR IS A SOFTWARE MODULE (AT ISO-OSI LEVEL 7) WHICH USES AN IMPLICIT KNOWLEDGE OF SOME DATA SETS OR SUBSETS TO CREATE KNOWLEDGE FOR A HIGHER APPLICATION LAYER (WIEDERHOLD)

ITS MAIN FUNCTION IS OBJECT FUSION

- TO GROUP INFORMATION ABOUT THE SAME ENTITY OF THE REAL WORLD
- TO REMOVE REDUNDANCIES AMONG DIFFERENT SOURCES
- TO SOLVE INCONSISTENCIES AMONG DIFFERENT SOURCES
MEDIATORS

DATA SOURCE → MEDIATOR → USER WORKSTATION

DATA → ABSTRACTION MECHANISM → KNOWLEDGE

FORMATTED QUERY ← QUERY

RAW ANSWERS ← USEFUL ANSWERS

TRIGGER EVENTS ← INSPECTIONS

©abio A. Schreiber
WRAPPER

TRANSLATE QUERIES IN ONE OR MORE COMMANDS/
QUERIES UNDERSTANDABLE BY THE SPECIFIC SOURCE

THEY CAN EXTEND THE QUERYING POWER OF A SPECIFIC
SOURCE

THEY CONVERT NATIVE FORMAT RESULTS TO A FORMAT
UNDERSTANDABLE BY THE APPLICATION

THEIR WRITING INVOLVES A LARGE IMPLEMENTATION
EFFORT, BUT SOME TOOLS CAN EASE THE TASK (e.g. THE
TSIMMIS TOOLKIT)
EXAMPLE OF SEMANTIC CONFLICT

CONTEXT C1:
- MONEY AMOUNTS IN ORIGINAL CURRENCY
- MONEY AMOUNTS SCALE 1:1 BUT FOR YEN WHICH IS SCALED 1:1000

<table>
<thead>
<tr>
<th>COMPANY</th>
<th>REVENUE</th>
<th>COUNTRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>1 000 000</td>
<td>USA</td>
</tr>
<tr>
<td>NTT</td>
<td>1 000 000</td>
<td>JPN</td>
</tr>
</tbody>
</table>

CONTEXT C2:
- MONEY AMOUNTS IN USD SCALE 1:1

<table>
<thead>
<tr>
<th>COMPANY</th>
<th>EXPENSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>1 500 000</td>
</tr>
<tr>
<td>NTT</td>
<td>5 000 000</td>
</tr>
</tbody>
</table>

source: C. H. GOH et Al.
AN EXAMPLE OF ARCHITECTURE WITH MEDIATORS (TSIMMIS)
NETWORK CATALOGUE

FULLY REPLICATED
- FAST ACCESS
- DISK OCCUPATION
- DATA CONSISTENCY

NO CATALOGUE AT ALL
- BROADCASTING NEED
- ACCESS OVERHEAD
NETWORK CATALOGUE

FULLY CENTRALIZED

- ACCESS BOTTLENECK
- DISK OCCUPATION
- DATA CONSISTENCY

PARTIALLY REPLICATED

- A GOOD COMPROMISE BETWEEN OVERHEAD AND ACCESS EFFICIENCY
OVERALL ARCHITECTURE

DATA TRANSMISSION NETWORK

LOCAL DB

CATALOGUE

LDBMS

INTERFACE

TRANSPORT

OPERATING SYSTEM
A MODEL FOR DISTRIBUTED AND HETEROGENEOUS INFORMATION SERVICES

- **Aplicazioni**
  - DW, EDI, INTRANET, ...

- **IS Architecture**
  - TSIMMIS, Workflow, CoopWARE, ...

- **IS Infrastructure**
  - CORBA, HTTP, ...

- **Data Support**
  - DBMS, browser HTML, ...

- **Telecommunications Support**
  - TCP/IP, ...