PERVASIVE DATA MANAGEMENT

REAL-TIME DATABASES (RTDB)

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

REAL-TIME DATABASE SYSTEMS

DATABASE SYSTEMS WHICH MANAGE

- **TIME CONSTRAINED DATA**
  - SPECIFIC TIME INTERVALS FOR DATA VALIDITY
  - VALIDITY TIME INTERVALS ARE MADE EXPLICIT

- **TIME CONSTRAINED TRANSACTIONS**
  - DEADLINES TO COMPLETE BY
  - PRESCRIBED EARLIEST/LATEST STARTING TIMES
  - PERIODIC INVOCATIONS
  - ……..

REAL-TIME DATABASE SYSTEMS

NOT

DATABASES WITH FAST RESPONSE TIME

REAL-TIME DATABASE SYSTEMS

NOT

DATABASES WHICH DEAL WITH TIME REPRESENTATION

CONSISTENCY IN RTDB

- TRANSACTION LOGICAL CONSISTENCY
  – AS IN TRADITIONAL DBMS
    - CORRECTNESS CRITERIA (SERIALIZABILITY, ...)
    - CONCURRENCY CONTROL (TWO PHASE LOCKING, …)
  – MAY BE RELAXED IN RTDB IN FAVOUR OF TEMPORAL CONSISTENCY

- DATA LOGICAL CONSISTENCY
  – AS IN TRADITIONAL DBMS
    - RANGE CONSTRAINTS
    - REFERENTIAL INTEGRITY
    - …….
  – MAY BE RELAXED IN RTDB ACCEPTING BOUNDED IMPRECISION

CONSISTENCY IN RTDB

- TRANSACTION TEMPORAL CONSISTENCY
  - TIMING
  - MODE
  - PREDICTABILITY
  - IMPRECISION

- DATA TEMPORAL CONSISTENCY
  - ABSOLUTE
  - RELATIVE

DATA TIME SEMANTICS

TIME CONSISTENT DATA

– **ABSOLUTE**

USED INDIVIDUAL DATA ARE WITHIN THEIR VALIDITY INTERVAL, i.e. THEY REFLECT THE TRUE STATE OF THE WORLD

– **RELATIVE**

UPDATE TIMES OF USED MULTIPLE DATA ITEMS FALL WITHIN SOME SPECIFIED TIME INTERVAL OF EACH OTHER, i.e. THE STATES OF THE REPRESENTED VARIABLES ARE TIME-COMPATIBLE

DATA TIME SEMANTICS

- **ABSOLUTE CONSISTENCY**
  - **SENSOR DATA**
  - **STOCK MARKET PRICES**
  - **LOCATIONS OF MOVING OBJECTS**

- **RELATIVE CONSISTENCY**
  - **TEMPERATURE AND PRESSURE DATA FOR A CHEMICAL REACTOR CONTROL SYSTEM**
  - **RELATIVE POSITIONS OF A SET OF MOVING OBJECTS**

REAL TIME REQUIREMENTS

- **SPEED**
  - **IS OFTEN A NECESSARY BUT NOT A SUFFICIENT CONDITION**

- **PREDICTABLY MEETING TIMING CONSTRAINTS**
  - **IS A SUFFICIENT CONDITION**

- **IN RTDB TRANSACTIONS ARE EQUIVALENT TO RT TASKS**

TRANSACTIONS PROPERTIES

ATOMICITY
- COMMIT PROTOCOLS
- ABORT-ROLLBACK-RESTART

CONSISTENCY
- CONCURRENCY CONTROL
- INTEGRITY CHECKS

ISOLATION
- CONCURRENCY CONTROL

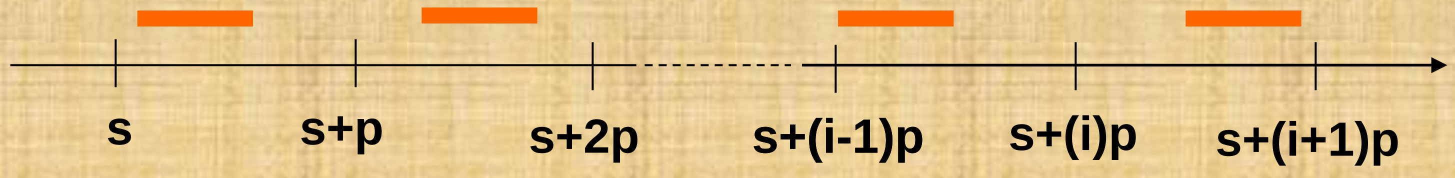
DURABILITY
- COMMIT PROTOCOLS
- RECOVERY MANAGEMENT

REAL-TIME TRANSACTIONS REQUIREMENTS
TIMING CONSTRAINTS

- **EARLIEST START TIME**
  - ABSOLUTE TIME BEFORE WHICH THE TRANSACTION MAY NOT START
- **LATEST START TIME**
  - ABSOLUTE TIME BEFORE WHICH THE TRANSACTION MUST START
- **DEADLINE**
  - ABSOLUTE TIME BEFORE WHICH THE TRANSACTION MUST END

REAL-TIME TRANSACTIONS REQUIREMENTS
TIMING CONSTRAINTS

- **PERIOD FRAME**
  - **DEFAULT EARLIEST START TIME AND DEADLINE FOR THE *i*th PERIODIC INSTANCE OF THE TRANSACTION**



TRANSACTIONS TYPES AND TIMING CONSTRAINTS

- **WRITE ONLY**

  GET STATE OF THE ENVIRONMENT (SENSORS) AND WRITE IT INTO THE DB

- **UPDATE**

  DERIVE NEW DATA AND STORE THEM IN THE DB

- **READ ONLY**

  READ DATA FROM THE DB AND SEND THEM TO ACTUATORS

TRANSACTIONS TYPES AND TIMING CONSTRAINTS

TIME CONSTRAINTS COMING FROM

- **TEMPORAL CONSISTENCY REQUIREMENTS**
  - PERIODICITY REQUIREMENTS

    EVERY 10 SEC SENSE GAS TEMPERATURE

- **REQUIREMENTS IMPOSED BY THE SYSTEM'S REACTION TIME**
  - DEADLINE CONSTRAINTS

    IF *TEMPERATURE* > *1000*

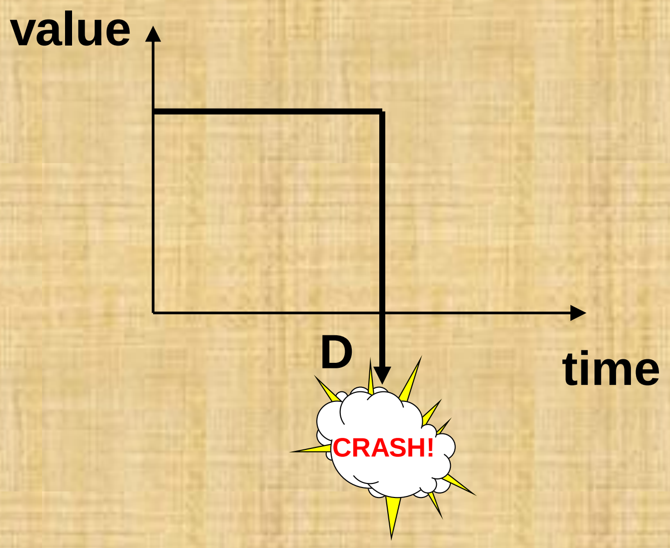
    WITHIN *5 SEC* CUT GAS

REAL-TIME TRANSACTIONS REQUIREMENTS
MODE CONSTRAINTS

- **HARD CONSTRAINTS**

  VIOLATION MEANS DISASTER

  - LIFE-CRITICAL SYSTEMS
  - CONTROL AND COMMAND SYSTEMS
  - ……..

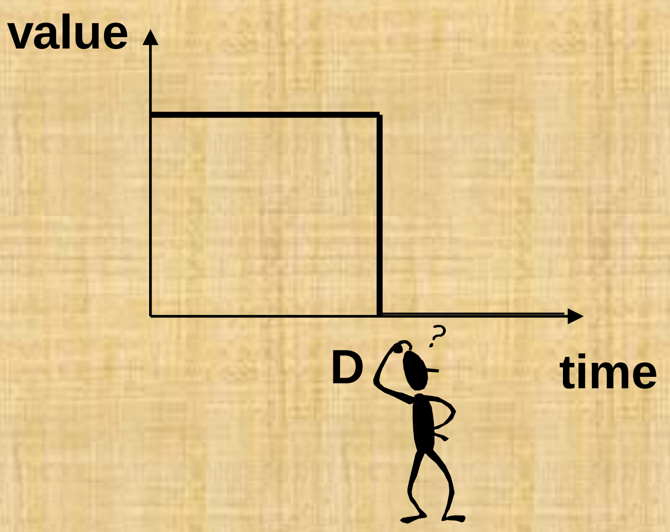


REAL-TIME TRANSACTIONS REQUIREMENTS
MODE CONSTRAINTS

- **FIRM CONSTRAINTS**

  VIOLATION MEANS NO VALUE

  – **FINANCIAL APPLICATIONS**

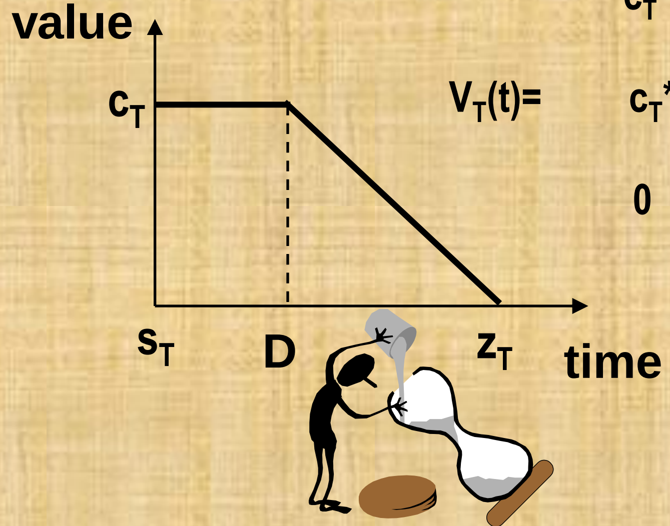


REAL-TIME TRANSACTIONS REQUIREMENTS
MODE CONSTRAINTS

- **SOFT CONSTRAINTS**

  VIOLATION MEANS DECREASING VALUE

  – **MULTIPLE STEPS INTERACTIVE TRANSACTIONS**



$$V_T(t) = \begin{cases} c_T & s_T \le t < d_T \\ c_T * (z_T - t)/(z_T - d_T) & d_T \le t < z_T \\ 0 & \text{otherwise} \end{cases}$$

REAL TIME TRANSACTIONS REQUIREMENTS
THREATS TO PREDICTABILITY

- **DEPENDENCE OF EXECUTION SEQUENCE ON DATA VALUES**
  - **AVOID RECURSIVE AND DYNAMIC DATA STRUCTURES**

- **DATA AND RESOURCE CONFLICTS**
  - **SPECIALIZED CONCURRENCY CONTROL PROTOCOLS**

- **DYNAMIC PAGING AND I/O**
  - **USE MAIN MEMORY DATA BASES**

REAL TIME TRANSACTIONS REQUIREMENTS THREATS TO PREDICTABILITY

- **TRANSACTIONS ABORTS AND ROLLBACKS/ RESTARTS**
  - **UNBOUNDED NUMBER OF ABORTS/RESTARTS**
  - **ABORTS DUE TO DEADLINE MISSES**
  - **BEGIN A TRANSACTION ONLY IF IT CAN BE COMMITTED WITHIN ITS DEADLINE**

REAL TIME TRANSACTIONS REQUIREMENTS
PREDICTABILITY

- **TWO PHASED TRANSACTIONS**
  - **PREFETCH**
    - **BRINGS DATA INTO MAIN MEMORY**
    - **DETERMINE COMPUTATIONAL NEEDS**
    - **PLAN THE EXECUTION RESPECTING CONFLICTS WITH OTHER GUARANTEED TRANSACTIONS**
  - **EXECUTION**
    - **STARTS ONLY IF THE PLAN IS FEASIBLE**
    - **LOCKS DATA**
    - **REPEATS PREFETCH IF DATA CHANGE IN THE MEANWHILE**

OVERHEAD OF THE PREFETCH PHASE

REAL TIME TRANSACTIONS REQUIREMENTS
PREDICTABILITY

ACCURATE ANALYSIS OF TIMING BEHAVIOUR

- **TRANSACTION SCHEDULING ALGORITHM**
- **TIME REQUEST FOR EACH RESOURCE**
- **WORST CASE ANALYSIS CAN BE TOO MUCH PESSIMISTIC OWING TO “LOGICAL” RESOURCES e.g. LOCKS ON DATA STRUCTURES AND BUFFERS**

REAL TIME TRANSACTIONS REQUIREMENTS IMPRECISION

- TIMING CONSTRAINTS ASK FOR A TRADE-OFF BETWEEN COMPUTATION TIME AND
  - COMPLETENESS

    A TRANSACTION SHOULD BE ALLOWED TO EXECUTE EVEN IF NOT ALL ITS ACTIONS MAY BE PERFORMED (ONLY THE MOST CRITICAL)
  - ACCURACY

    APPROXIMATE QUERY PROCESSING BY SAMPLING DATA
  - CONSISTENCY

    RELAXING SERIALIZABILITY ALLOWS MORE CONCURRENCY THEREBY IMPROVING PERFORMANCE
  - CURRENCY

    AVAILABLE OLDER VERSIONS OF DATA ITEMS CAN BE USED (e.g. IN EVALUATING TRENDS)

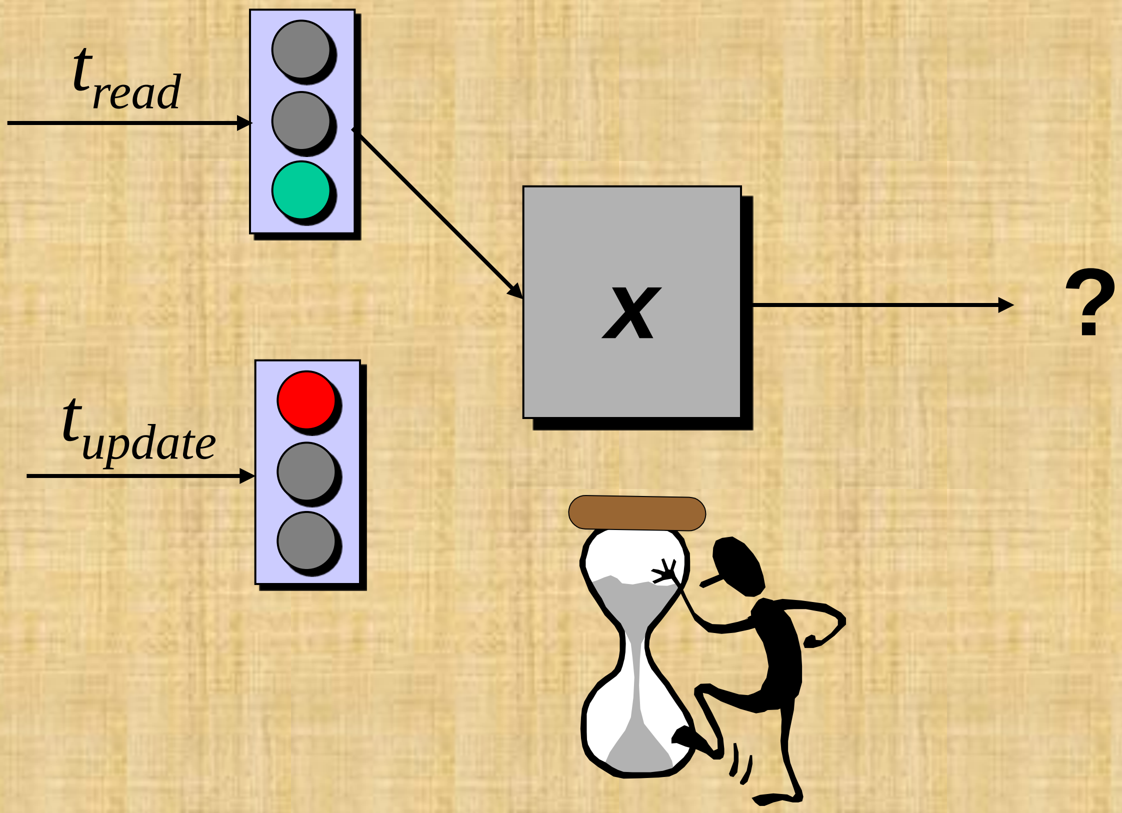
REAL TIME TRANSACTIONS REQUIREMENTS IMPRECISION

CONFLICTS RESULT AMONG THE FOUR FORMS OF CONSISTENCY CONSTRAINTS

PRECISION BOUNDS MUST BE SET FOR EACH TYPE OF CRITICAL RESULT

- A FEW METERS FOR A RADAR TRACKING SYSTEM
- A FEW CENTS FOR A STOCK EXCHANGE BROKER
- ……..

REAL TIME TRANSACTIONS REQUIREMENTS
IMPRECISION



REAL TIME TRANSACTIONS REQUIREMENTS IMPRECISION

- t_{update} SHOULD UPDATE DATA ITEM *X*
- t_{update} IS BLOCKED BY t_{read} READING *X*
- *X* IS GETTING *OLD*
- ALLOW t_{update} TO EXECUTE
  - VIOLATES *X* PRECISE LOGICAL CONSISTENCY
  - VIOLATES t_{read} PRECISE LOGICAL CONSISTENCY

REAL TIME TRANSACTIONS REQUIREMENTS IMPRECISION

PRIORITY INVERSION

- A LOWER PRIORITY TRANSACTION IS NOT PREEMPTED BY A HIGHER PRIORITY ONE
- IN THIS CASE LOGICAL CONSISTENCY IS PREFERRED TO TEMPORAL CONSISTENCY

IMPRECISION MIGHT ACCUMULATE

- THE SYSTEM MUST CONTROL AND MANAGE PRECISION BOUNDS

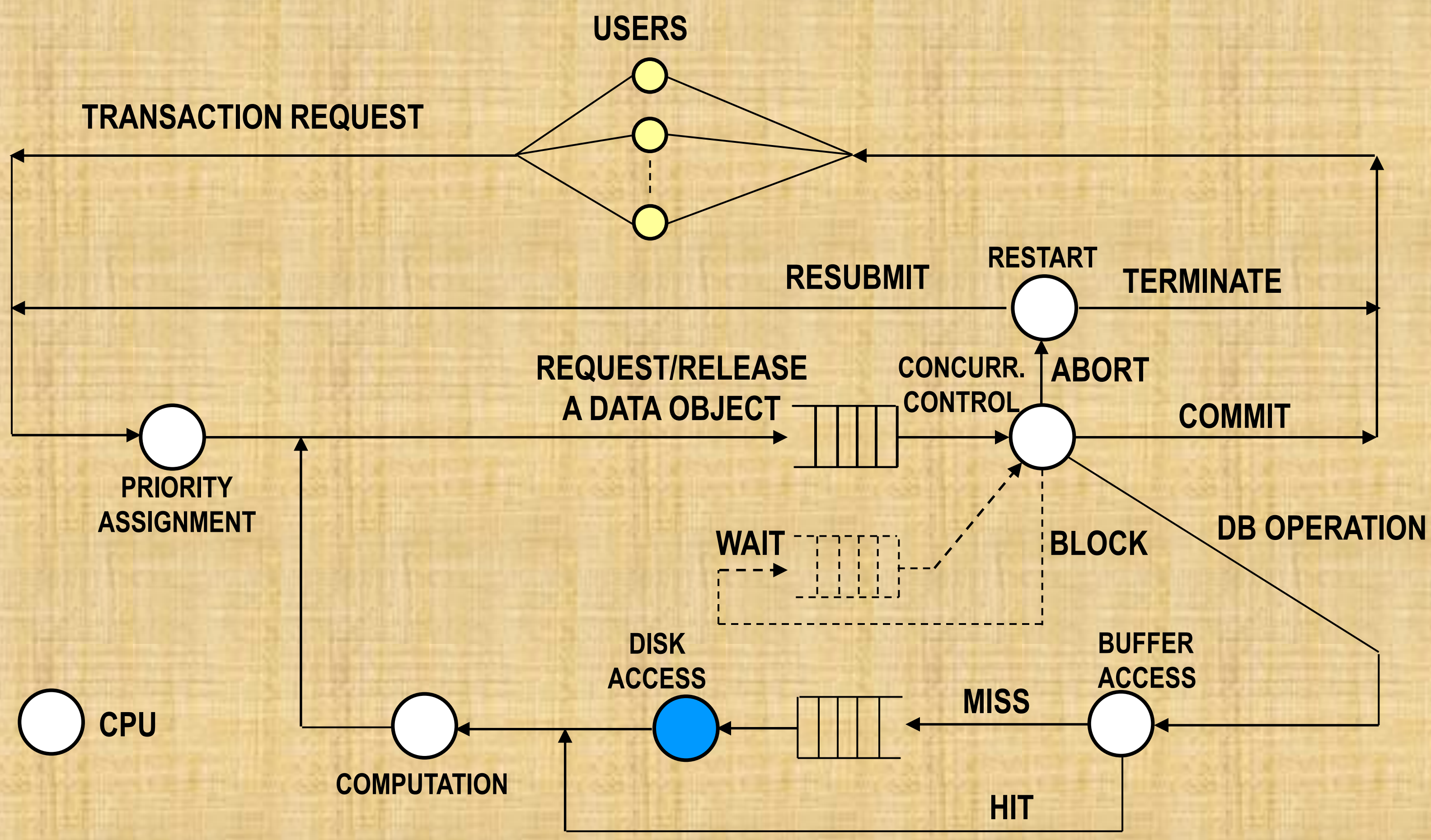
ISSUES IN AN RTDBMS

- **RESOURCES SCHEDULING**
  - **OVERLOAD MANAGEMENT**

- **BUFFER MANAGEMENT**

- **CONCURRENCY CONTROL**

A RTDB MODEL*



* from: J. A. Stankovic et Al.

RESOURCE (CPU) SCHEDULING

- **ASSIGNING A PRIORITY TO INCOMING TRANSACTIONS**
- **RUNNING TRANSACTIONS**
- **RESOLVING CONFLICTS**

RESOURCE (CPU) SCHEDULING HARD DEADLINES

APPROACHES SIMILAR TO REAL-TIME SYSTEMS, BUT WITH MANY RESTRICTIONS IN ORDER TO CHARACTERISE TRANSACTIONS A-PRIORI

- **INVOCATION TIME**
  - **PERIODIC TRANSACTIONS: AVAILABLE**
  - **APERIODIC TRANSACTIONS: *PERIOD* IS THE SMALLEST SEPARATION TIME BETWEEN TWO CONSECUTIVE INVOCATIONS**
- **WORST CASE EXECUTION TIME**
  - **POOR RESOURCES UTILISATION**

RESOURCE (CPU) SCHEDULING HARD DEADLINES

- **STATIC *TABLE-DRIVEN* SCHEDULERS**
  - RESERVE SPECIFIC TIME SLOTS FOR EACH TRANSACTION
  - TIME IN EXCESS CAN BE RECLAIMED FOR OTHER APPLICATIONS
  - VERY INFLEXIBLE APPROACH
- **PREEMPTIVE *PRIORITY-DRIVEN* APPROACH**
  - RATE-MONOTONIC PRIORITY ASSIGNMENT
  - SCHEDULABILITY ANALYSIS GIVEN THE PERIODS AND DATA REQUIREMENTS

RESOURCE (CPU) SCHEDULING SOFT DEADLINES

- **REAL-TIME SYSTEMS**
  - –MOST CRITICAL (HIGHEST-VALUE) FIRST (HVF) $P_T=1/c_T$
  - –EARLIEST DEADLINE FIRST (EDF) $P_T=d_T$
    - LIGHTLY TO MODERATELY LOADED SYSTEMS
    - FEWEST MISSED DEADLINES
- **REAL-TIME TRANSACTIONS**

  BOTH DEADLINE AND VALUE MUST BE CONSIDERED

  - –VALUE-INFLATED RELATIVE DEADLINE (VRD) $P_T=(d_T-s_T)/c_T$
    - OVERALL BEST PERFORMANCE UNDER MANY CONDITIONS

  AT COMMIT, PRIORITY IS RAISED AT THE HIGHEST VALUE AMONG THE ACTIVE TRANSACTIONS

RESOURCE (CPU) SCHEDULING OVERLOAD MANAGEMENT

IN HIGHLY LOADED SYSTEMS UNDER EDF HIGH PRIORITY CAN BE GIVEN TO TRANSACTIONS TOO CLOSE TO THEIR DEADLINES TO COMPLETE WITHIN DUE TIME.

THIS CAN PREVENT OTHER TRANSACTIONS TO MEET THEIR DEADLINES

- FLOW CONTROL SYSTEM TO ABORT EARLY TRANSACTIONS LIKELY TO MISS THEIR DEADLINES
- ADAPTIVE SCHEMES CREATE DIFFERENT CLASSES OF TRANSACTIONS WITH DIFFERENT SCHEDULING POLICIES

ACTIVE AND REAL TIME DATABASES

- **ACTIVE: APPLICATIONS REQUIRING AUTOMATIC SITUATION MONITORING AND NEED *TO REACT TO AN EVENT IN AN EFFICIENT AND AUTONOMOUS WAY***
- **REAL TIME: APPLICATIONS WHERE TRANSACTIONS NEED TO BE COMPLETED *WITHIN TIME CONSTRAINTS***
- **ACTIVE AND REAL TIME: ... NEED *TO REACT TO AN EVENT IN AN EFFICIENT AND AUTONOMOUS WAY WITHIN TIME CONSTRAINTS***

TYPICAL RULES FOR ARTDB's

DESIRABLE:

ON event E

IF condition C

DO "*complete*" action A "*within* t seconds"

SIMULATION ON MOST CURRENT SYSTEMS:

ON event E

IF condition C

DO action A "*within* t seconds"

THE TIME CONSTRAINT “within”

CAN REFER TO:

- TIME OF EVENT OCCURRENCE (t)
- TIME OF EVENT DETECTION (t+n)

- PROBLEM: DETECTING COMPOSITE EVENTS IN A REASONABLE TIME -> LARGE n!!!

NEW SEMANTIC ISSUES RAISED BY ARTDB's

- **COUPLING MODE**

  COMPOSITE EVENT DETECTION TOO TIME EXPENSIVE $\rightarrow$ COMPOSITE EVENTS NOT APPROPRIATE IN IMMEDIATE COUPLING MODE

- **COMPOSITE EVENTS**

  ASSOCIATE AS FEW RULES AS POSSIBLE TO THE SAME EVENT: EACH EVENT ASSOCIATED TO A SINGLE RULE

  TRICK (REACH): HIERARCHY OF LOGICAL EVENTS CORRESPONDING TO THE SAME PHYSICAL EVENT ($E \Longleftarrow E_1, \ldots, E_n$)

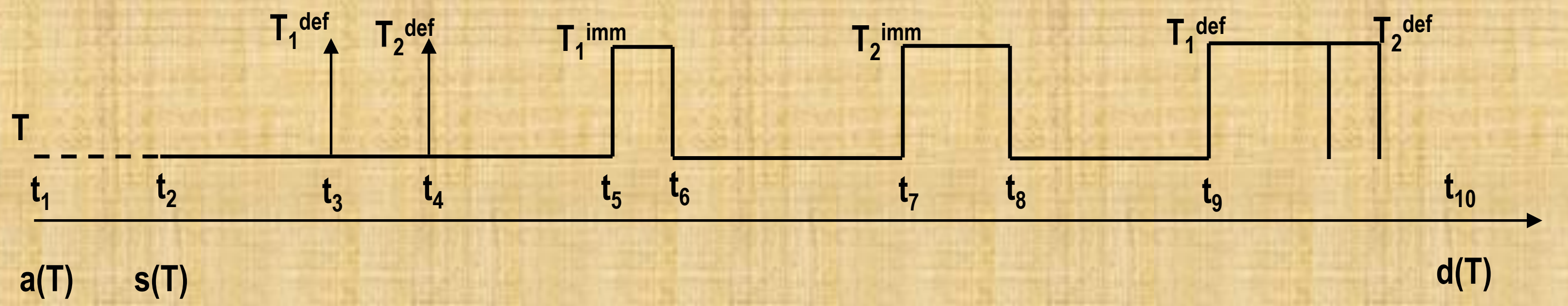
- **CONFLICT RESOLUTION**

  RULE SELECTION BASED ON RT REQUIREMENTS: HARD RT ACTIONS FIRED BEFORE FIRM AND SOFT ONES:

  HARD -> FIRM -> SOFT -> NOCONSTR

RESOURCE (CPU) SCHEDULING

PRIORITY ASSIGNMENT IN ACTIVE DATABASES



HOW TO ASSIGN PRIORITY TO A SUBTRANSACTION GIVEN THE PRIORITY OF THE PARENT TRANSACTION

- TRIGGERING TRANSACTIONS ARE LESS LIKELY TO COMPLETE SUCCESSFULLY THAN NON TRIGGERING ONES WITH THE SAME DEADLINE
- ASSIGN PRIORITY TO T_i^{def} AND TO T_i^{imm}
- DYNAMICALLY REASSIGN PRIORITY TO T

RESOURCE (CPU) SCHEDULING

PRIORITY ASSIGNMENT IN ACTIVE DATABASES (STATIC)

PRIORITIES ARE ASSIGNED TO TRIGGERED TRANSACTIONS WHEN THEY START AND THEY NEVER CHANGE DURING EXECUTION

- **PD (Parent Deadline)**
  - **SUBTRANSACTIONS ARE ASSIGNED A PRIORITY EQUAL TO THE DEADLINE OF THE PARENT WHENEVER THEY ARE TRIGGERED**

RESOURCE (CPU) SCHEDULING

PRIORITY ASSIGNMENT IN ACTIVE DATABASES (DYNAMIC)

- **DIV (DIViding parent's slack)**
  - IT USES THE ESTIMATES OF EXECUTION TIMES OF SUBTRANSACTIONS THAT HAVE ALREADY BEEN TRIGGERED
    - THE PARENT'S ESTIMATED EFFECTIVE SLACK IS DIVIDED EQUALLY AMONG THE CURRENT T^{imm}, THE T^{def} TRIGGERED PRIOR TO THE CURRENT TIME AND THE PARENT AND ADDED TO THE ESTIMATED COMPLETION TIME OF THE CURRENT T^{imm} TO GIVE ITS PRIORITY
    - THE PARENT'S PRIORITY IS ADJUSTED TO ACCOUNT FOR THE COMPLETION TIME OF THE SUBTRANSACTION

RESOURCE (CPU) SCHEDULING
PRIORITY ASSIGNMENT IN ACTIVE DATABASES (DYNAMIC)

- **SL (average case Slack)**
  - THE INITIAL VALUE OF SLACK IS BASED ON THE ESTIMATES OF THE REMAINING EXECUTION TIME FOR A TRANSACTION AND ITS SUBTRANSACTIONS.
  - THE SLACK IS THEN ADJUSTED AT EACH EVENT BASED ON WHETHER THE PARENT TRANSACTION TRIGGERS A SUBTRANSACTION OR NOT
  - TRIGGERED SUBTRANSACTIONS ARE GIVEN THE SAME SLACK AS THE PARENT

BUFFER MANAGEMENT

CONTENTION ARISE WHEN A TRANSACTION NEEDS BUFFER PAGES HELD BY OTHER ACTIVE TRANSACTIONS

$\{T_1, \ldots, T_k\}$	**k CONCURRENT TRANSACTIONS**
b_i	**BUFFER PAGE REQUIREMENT OF** T_i
N	**TOTAL NUMBER OF BUFFER PAGES**

CONTENTION FACTOR

$$f = \frac{\sum_{i=1}^{k} b_i}{N}$$

BUFFER MANAGEMENT POLICIES

NON REAL-TIME

- WAIT UNTIL IT GETS A FREE PAGE
- TIME-OUT IN THE ORDER OF MAGNITUDE OF EXECUTION TIME

ABORT

- ABORT A LOWER PRIORITY TRANSACTION
- CHOOSE THE ONE HOLDING THE MAX NUMBER OF PAGES
- BEST AT HIGH CONTENTION FACTORS

PRIORITY INHERITANCE

- INCREASE THE PRIORITY OF A LOWER PRIORITY TRANSACTION TO ITS OWN PRIORITY
- CHOOSE THE ONE CLOSEST TO ITS DEADLINE
- BEST AT LOW TO MEDIUM CONTENTION FACTORS

CHOOSE A MANAGEMENT POLICY WHOSE DECISIONS ARE BASED ON THE LEVEL OF CONTENTION

CONCURRENCY CONTROL

THE GOAL IS TO ALLOW TRANSACTIONS TO MEET THEIR DEADLINES WITHOUT REDUCING THE CONCURRENCY LEVEL IN ABSENCE OF *A PRIORI* INFORMATION

- **PESSIMISTIC (LOCK BASED) CC**
- **OPTIMISTIC (VALIDATION BASED) CC**

CONCURRENCY CONTROL

IF SPECIFIC SEMANTICS OF THE APPLICATION IS KNOWN, *AD HOC* CORRECTNESS CRITERIA CAN BE USED WHICH RELAX SERIALIZABILITY AND BOUND THE RESULTING IMPRECISION

- **TRANSACTION BASED**
  - **ALLOWS OTHERWISE FORBIDDEN TRANSACTIONS INTERLEAVING**
- **OBJECT BASED**
  - **ACCESS ON EACH OBJECT IS GRANTED BASED ON THE SEMANTICS OF THE OPERATIONS ON THE OBJECT**

CONCURRENCY CONTROL PESSIMISTIC

TWO-PHASE LOCKING HIGH-PRIORITY (2PL-HP)

IF

REQUESTOR’S PRIORITY > LOCK HOLDER’S PRIORITY

THEN

RESTART HOLDER AND GRANT LOCK TO REQUESTOR

ELSE

REQUESTOR WAITS FOR LOCK RELEASE

FI

IT PREVENTS DEADLOCKS IF PRIORITY VALUES GIVEN TO TRANSACTIONS ARE STATIC

CONCURRENCY CONTROL
PESSIMISTIC

OTHER VARIATIONS ON THE THEME OF 2PL-HP

- **TAKE INTO ACCOUNT DYNAMIC FACTORS SUCH AS WORKLOAD AND USE PRIORITY INHERITANCE (Hung & Lam H2PL)**

IF
REQUESTOR'S PRIORITY > LOCK HOLDER'S PRIORITY
THEN
BEGIN
PRIORITY(HOLDER):=PRIORITY(REQUESTOR)
REQUESTOR WAITS
END
ELSE
REQUESTOR WAITS FOR LOCK RELEASE
FI

- **ADAPT SERIALIZATION TO TRANSACTION PRIORITY (Lin & Son)**

CONCURRENCY CONTROL OPTIMISTIC

OPTIMISTIC CC WITH BROADCAST COMMIT

WHEN A TRANSACTION COMMITS IT CAUSES THE RESTART OF ALL THE CONFLICTING TRANSACTIONS

- **NO NEED TO CHECK WITH ALREADY COMMITTED TRANSACTIONS**
- **VALIDATING TRANSACTION IS CERTAIN TO COMMIT**
- **DETECTS CONFLICTS EARLIER THAN PURE OPTIMISTIC**
- **LESS WASTED RESOURCES AND EARLIER RESTARTS**
- **TRANSACTIONS PRIORITIES ARE NOT USED**

CONCURRENCY CONTROL OPTIMISTIC

OPTIMISTIC CC WITH PRIORITY WAIT

UPON REACHING VALIDATION, IF HIGHER PRIORITY TRANSACTIONS ARE FOUND IN THE CONFLICTING SET, THE TRANSACTION IS PUT IN A WAIT LINE

- WHILE WAITING, IT CAN BE RESTARTED BY A HIGHER PRIORITY COMMITTING TRANSACTION
- IF DEADLINE REACHED WHILE WAITING, IT IS ABORTED AND DISCARDED

RT-DBMS SUMMARY

STRONG INTERACTION AMONG

- RESOURCE SCHEDULING
- BUFFER MANAGEMENT
- CONCURRENCY CONTROL

- NEED TO SIMULATE
  - THE COMBINED ADOPTION OF DIFFERENT ALGORITHMS
  - THE BEHAVIOUR AT DIFFERENT WORKLOADS

RT-DBMS
WHAT HAS BEEN LEFT OUT

- **I/O AND DISK SCHEDULING**
  - **VERY GENERAL SOLUTIONS BELONGING TO THE RT-OS SPHERE (VARIANTS OF *SCAN*)**
  - **AVOID I/O SOLUTIONS BY CACHING OR STORING DATA IN MAIN MEMORY (MMDB)**
- **ABORT-ROLLBACK-RESTART**
  - **VERY PARTICULAR APPLICATION DEPENDENT RECOVERY/COMPENSATION TECHNIQUES**
  - **INVALIDATE DATA OF THE ABORTED EXECUTION CYCLE REFRESHING THEM AT THE NEXT CYCLE**
  - **JUST INFORM THE SYSTEM (THE OPERATOR) OF THE IMPOSSIBILITY TO COMMIT**

BIBLIOGRAPHY

- R.F.M. Aranha et Al. - Implementation of a Real-Time Database System - *Information Systems*, Vol. 21, n. 1, 1996, pp. 55-74
- L. Cingiser DiPippo, V.F. Wolfe - Real-Time Databases - Database Systems Handbook, Multiscience Press, 1997
- J.R. Haritsa et Al. - Value-Based Scheduling in Real-Time Database Systems - *VLDB Journal*, Vol 2, 1993, pp. 117-152
- K.Y. Lam et Al. – Concurrency Control in Mobile Distributed real-Time Database Systems – Information Systems, Vol. 5, n. 3, 2000, pp. 261-286
- K. Ramamritham, H. Son, L.C. Dipippo - Real-Time Databases and Data Services – *Real Time Systems Journal*, Vol. 28, 2004, pp. 179-215 http://rtdoc.cs.uri.edu/downloads/CP04-001.pdf
- R.M. Sivasankaran et Al. - Priority assignment in real-time active databases - The VLDB Journal, Vol. 5, n. 1, 1996, pp. 19-34
- J.A. Stankovic et Al. - Scheduling in Real-Time Transaction Systems - in A.M. van Tilborg, G. M. Koob "Foundations of Real-Time Computing: Scheduling and Resource Management", Kluwer 1991, pp.157-184
- J.A. Stankovic et Al. - Misconceptions About Real-Time Databases - *IEEE Computer*, Vol. 32, n. 6, 1999, pp. 29-36
- P.S. Yu et Al. - On Real-Time Databases: Concurrency Control and Scheduling - *Proc. of IEEE*, Vol. 82, n. 1, 1994, pp. 140-156