Performance Studies of Shared-Nothing Parallel Transaction Processing systems

Jie Li

Institute of Information Sciences & Electronics

University of Tsukuba, Tsukuba Science City

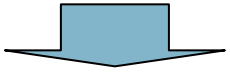
Japan

Contents

- Introduction
- Analytic Model
- Computer Simulation Study
- Scheduling Algorithms
- The WDPP (Wait-Depth Priority Protocol) CC Algorithm
- Concluding remarks and future work

Background

- Demands of parallel Transaction Processing (TP)
  - Traditional On-Line Transaction Processing (OLTP) applications: More and more users
    - Banking, Airline reservation ……
  - Internet-based electronic commerce: Accessible to anyone in the world
    - Home shopping, Home banking ……



- High performance parallel TP systems: No matter how much transactions volume grows, the systems should be able to be scaled to meet demands
  - To support thousands of processors and millions of device connections.
  - To scale throughput to accommodate high transaction processing rate.

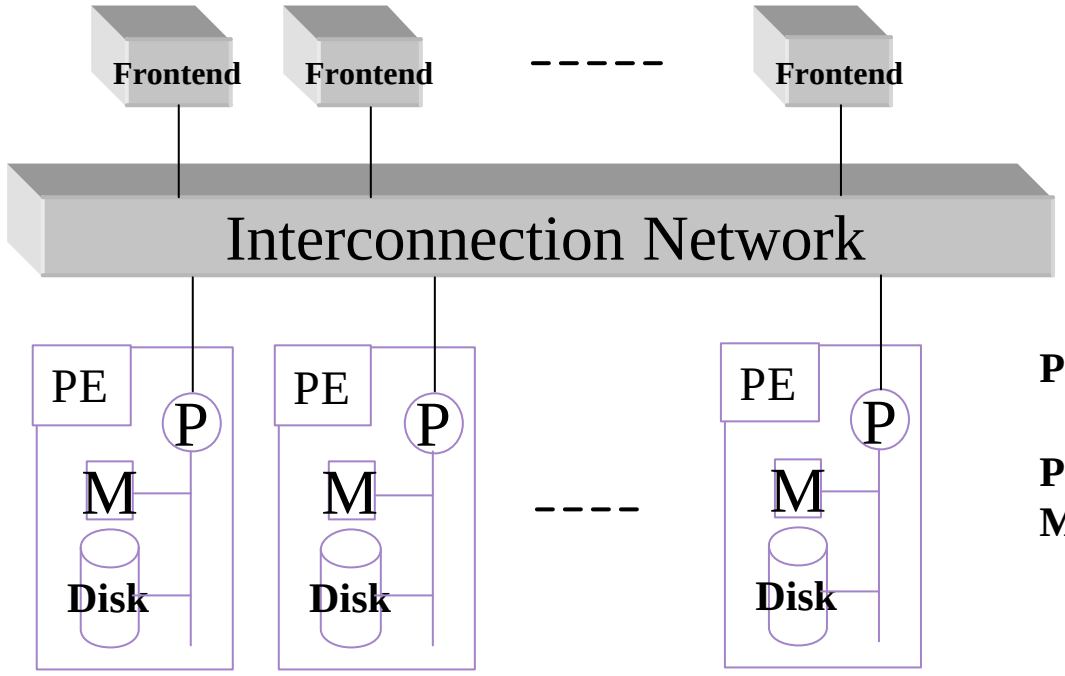
Two Performance Factors for TP systems

- The degree of contention for hardware resources
  - Hardware resource contention: Competition for hardware resources
  - Hardware resource requirement can be met cost-effectively with shared-nothing parallel TP architecture
- The degree of contention for data resources
  - Data contention: competition for data resources
  - Concurrency control (CC): Two-Phase Locking (2PL)
  - Scheduling Algorithms: FCFS (First-Come-First-Served), SCST (Synchronizing Completion of Sub-Transactions)

Performance Factors for Parallel TP

- The Degree of Declustering (DD)
  - Decluster a relation in a database, i.e., decide the number of nodes over which the relation should be declustered.
  - An essential factor that reflects the degree of parallelism.
- Two other factors
  - Additional overheads caused by the parallel processing.
  - Concurrency control (CC) algorithms, e.g., Two-Phase Locking (2PL).

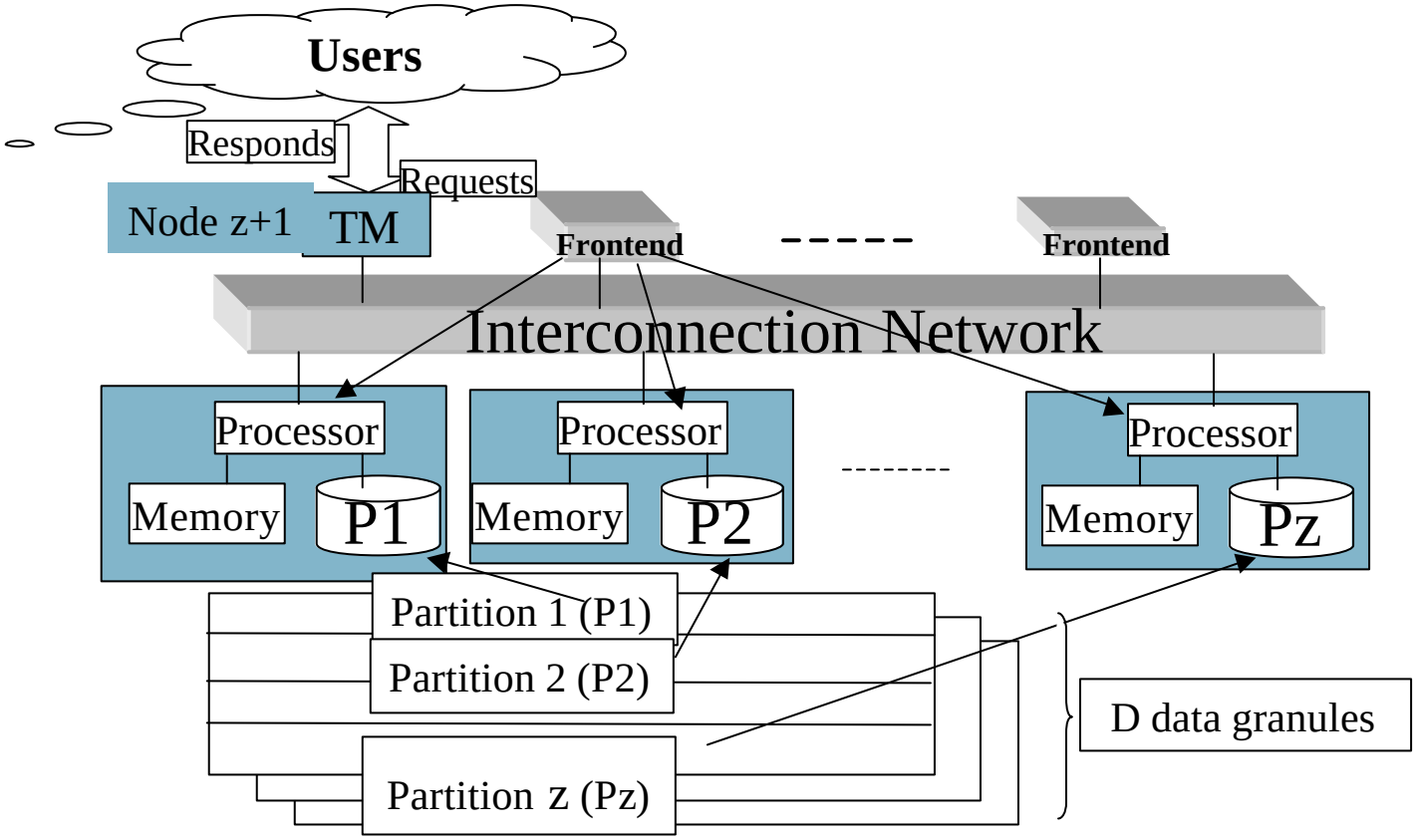
Shared-nothing parallel transaction (TP) Systems



PE: Processing Element
P: Processor
M: Memory

Examples: NonStop SQL of Tandem, Bubba of MCC, Gamma of the University of Wisconsin

Parallel transaction processing model



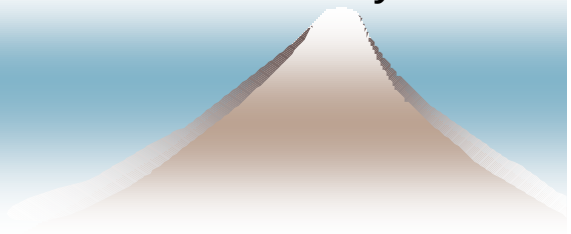
TM: Transaction Manager

Node z+1: management node

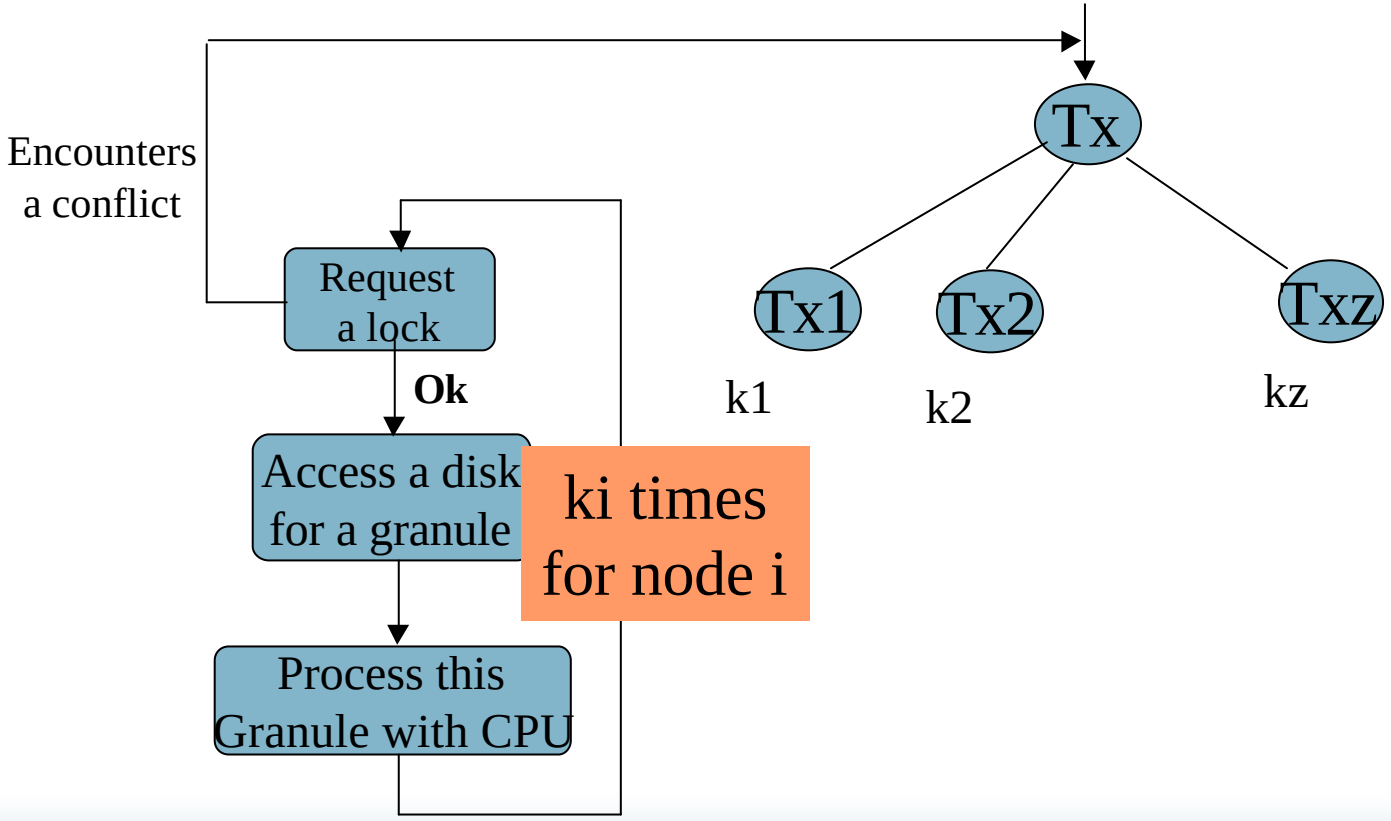
Node 1, 2, …, z: data processing node

System Description

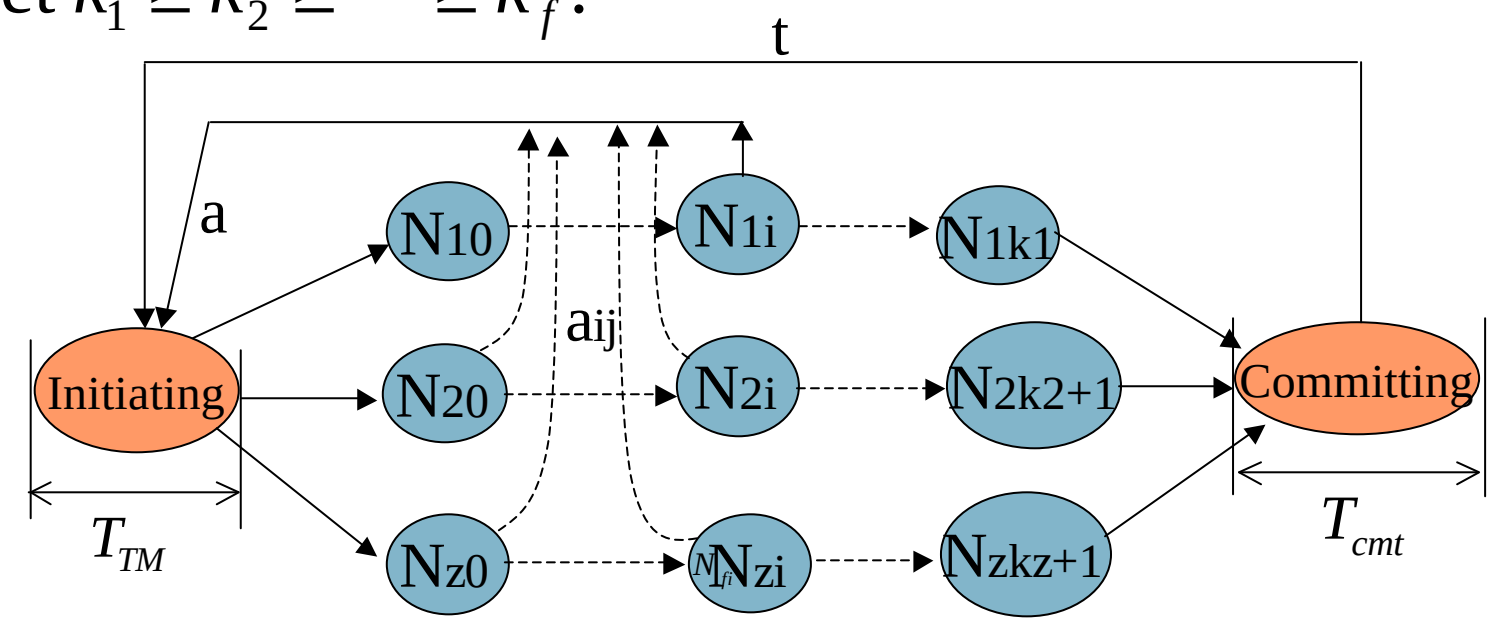
- Transactions arriving at the system are accepted and started by transaction managers that reside at the frontend nodes. The results are also routed by these frontend nodes to their users. Each frontend node has an identical copy of a global directory of a relation in the database telling which PE holds which data of this relation.
- A transaction consists of several sub-transactions, each of which works on the PE where the data to be used reside, and all of which are executed in parallel. Assume that sub-transactions do not need data from each other.
- A sub-transaction is modeled as a sequence of data-processing steps. The number of steps is called the size of the sub-transaction. Each step involves a lock request for the granule to be accessed, followed by the granule access to disk, and a period of CPU usage for processing this granule. The 2PL CC method is used to resolve data contention and to maintain data consistency and integrity. To ensure transaction atomicity, the 2PC is applied.



Flow diagram used to characterize the parallel TP systems with dynamic 2PL with no-waiting policy



Let $k_1 \geq k_2 \geq \cdots \geq k_f$.



Flow diagram for characterizing a parallel TP system

a: abort rate; t: throughput;

N_{fi}: the number of subtransactions holding i locks at node f;

N_{fk_f+1}: the number of subtransactions holding k_f locks and waiting for its siblings for 2PC;

T_{TM}: mean residence time of a transaction at initiating stage.

T_{cmt}: mean residence time of transactions at committing stage.

M: MPL.

An assumption

- If a subtransaction encounters a lock conflict, its siblings can be informed of the fact with little delay compared to the duration of a transaction step.
- Ex: Processor speed: 100MIPS
  Network bandwidth: 100Mbps
  Disk access: 13ms

  The duration of a transaction step (Cs) is at least 14 ms. A lock conflict message is broadcasted to all the related nodes from the node where the lock occurs. The message is assigned a high scheduling priority. The communication time delay (Cc) is at most 0.1 ms. Cc/Cs=0.0071 <<1.

The assumption is acceptable.

Basic equations (1)

- Lock conflict probability when requesting the *i+1*th lock at stage (f,i)

$$P_{f,i}^{c} = \begin{cases} \dfrac{G_f - i}{D_f - i} = \dfrac{G_f}{D_f}, & \text{for } f = 1,2,\ldots,Z,\ i = 0,1,\ldots,k_f - 1, \\ 0, & \text{otherwise.} \end{cases}$$

$$G_f = \sum_{j=1}^{k_f} (jN_{f,j}) + k_f N_{f,k_f+1} + k_f M_{cmt}$$

The number of locks held by the substransactions at node f.

Basic equations (2)

- MTM: number of transactions at the initiating stage.

$$M_{TM} = (a+t)T_{TM}, \quad M_{cmt} = tT_{cmt}, \text{ (Little's result)}$$

$$N_{f,i} = \begin{cases} c_{f,i}T, & \text{for } f = 1,2,...,Z, \ i = 0,1,...,k_f \\ c_{f,k_f+1}T_{f,k_f+1}, & \text{for } f = 1,2,...,Z, \ i = k_f + 1 \end{cases}$$

Basic equations (3)

- Mcmt: number of transactions at the committing stage.
- M_{DP}: number of Transactions at the data processing stages.

$$M = M_{TM} + M_{DP} + M_{cmt}$$

$$M_{DP} = M - T_{TM}(a+t) - T_{cmt}t$$

$$M_{DP} = \sum_{j=0}^{k_f+1} N_{f,j}, \quad \text{for } f = 1,2,...,Z$$

Basic equations (4)

- $a_{f,i}$: abort rate of subtransaction at stage (f,i).
- $c_{f,I}$: rate at which substransactions enter stage (f,i).
- $P_{f,i}^{a}$: abort probability of a substransaction at stage (f,i).

$$a_{f,i} = c_{f,i} P_{f,i}^{a} \quad \text{(Flow conservation law)}$$

$$c_{f,i} = c_{f,i-1} - a_{f,i-1} = c_{f,0} \prod_{j=0}^{i-1} (1 - P_{f,j}^{a}),$$

$$f = 1,2,...,Z, \; i = 1,2,...,k_f + 1$$

Basic equations (5)

- Number of subtransactions:

$$N_{f,i} = N_{f,0} \prod_{j=0}^{i-1} (1 - P_{f,j}^{a})$$

for $f = 1,2,\ldots, Z$, $i = 1,2,\ldots, k_f$

$N_{f,0}$ remains unsolved.

Basic equations (6)

$$N_{f,0} = \frac{M_{DP} - N_{f,k_f+1}}{\sum_{i=0}^{k_f} \prod_{j=0}^{i-1} (1 - P_{f,j}^{a})}$$

$$G_f = \sum_{i=1}^{k_f} (iN_{f,i}) + k_f N_{f,k_f+1} + k_f T_{cmt} t$$

$$= N_{f,0} \sum_{i=1}^{k_f} [i \prod_{j=0}^{i-1} (1 - P_{f,j}^{a})] + k_f N_{f,k_f+1} + k_f T_{cmt} t$$

$$P_f^c = \frac{G_f}{D_f}$$

N_{f,k_f+1} remains unsolved.

Solve N_{f,k_f+1}

$$N_{f,k_f+1} = c_{f,k_f+1} T_{f,k_f+1}$$

$$G_f = \sum_{i=1}^{k_f} (iN_{f,i}) + k_f N_{f,k_f+1} + k_f T_{cmt} t$$

$$= N_{f,0} \sum_{i=1}^{k_f} [i \prod_{j=0}^{i-1} (1 - P_{f,j}^a)] + k_f N_{f,k_f+1} + k_f T_{cmt} t$$

$$P_f^c = \frac{G_f}{D_f}$$

$P_{f,j}^a$ and T_{f,k_f+1} remain unsolved.

Let $e_{i,j} = \begin{cases} 1, & \text{for } i = 1,2,\ldots, Z,\ 0 \le j < k_i, \\ 0, & \text{otherwise.} \end{cases}$

$$T_{f,k_f+1} = T(k_1 - k_f)\prod_{i=1}^{f-1}\prod_{j=1}^{k_i-k_f-1}(1-P_i^c)$$

$$+T\sum_{j=1}^{k_1-k_f-1}\{j[\prod_{l=1}^{j-1}(1-e_{i,k_f+l}P_i^c)][1-\prod_{i=1}^{f-1}(1-e_{i,k_f+j}P_i^c)]\}$$

$$P_{f,i}^a = 1-\prod_{j=1}^{Z}(1-e_{j,i}P_j^c), \quad \text{for } f = 1,2,\ldots,Z, i = 0,1,\ldots,k_f.$$

Transaction throughput and abort rate

$$t = \frac{N_{1,k_1}}{T},$$

$$a = \sum_{i=0}^{k_1-1} a_{1,i} = \sum_{i=0}^{k_1-1} c_{1,i} P_{1,i}^{a}$$

$$= \frac{1}{T} \sum_{i=0}^{k_1-1} [N_{1,i} (1 - \prod_{j=1}^{Z} (1 - e_{j,i} P_j^{c}))].$$

Case study: no access skew (1)

- Given
  - K: transaction size
  - M:MPL
  - Z: number of processing nodes
  - D: database size
  - Times: T , T_{TM} , T_{cmt}
- Solve q, t, a, P_f^c.

Case study: no access skew (2)

- All the subtransactions have the same size (k).

$$P^{a}_{f,k} = P^{a}_{f,k+1} = 0, \;\; N_{f,k+1} = 0,$$

$$k = K / Z, \;\; D_f = D / Z,$$

$$P^{c}_{f_1} = P^{c}_{f_2} \;\; \text{for any two different nodes } f_1 \text{ and } f_2.$$

Case study: no access skew (3)

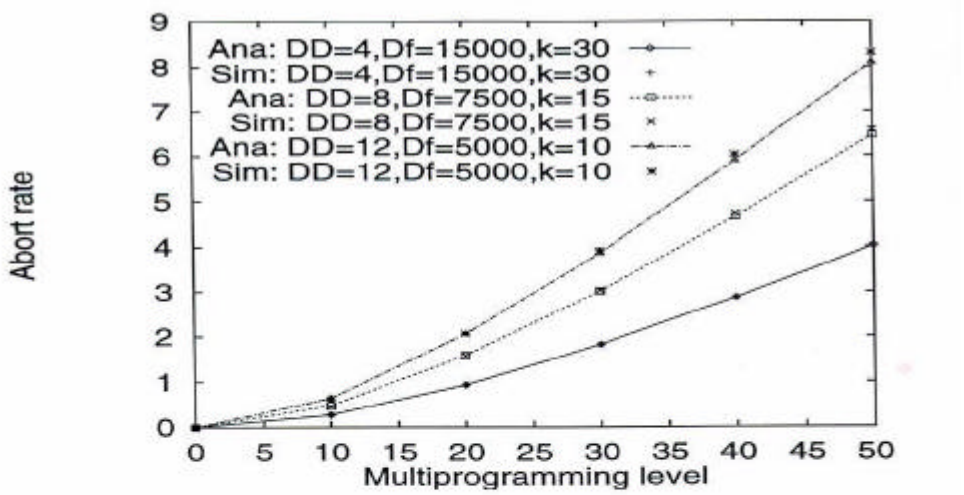
Let $P_f^c = P^c,\ (1-P^c) = q.$

$$\frac{\frac{M}{D_f}\left[(1-q^{k+1})T * \frac{\sum_{i=1}^{k} iq^i}{\sum_{i=0}^{k} q^i} + kT_{cmt}(1-q)q^k\right]}{(1-q^{k+1})T + (1-q)[T_{TM} + q^k T_{cmt}]} + q^{1/Z} - 1 = 0,$$

$$t = \frac{M(1-q)q^k}{(1-q^{k+1})T + (1-q)[T_{TM} + q^k T_{cmt}]},$$

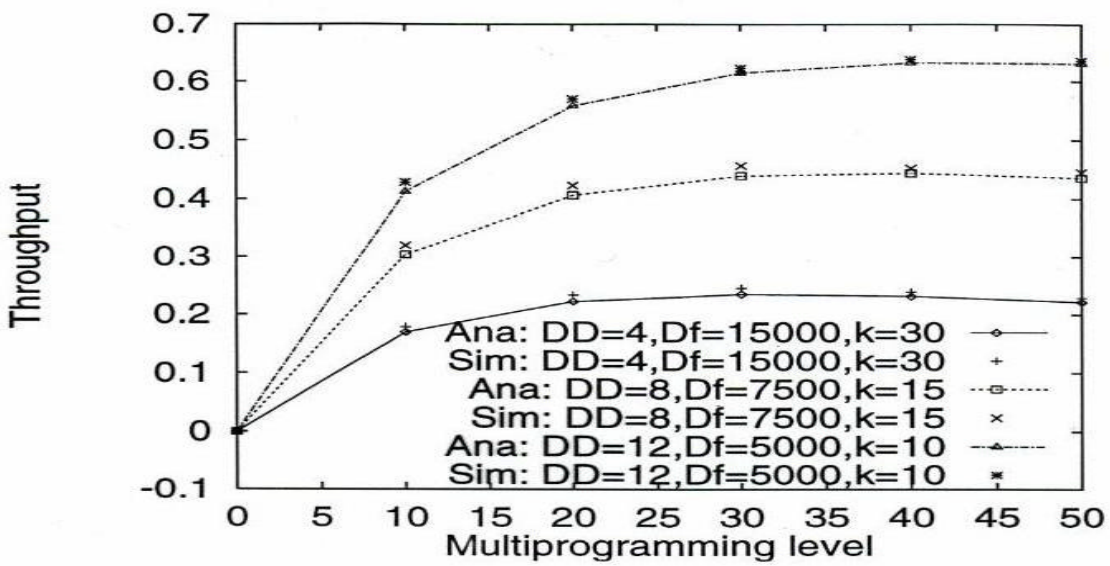
$$a = \frac{M(1-q)(1-q^k)}{(1-q^{k+1})T + (1-q)[T_{TM} + q^k T_{cmt}]}.$$

Analytic results – Abort rate



Note:
Ana: analysis;
Sim: simulation
D=60000,
K=120,
T=1.0,
T_{TM}=1.5,
T_{cmt}=2.0

Analytic results - Throughput



Note:
Ana: analysis;
Sim: simulation
D=60000,
K=120,
T=1.0,
T_{TM}=1.5,
T_{cmt}=2.0

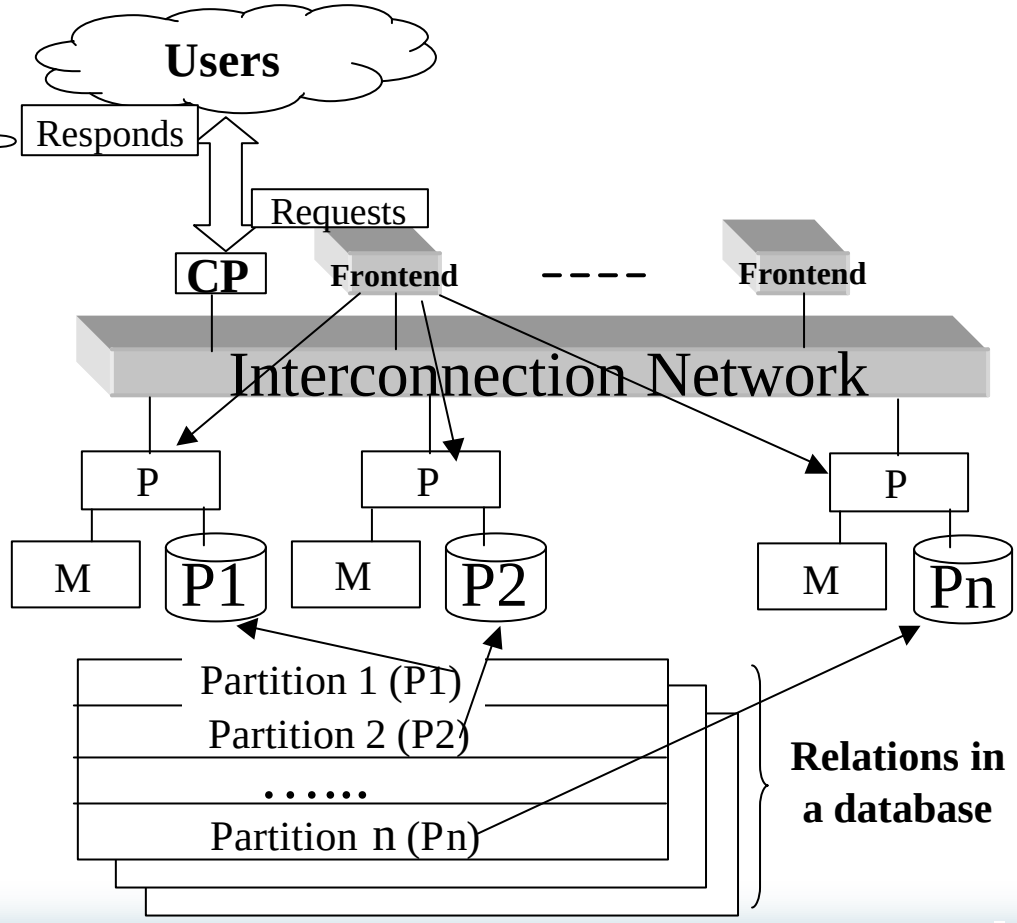
For the case with access skew

- The analytic results are available in the following reference:

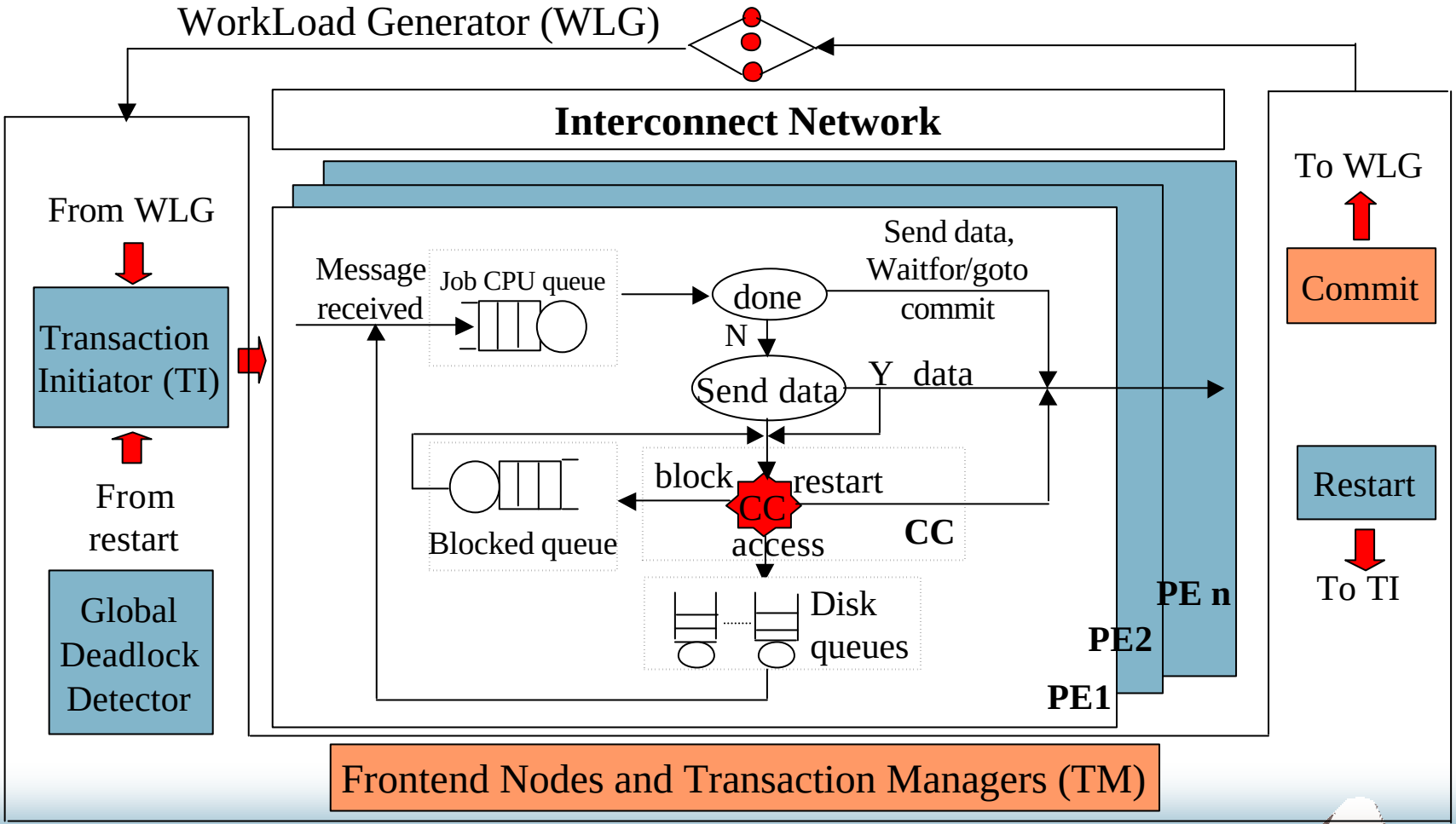
  J. Wang, J. Li, and H. Kameda, "Performance Study of Shared-Nothing Parallel Transaction Processing Systems", *Performance and Management of Complex Communication Networks, H. Hasegawa, H. Takagi, and Y. Takahashi (Editors)* , Chapman & Hall, Inc., pp. 154-172. 1998.

Computer simulation study for parallel OLTP systems

- **Computer system: PEs, interconnection network**
- **Database**
- **Transaction**
- **Transaction processing**



Model of the computer simulator



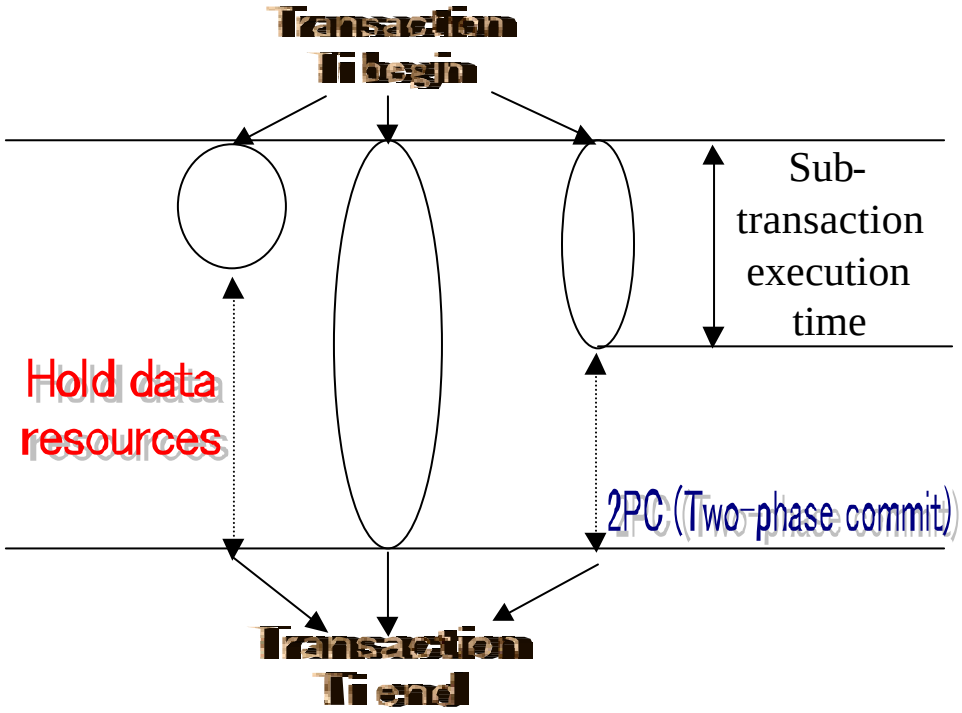
Structure of the simulator

Scheduling Algorithms:

- **FCFS (First-Come-First-Served): Conventional scheduling algorithm**

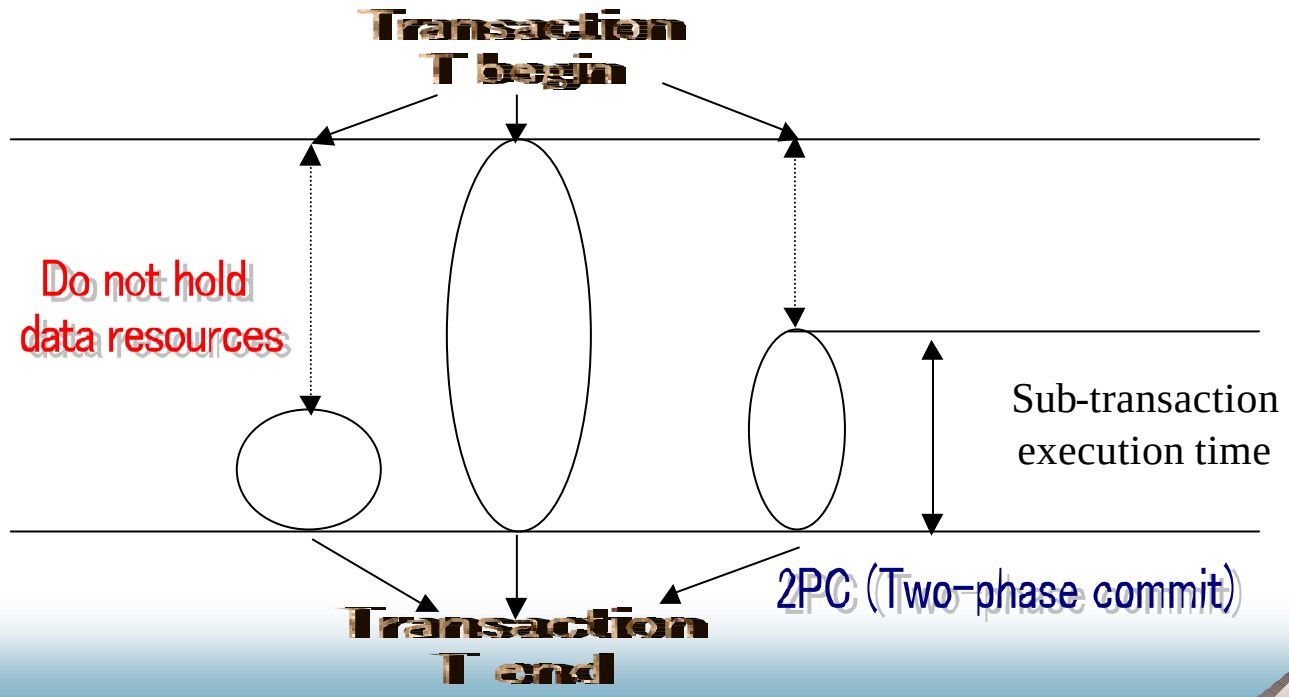
Access-skew: Each sub-transaction may not access the same number of data granules.

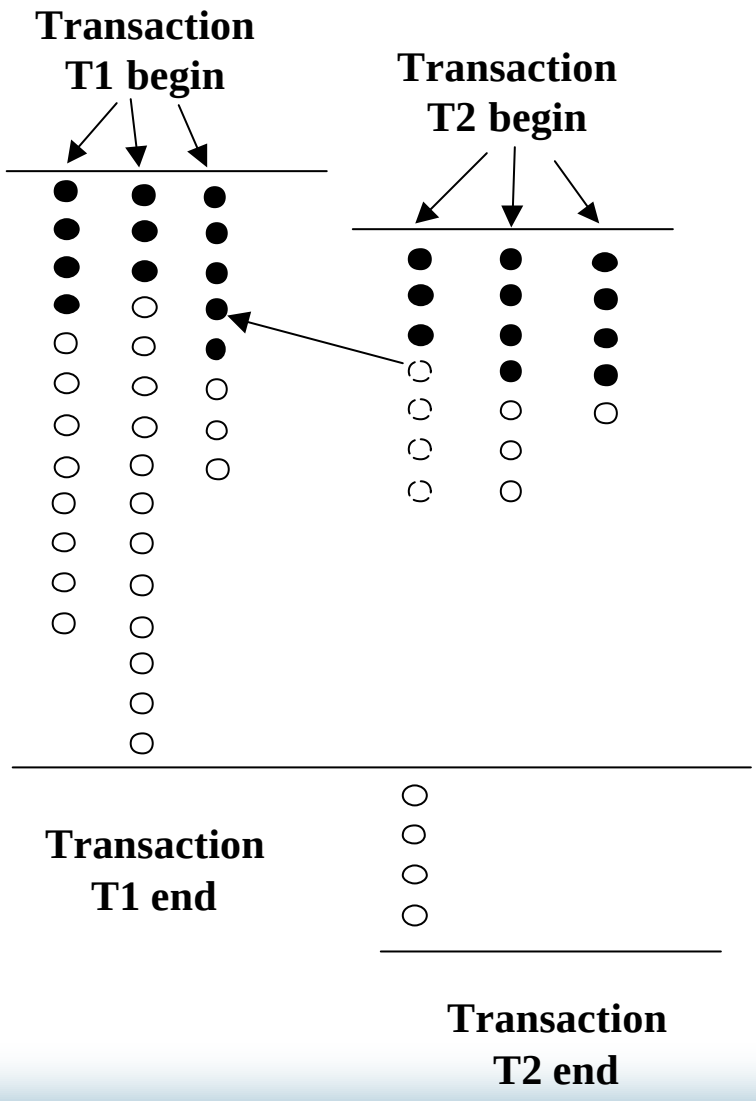
Data-processing-skew: Different PEs may have different data processing rates



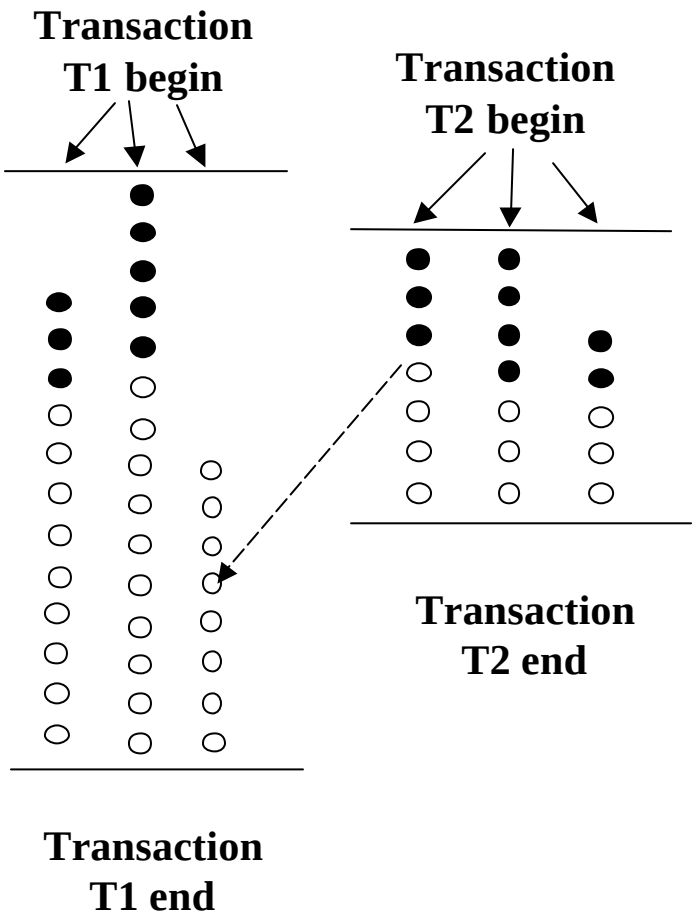
Scheduling Algorithms:

- **SCST (Synchronizing Completion of Sub-Transactions: Proposed scheduling algorithm**




Transaction
T1 begin
Transaction
T2 begin
Transaction
T1 end
Transaction
T2 end

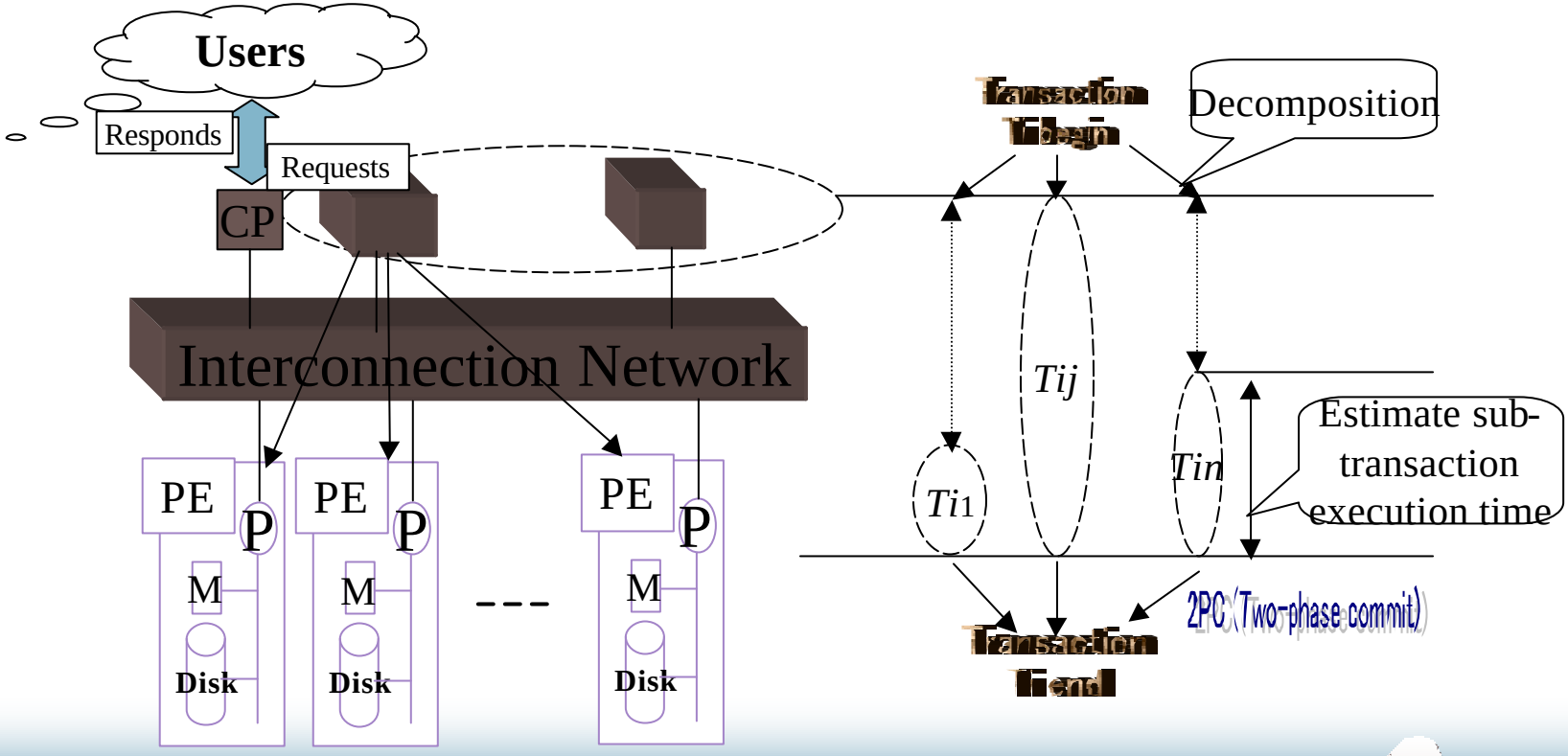
FCFS


Transaction
T1 begin
Transaction
T2 begin
Transaction
T2 end
Transaction
T1 end

SCST

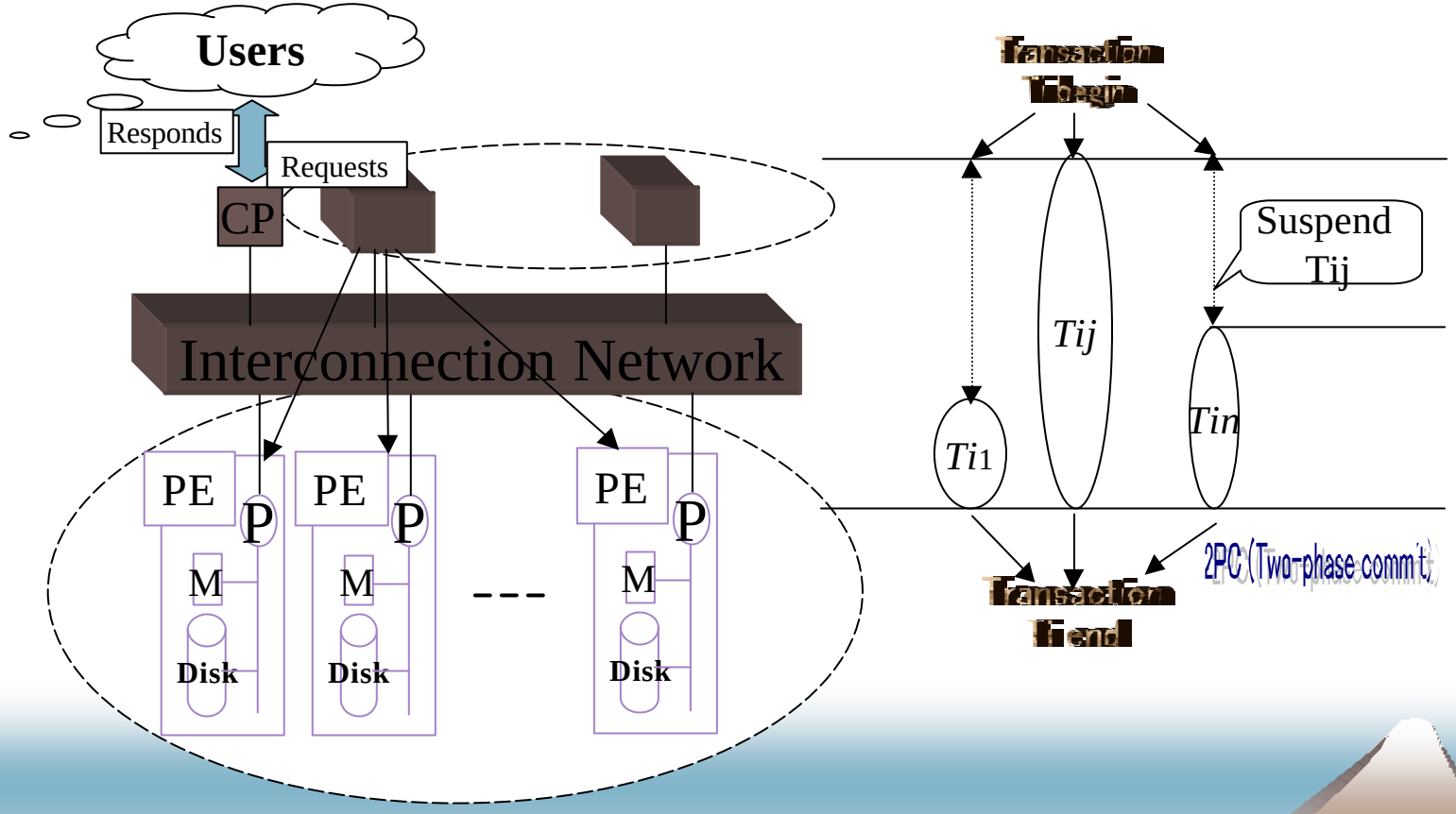
Implementation of the SCST Scheduling Algorithm

Part 1. In a frontend node



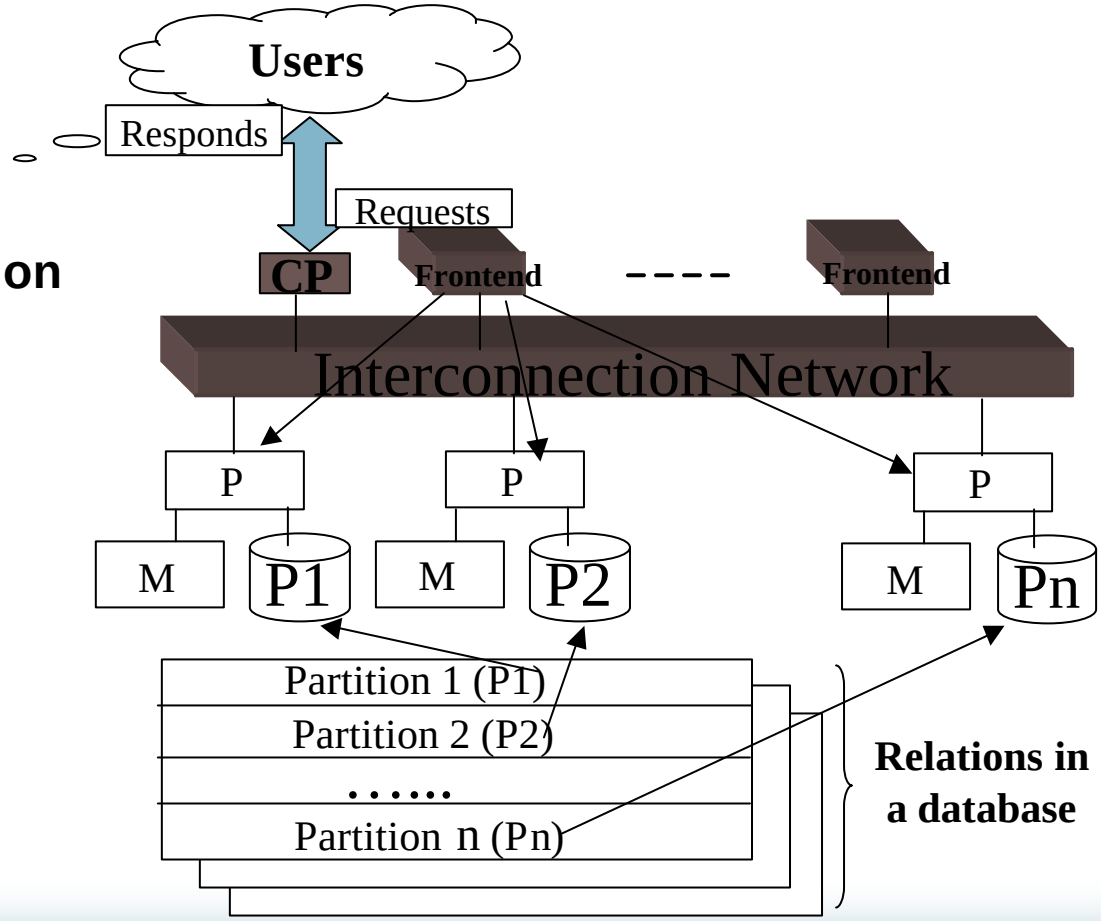
Implementation of the SCST Scheduling Algorithm

Part 2. In PEs



Developing a Simulator

- **Computer system: PEs, interconnection network**
- **Database**
- **Transaction**
- **Transaction processing**



Performance Study

Table 1: Computer system related parameters

Number of PEs in the system	32
Number of disks per PE	10
Network speed	17.76 Mbits/sec.
CPU time for sending/receiving a message	0.05 ms
Control/Data message size	512/4,096 bytes
I/O time for accessing a granule from disk	13.0ms
Hit ratio of database cache	60%

Table 2: Database related parameters

Number of relations in the database	8
Number of records per relation	320,000
Number of partitions per relation	32
Number of records per partition	10,000
Record size in byte	200
Lock granularity (record number per lock)	1

Performance Study

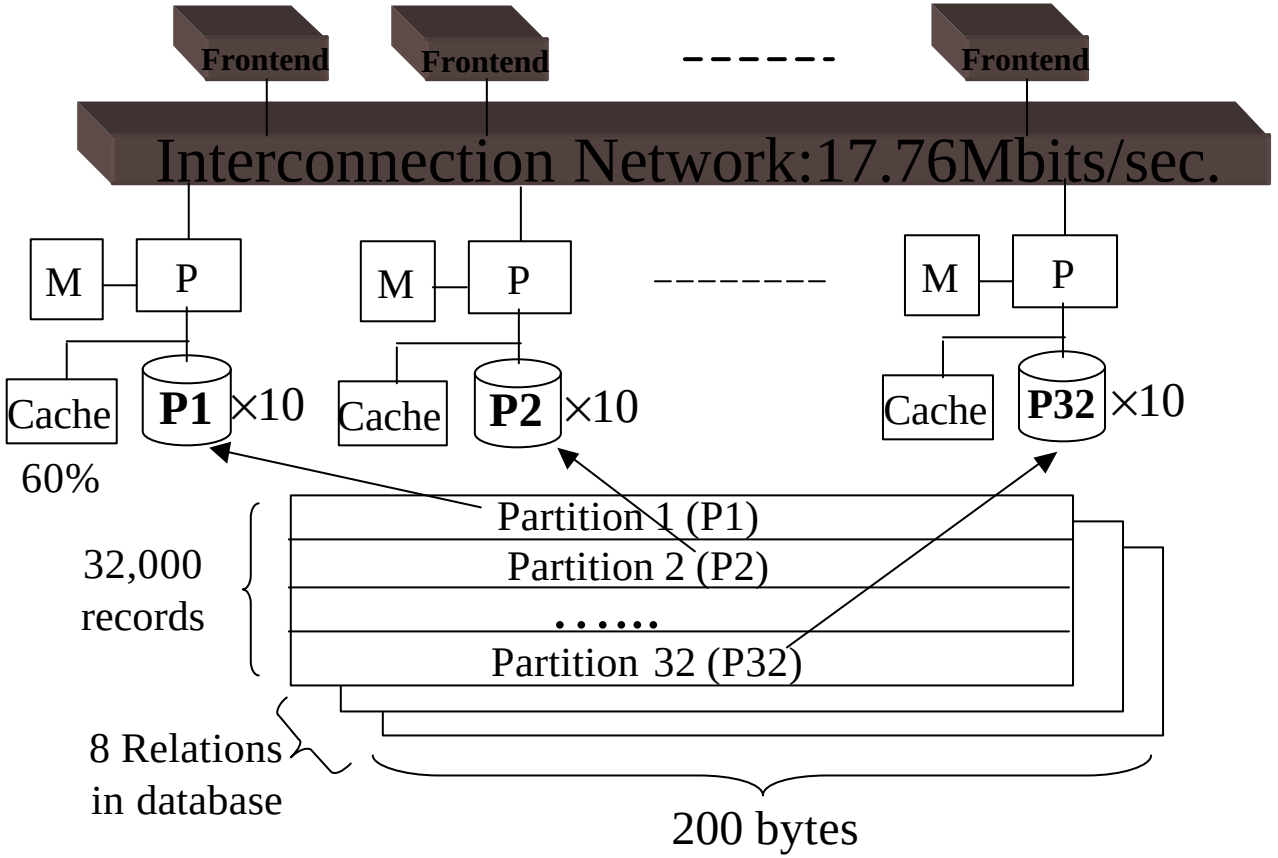
Table 3: Workload related parameters

Number of relations accessed by a transaction	1
Number of partitions accessed per relation	32
Number of granules accessed per partition, *Erlang* distribution	mean:6, dev: 0.3, 0.6
Number of transactions executed concurrently	MPL

Table 4: Transaction processing related parameters

Deadlock detection interval	100-500ms
CPU time for processing a granule	0.05ms
CPU time for (re-)starting a transaction at frontend nodes	1.0ms (0.5ms)
CPU time for (re-)intiating a sub-transaction at PEs	0.1ms (0.05ms)
CPU time for commit at frontend nodes/PEs	0.35/0.26ms
CPU time for abort at frontend nodes/PEs	0.12/0.10ms

Experiment parameters



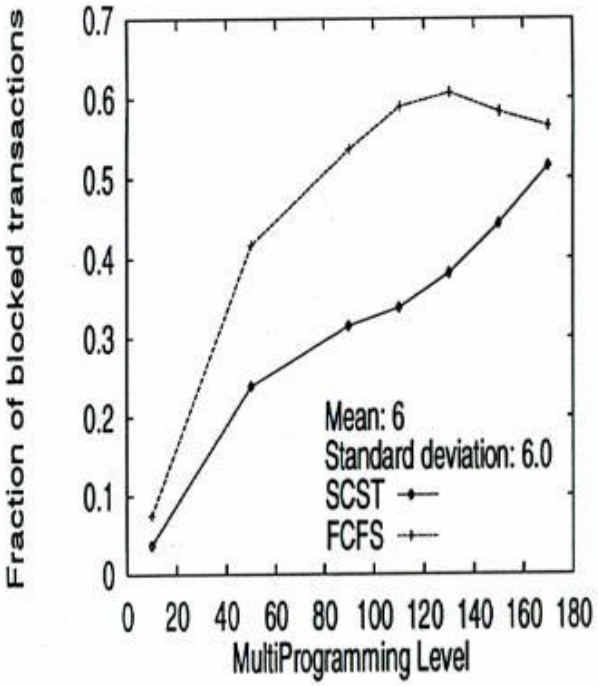
Assumption: All the PEs have the same processing capacity.

It means that the data-processing-skew is caused mainly by the access-skew.

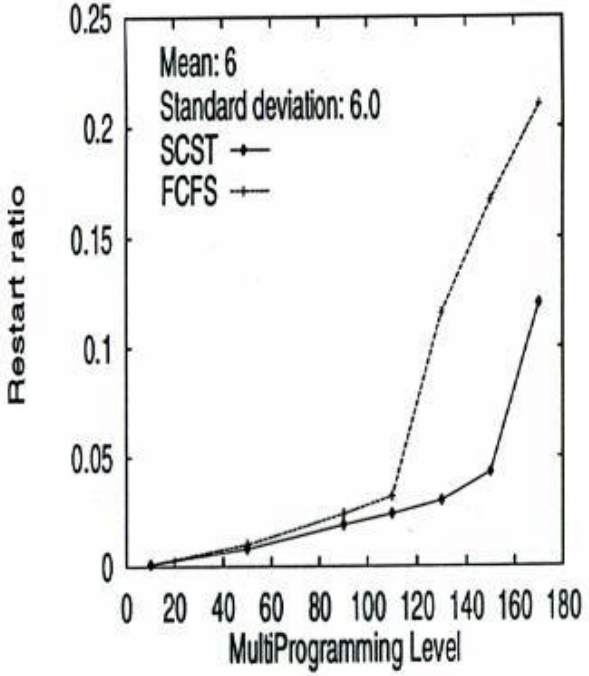
Performance metrics

- **Primary metric:** Throughput, the transaction completion rate.
  - Average response time of a transactions
- **Three other important metrics**
  - Restart ratio: the average number of transaction aborts per commit.
  - Fraction of blocked transactions: the average number of blocked transactions in the steady-state divided by MPL.
  - 95th percentile of the response time: the value which is greater than or equal to steady-state observations of response time 95% of the time and less than them the other 5%.

Simulation results – Fraction of blocked transactions and Restart ratio

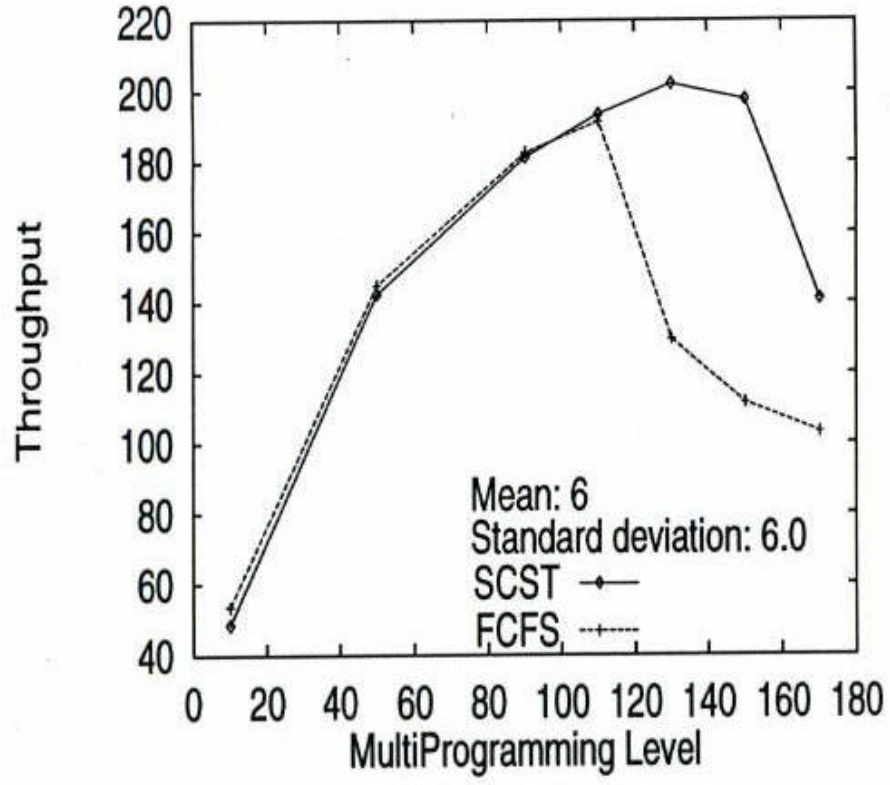


Fraction of blocked transactions with high level of access-skew



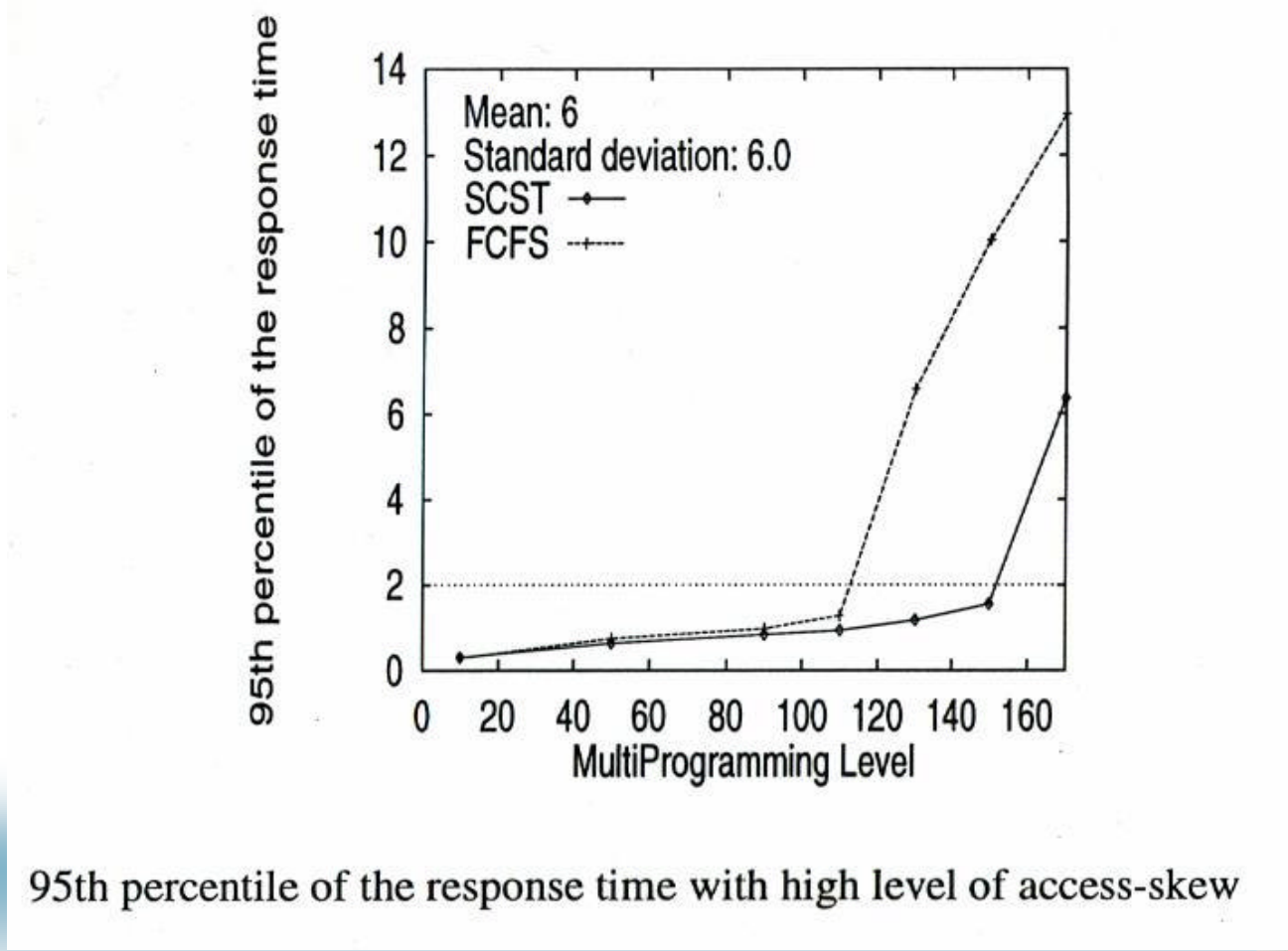
Restart ratio with high level of access-skew

Simulation results - Throughput



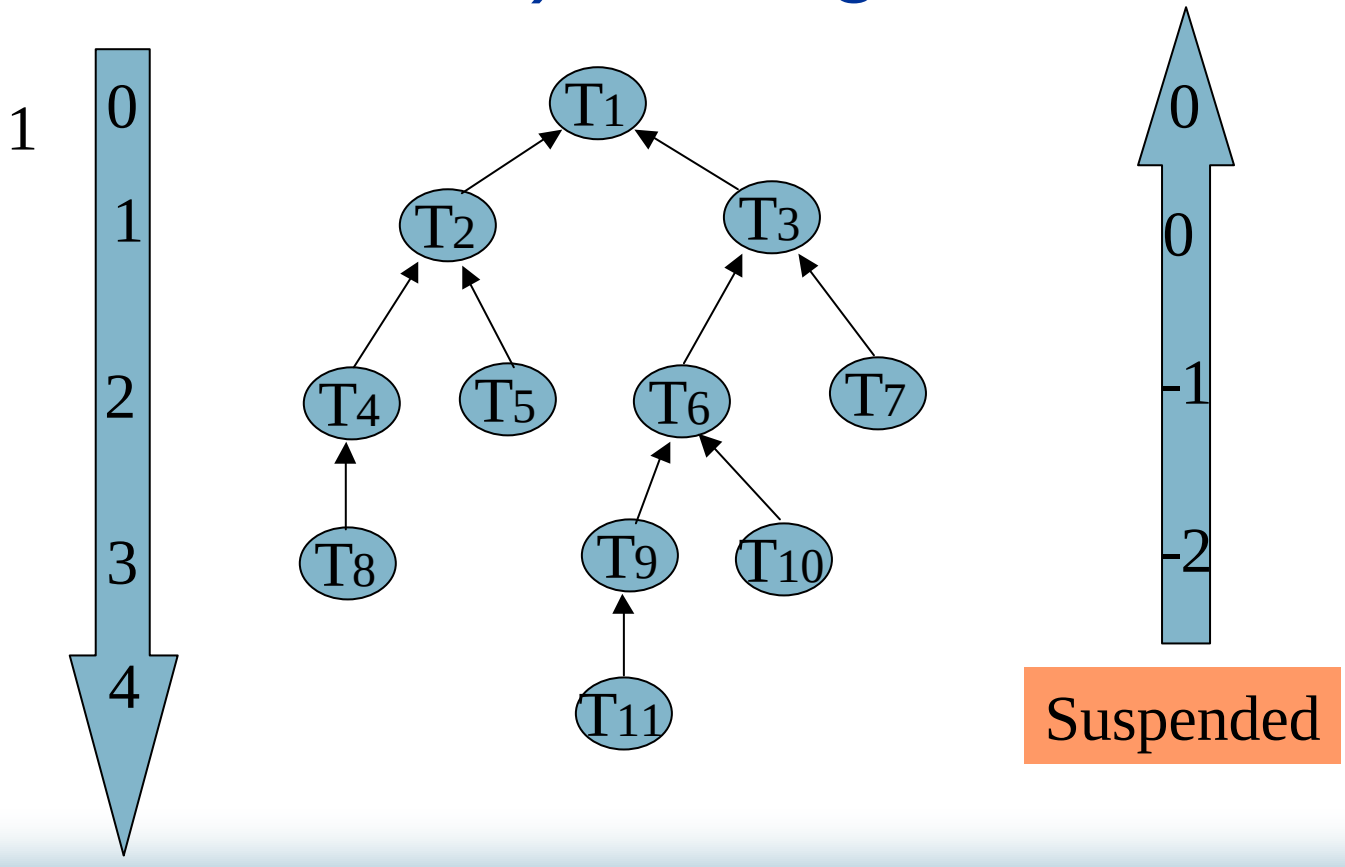
Throughput characteristics with high level of access-skew

Simulation results – 95th percentile of the response time



95th percentile of the response time with high level of access-skew

The WDPP (Wait-Depth Priority Protocol) CC algorithm



Concluding remarks (1)

- An analytic model is successfully established for the performance study of shared-nothing parallel TP systems with 2PL with no-waiting policy.
- A computer simulator is developed for the performance study of shared-nothing parallel TP systems.

Concluding remarks (2)

- An effective scheduling is developed for the parallel TP systems. By using the simulator, we show the performance gain with the scheduling algorithm.
- An effective concurrency control algorithm is also proposed for the parallel TP systems.

Future work

- Develop the general analytic model for the performance of parallel/distributed TP systems.
- Develop more accurate computer simulators of parallel/distributed TP systems.
- Study and propose more effective scheduling algorithms and concurrency control algorithms for parallel/distributed TP systems.