Query Evaluation Techniques for Cluster Database Systems

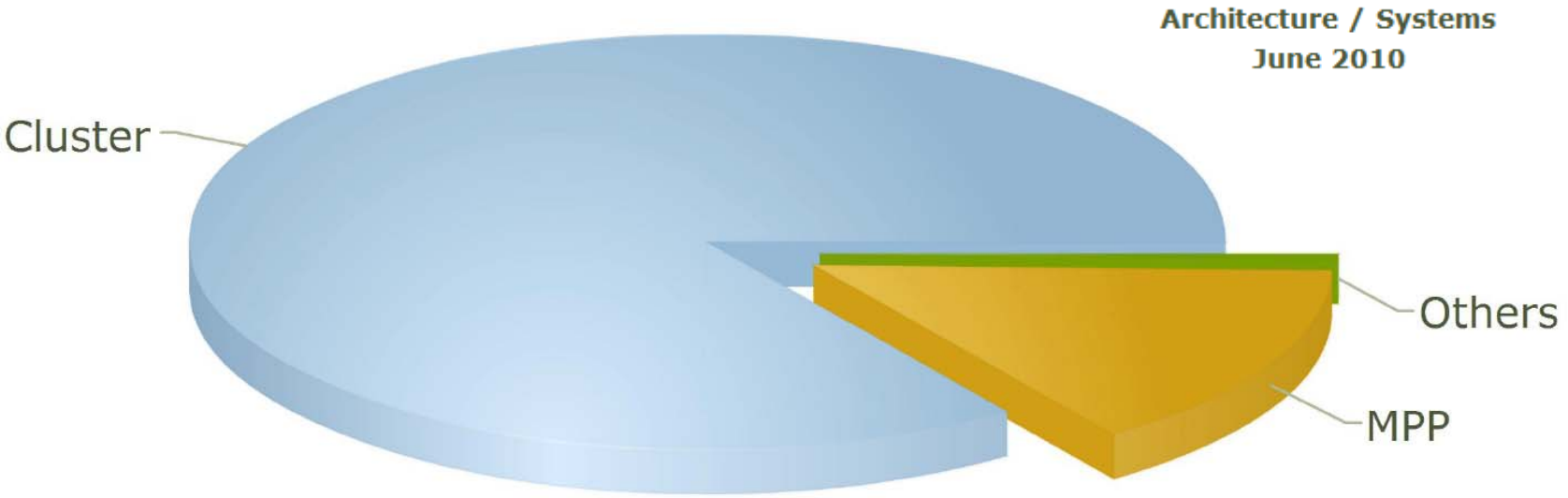
Andrey V. Lepikhov, Leonid B. Sokolinsky
South Ural State University
Russia

22 September 2010

Outline

- Motivation
- Problem Statement
- Background
- Partial mirroring method
- Results
- Future work

Motivation



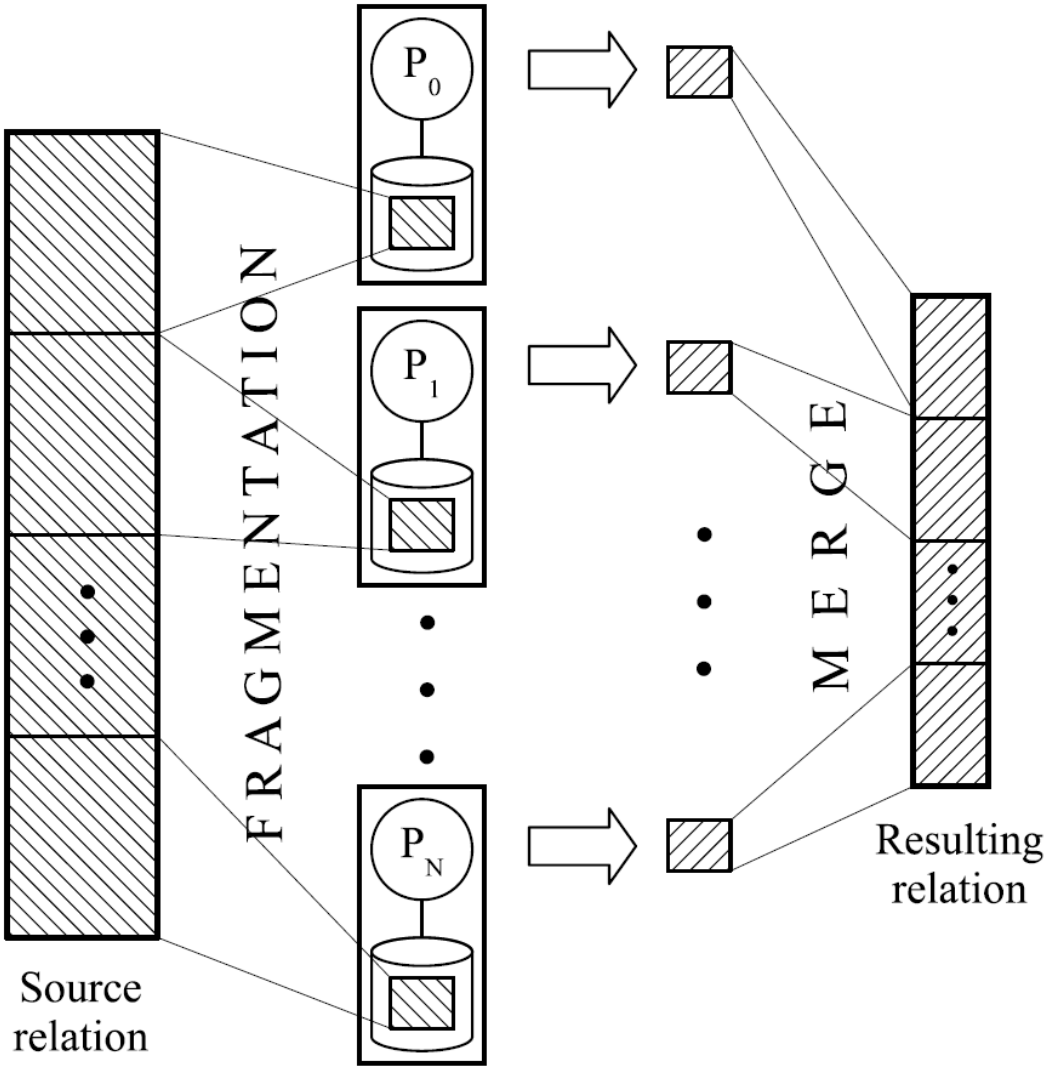
Top500

- Cluster: 84.8%
- MPP: 14.8%
- Others: 0.4%

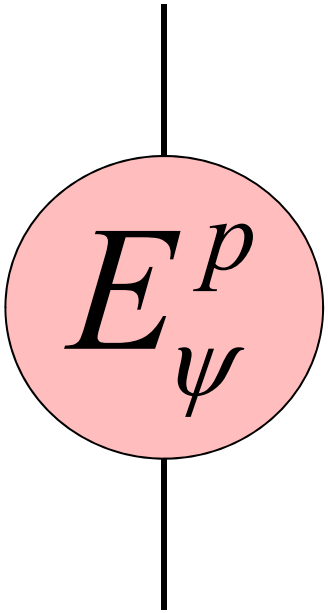
Problem Statement

- Not expensive parallel hardware needs not expensive parallel database management system
- Today we have no such chip parallel database management system

Background

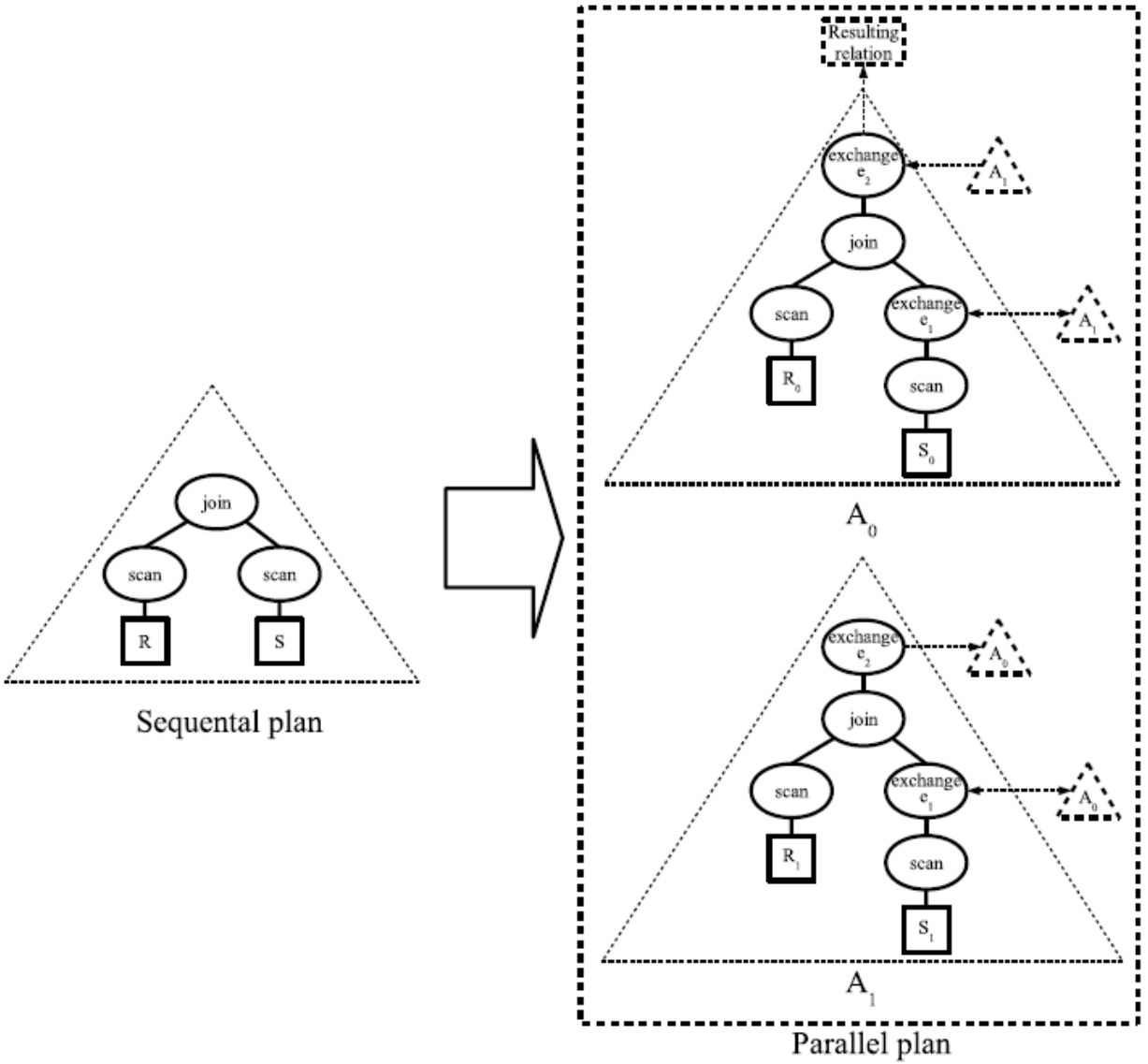

P0
P1
PN
FRAGMENTATION
MERGE
Source relation
Resulting relation

Exchange operator

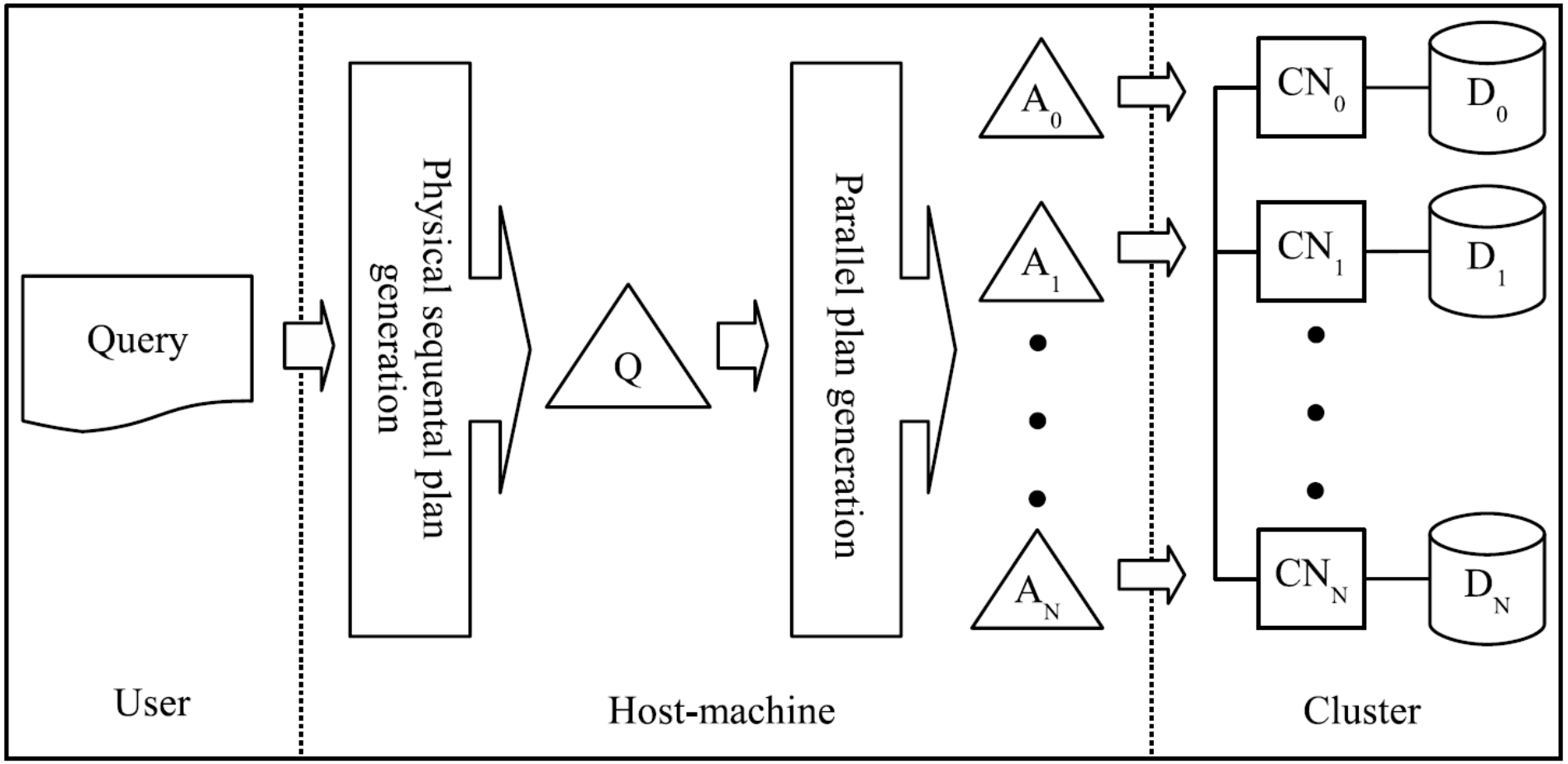


- p: port
- ψ: distributing function

Parallel plan for query Q = R⋈S



Query processing in cluster system



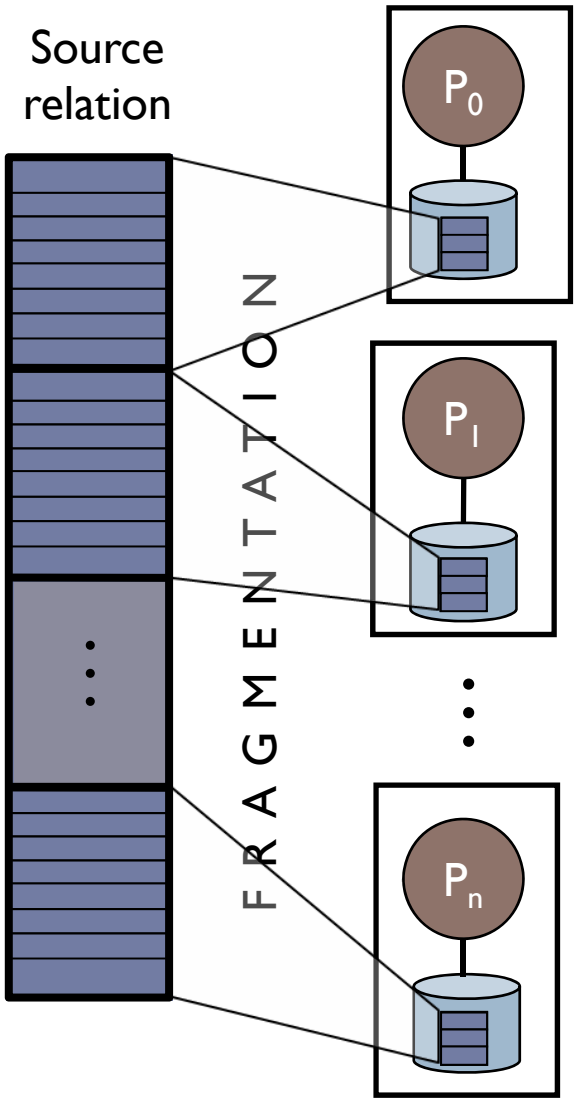
The problem

- **Load balancing**

Partial mirroring method

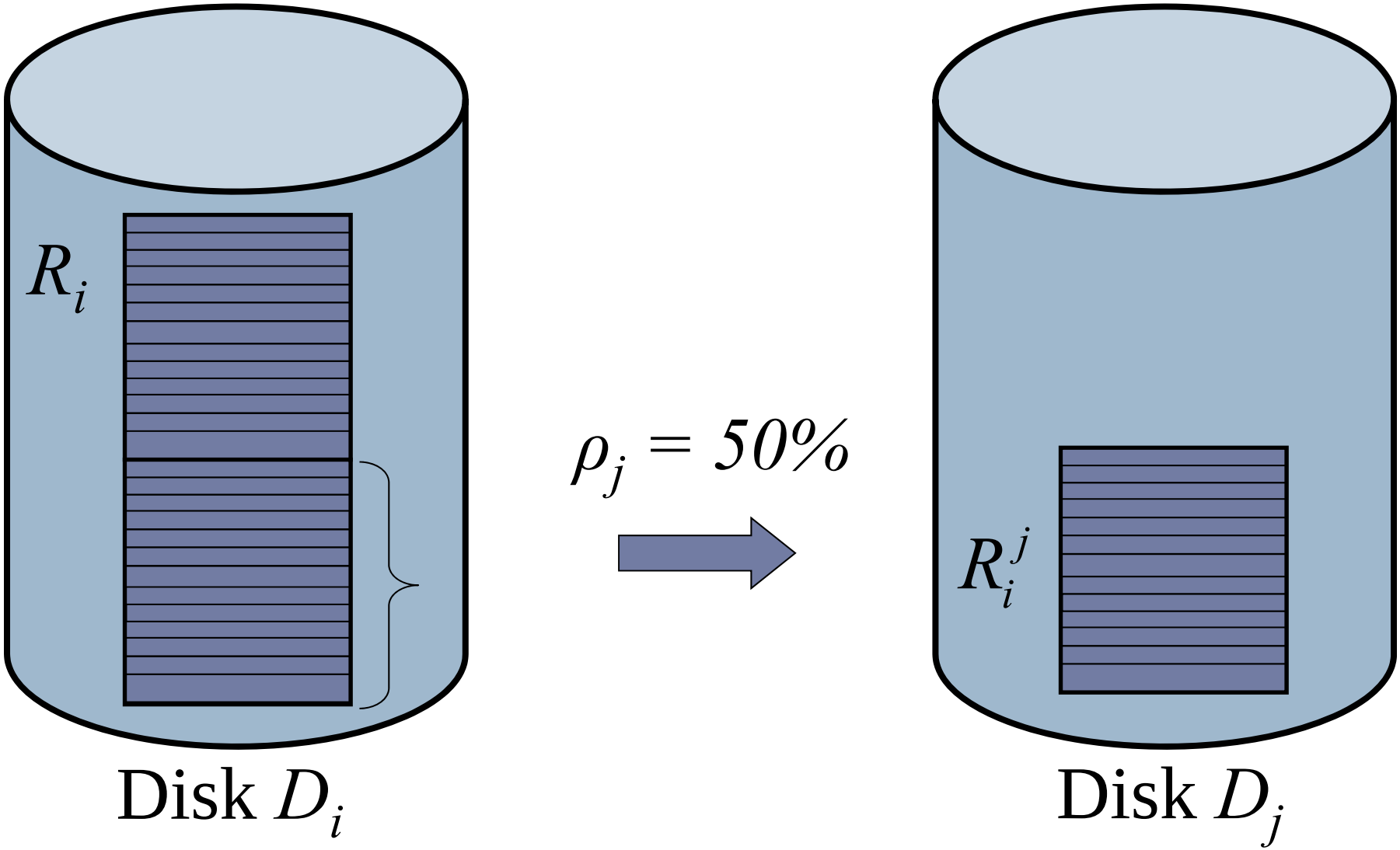
- Fragmentation strategy
- Replication strategy

Fragmentation strategy

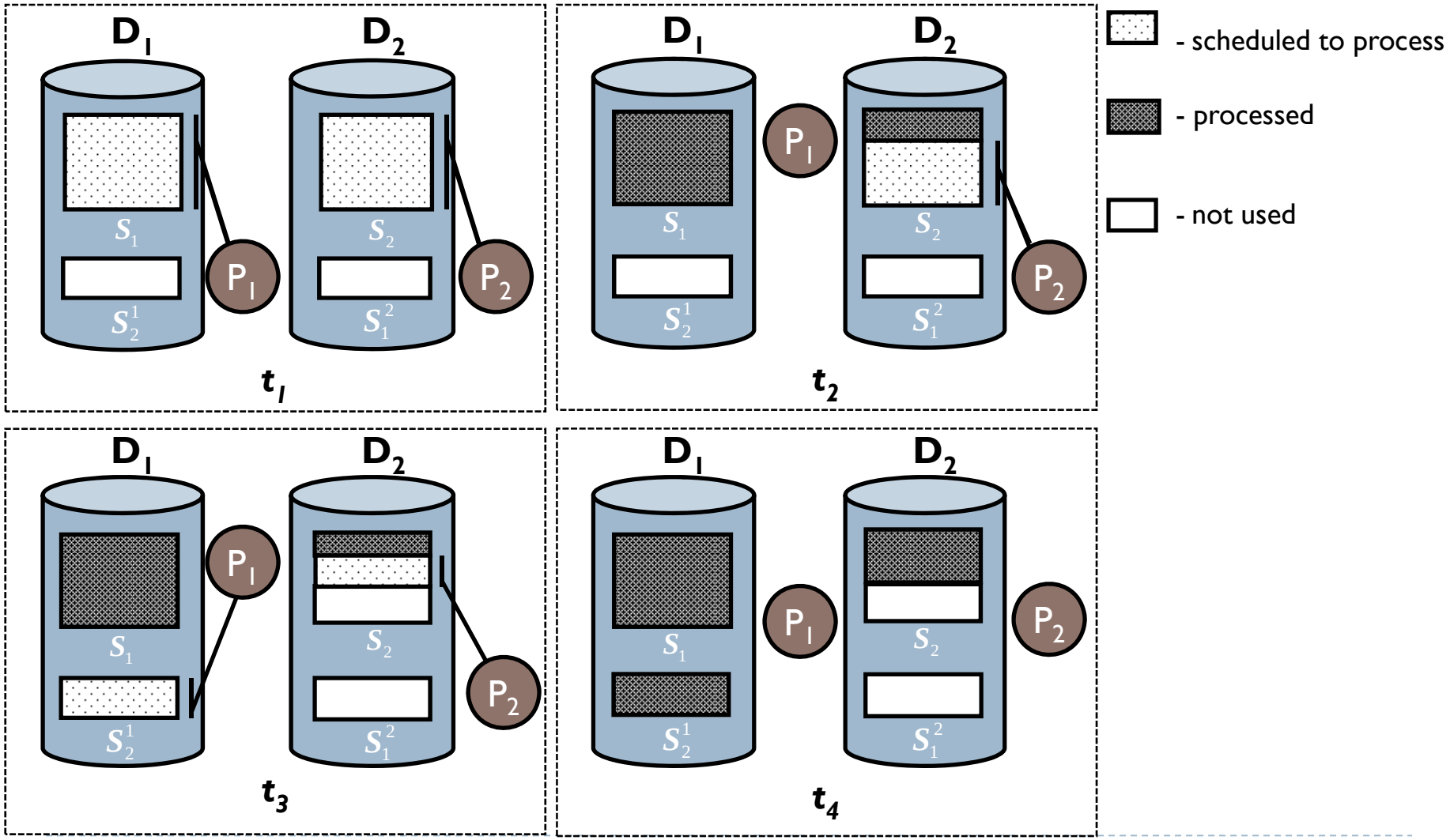


- Relation is divided into fragments distributed among cluster nodes
- Each fragment is divided into sequence of segments with an equal length
- Segment is the minimal unit of replication

Replication strategy



Load balancing method

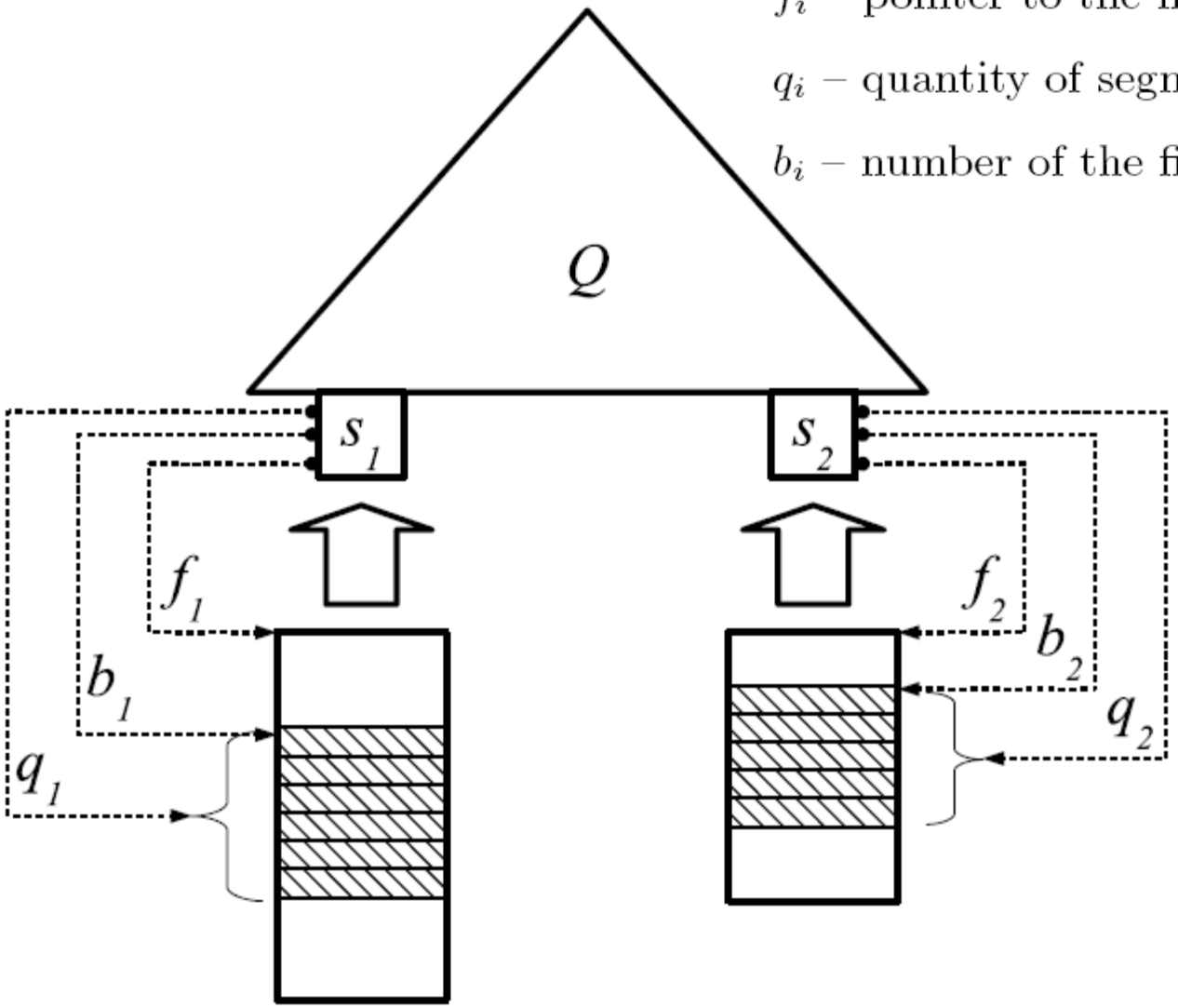


Parallel agent with two input streams

f_i – pointer to the fragment;

q_i – quantity of segments in part to be processed;

b_i – number of the first segment in the part;



Load balancing algorithm

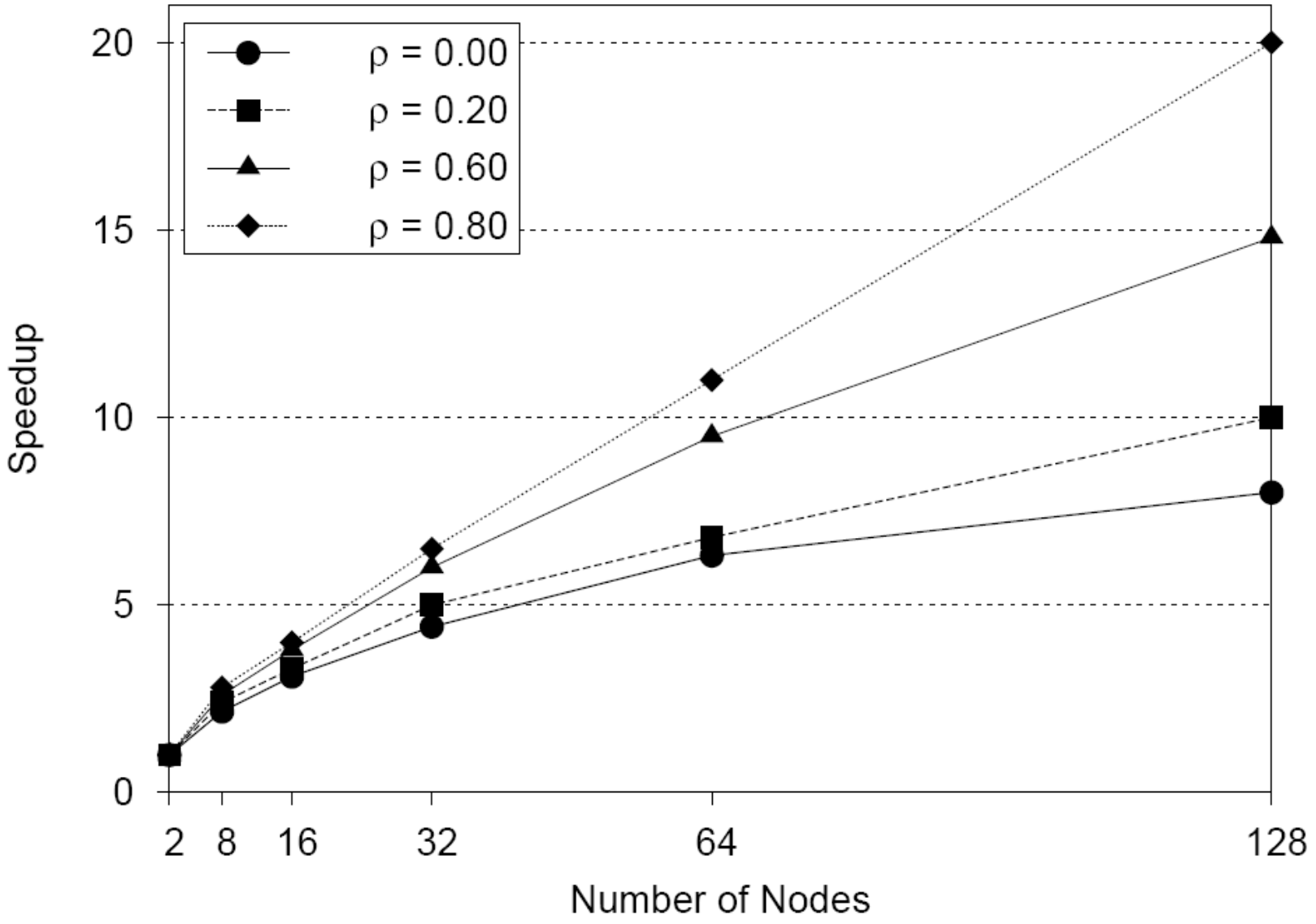
/* load balancing procedure between agents $\bar{Q}$ (forward) and $\tilde{Q}$ (backward). */
$\bar{u} = Node(\bar{Q})$; // pointer to the agent node $\bar{Q}$
pause $\tilde{Q}$; // turn agent $\tilde{Q}$ into passive state
for (i=1; i<=n; i++) {

if($\tilde{Q}.s[i].a == 1$) {
$\tilde{f}_i = \tilde{Q}.s[i].f$; // fragment assigned to agent $\tilde{Q}$
$\bar{r}_i = Re(\tilde{f}_i, \bar{u})$; // replica $\tilde{f}_i$ into the node $\bar{u}$
$\delta_i = Delta(\tilde{Q}.s[i])$; // quantity of segments to transfer
$\tilde{Q}.s[i].q- = \delta_i$;
$\bar{Q}.s[i].f = \bar{r}_i$;
$\bar{Q}.s[i].b = \tilde{Q}.s[i].b + \tilde{Q}.s[i].q$;
$\bar{Q}.s[i].q = \delta_i$;
} **else**
print("Load balancing is not permitted.");
};
activate $\tilde{Q}$ // turn agent $\tilde{Q}$ into active state
activate $\bar{Q}$ // turn agent $\bar{Q}$ into active state

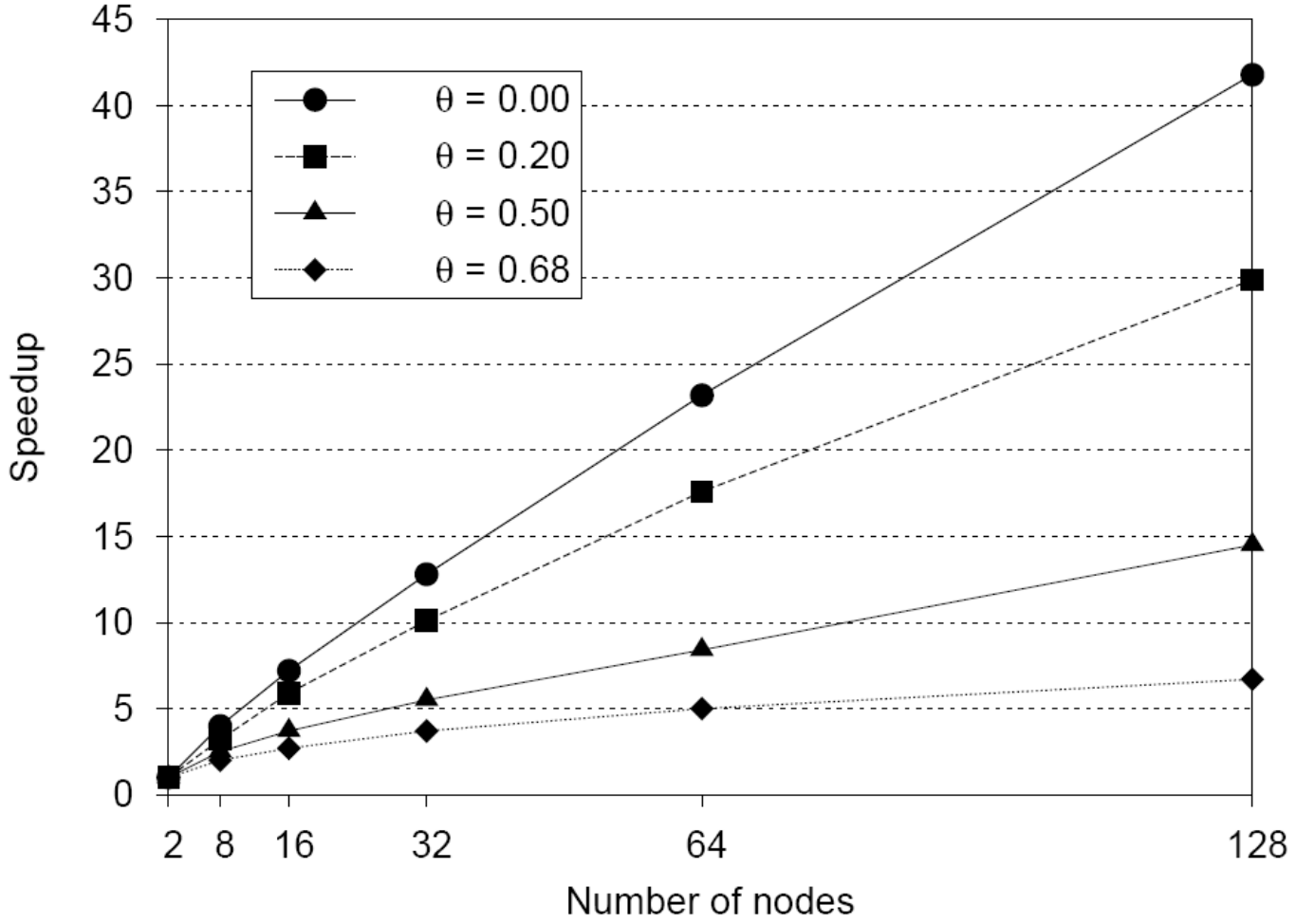
Parameters of experiments

Parameter	Value
Parameters of a cluster system	
Quantity of processing nodes	128
Processor type	Intel Xeon E5472 (4 cores with 3.0 GHz)
RAM size	8 GB/node
Disk memory size	120 GB/node
Communication Network type	InfiniBand (20 Gb/s)
Operating system	SUSE Linux Enterprise Server 10
database parameters	
Number of tuples in relation R	60 million
Number of tuples in relation S	1.5 million
Query parameters	
Load balancing indicator for relation R	0 (load balancing is not admitted)
Load balancing indicator for relation S	1 (load balancing is admitted)

Speedup versus replication factor



Speedup versus skew factor θ



- 0.68 corresponds to the ”80-20” rule (80 percents of tuples of the relation will be stored in 20 percents of fragments)
- 0 corresponds to the uniform distribution

Future Work

- To incorporate the proposed technique of parallel query execution into open source PostgreSQL DBMS.
- To extend this approach on GRID DBMS for clusters with multicor processors.

Thank you