Distributed Clustering-Based Index

Verónica Gil-Costa, Mauricio Marín , Nora Reyes

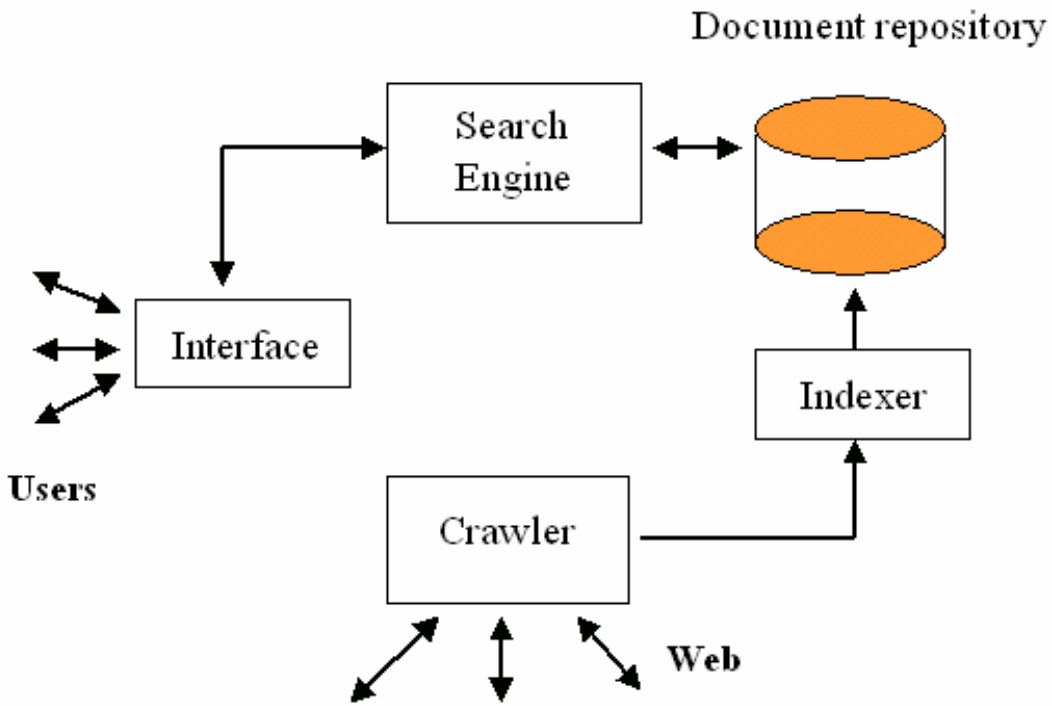
Universidad Nacional de San Luis, Argentina

Yahoo! Research, Universidad de Chile

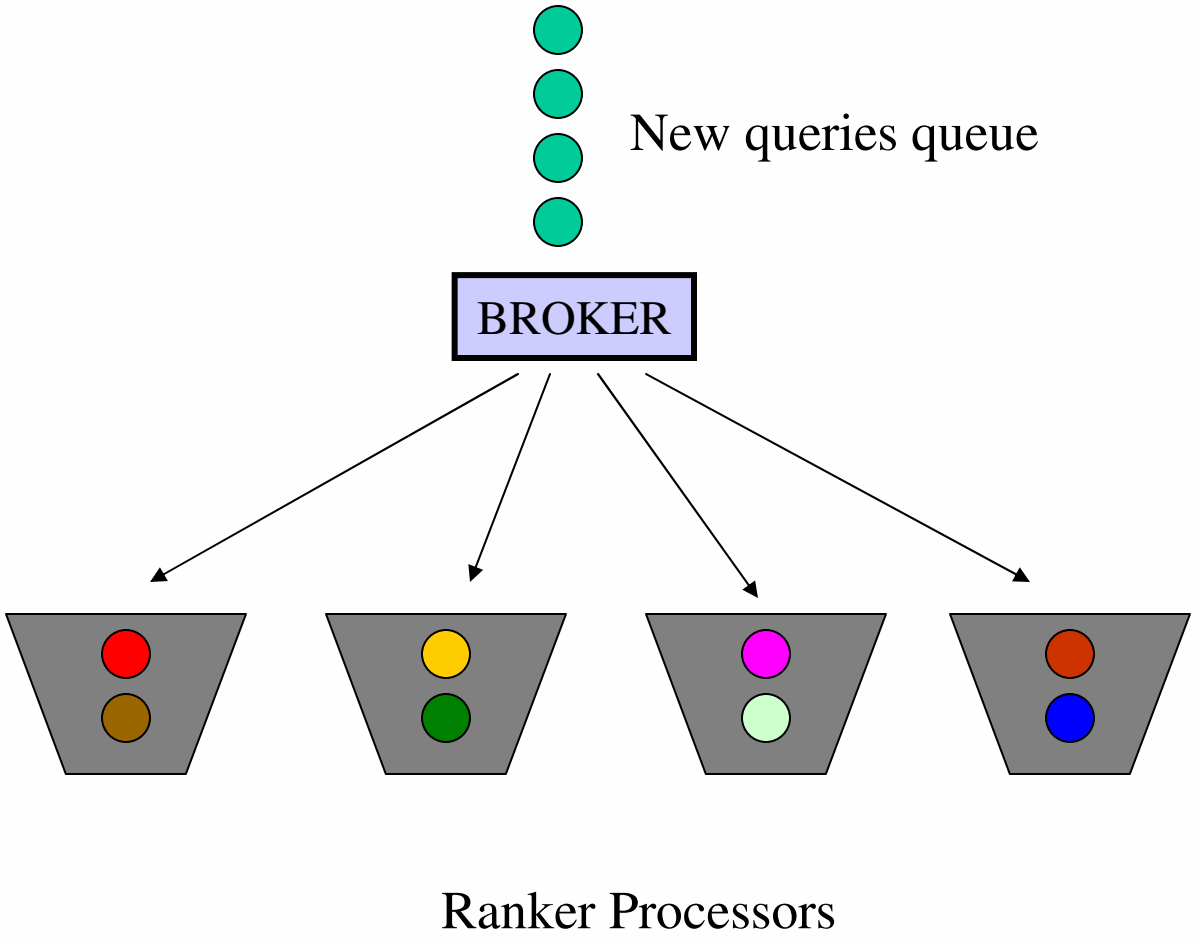
Outline

- Synchronous versus Asynchronous modes of computing
- Round-robin query processing.
- Efficient distributed indexing and parallel query processing.

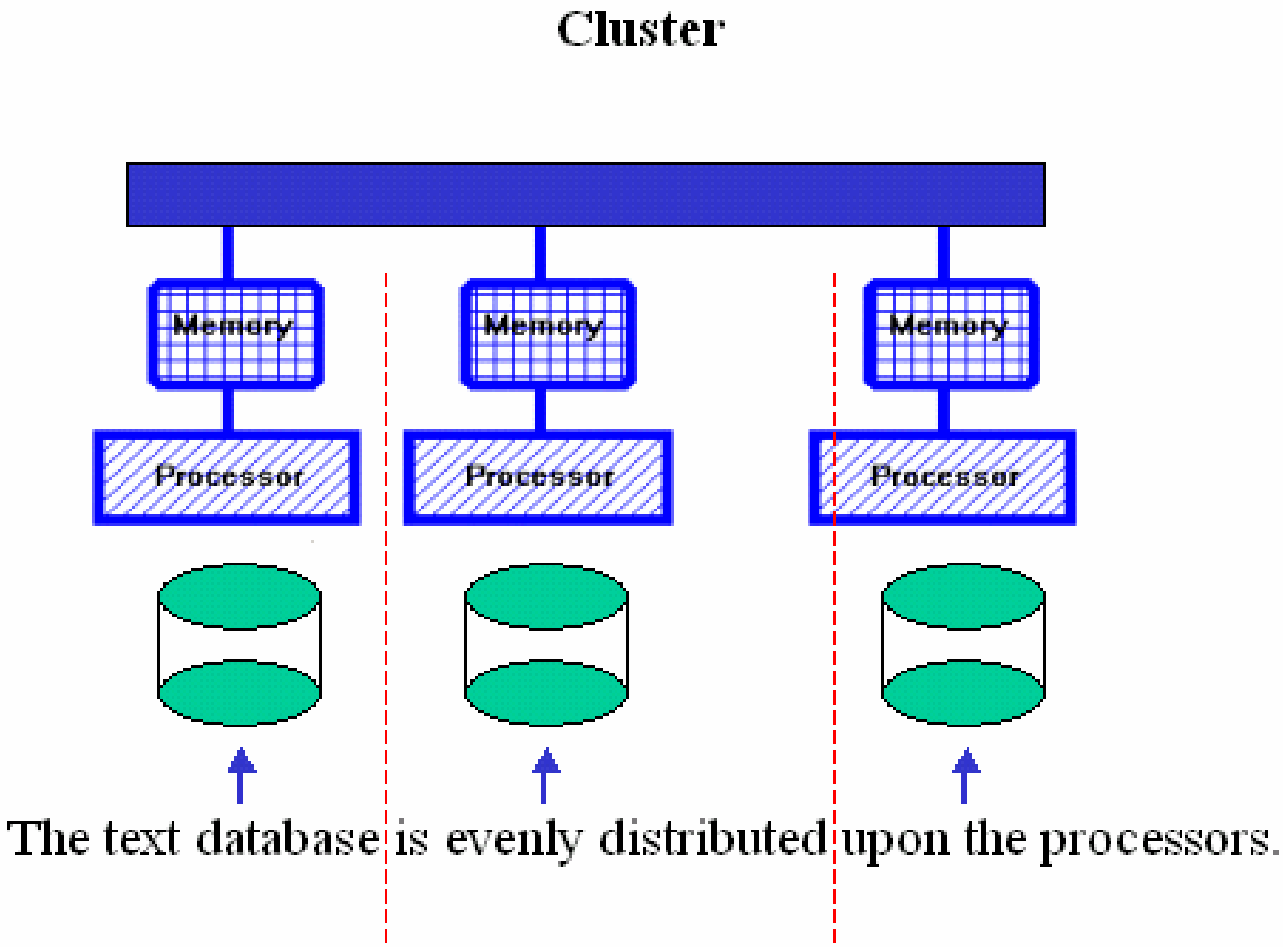
Search Engines



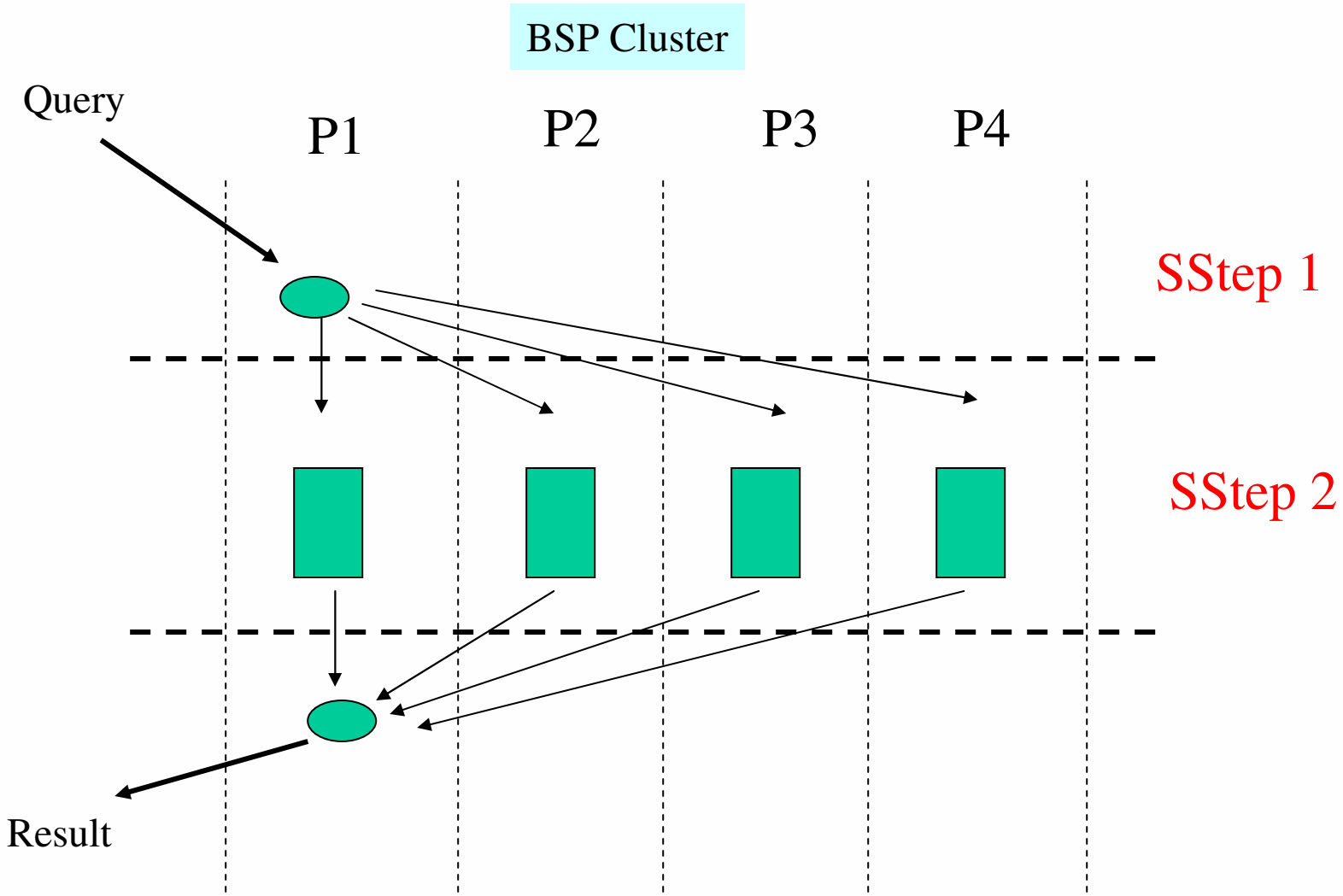
Circular selection of the ranker processor



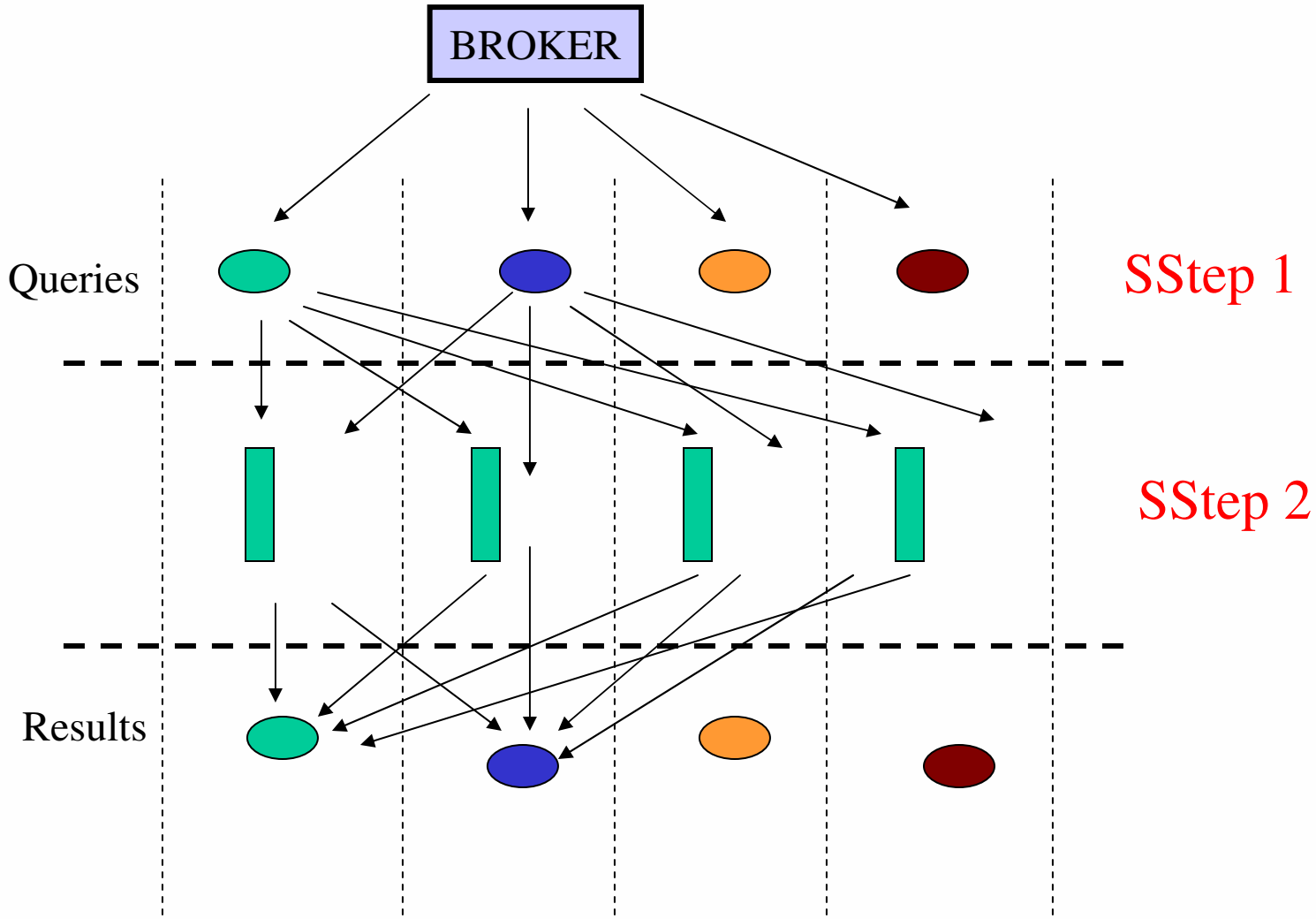
Sharing nothing model



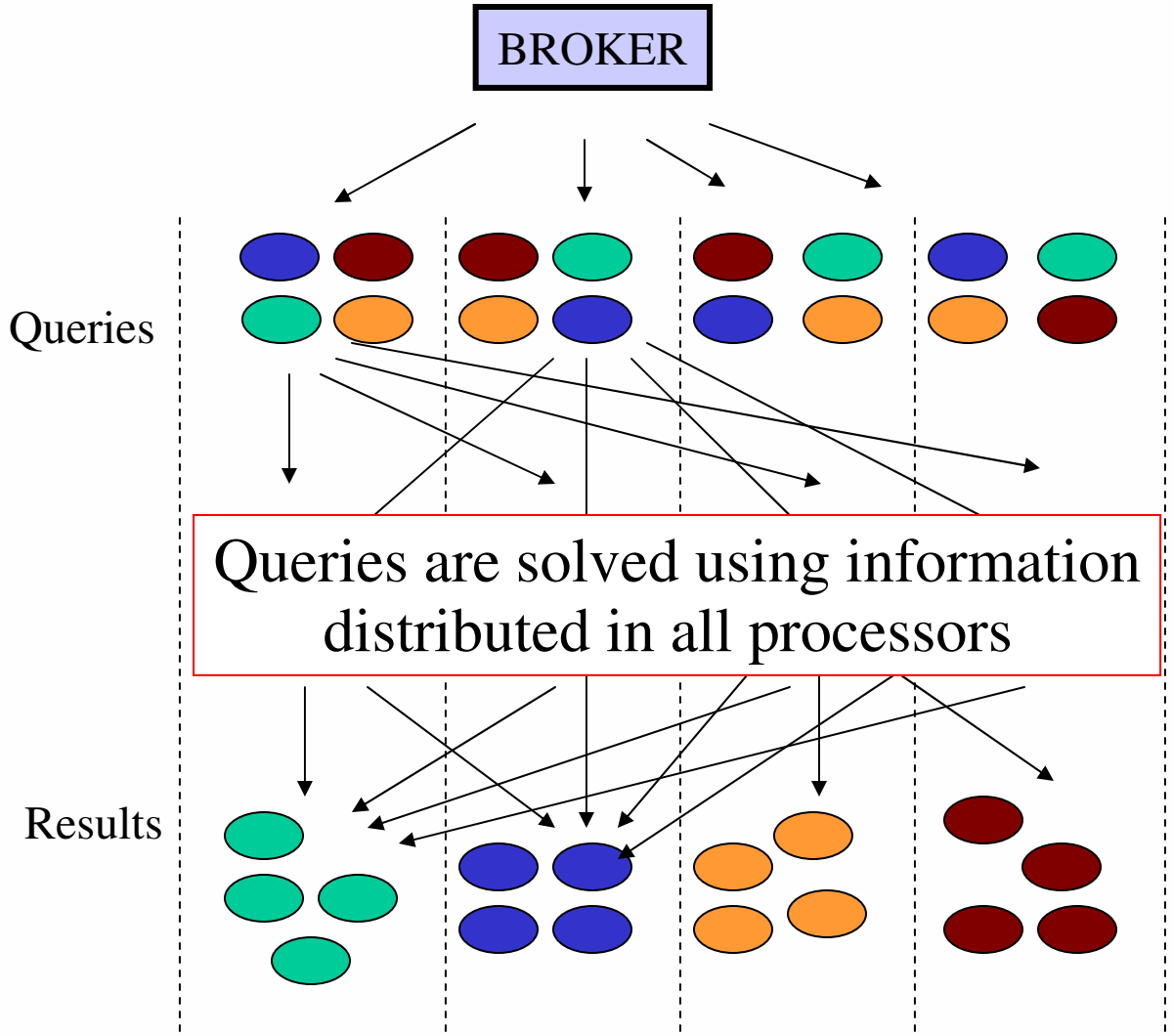
Processing a single query



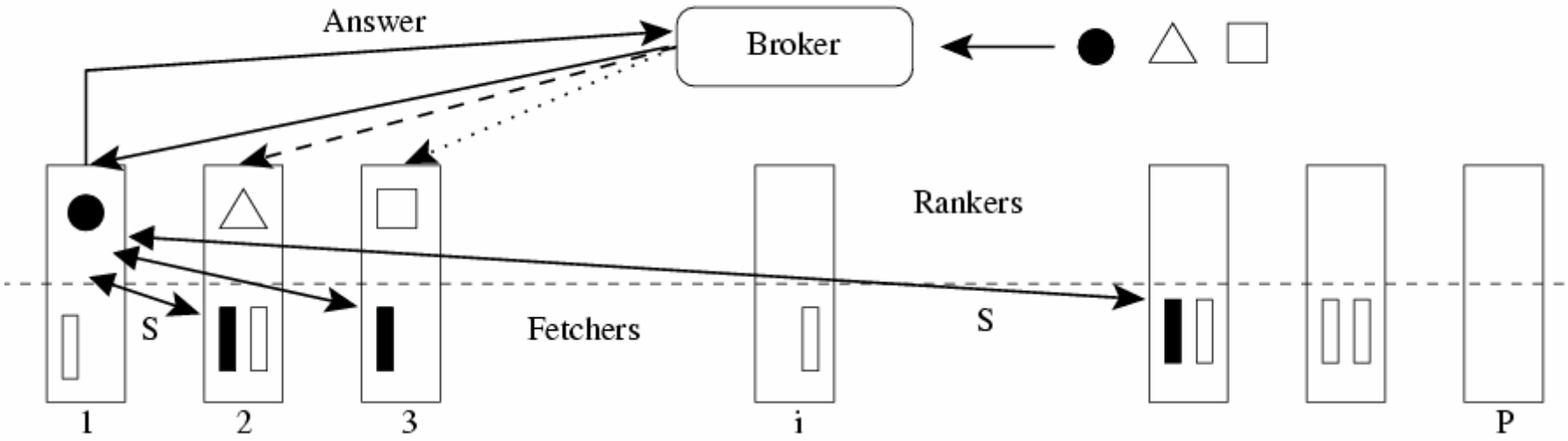
Processing *P* queries



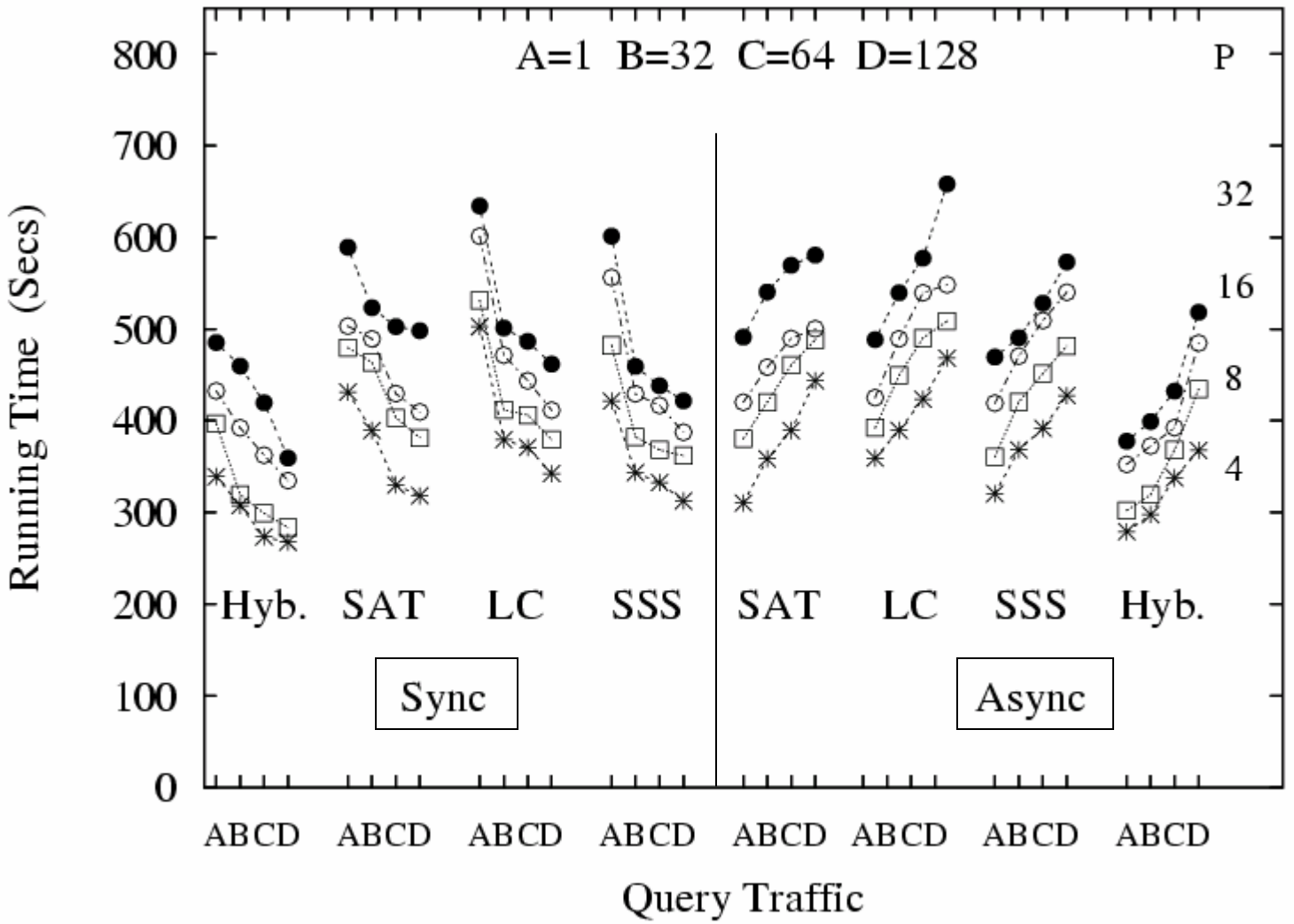
Query Processing under High Traffic

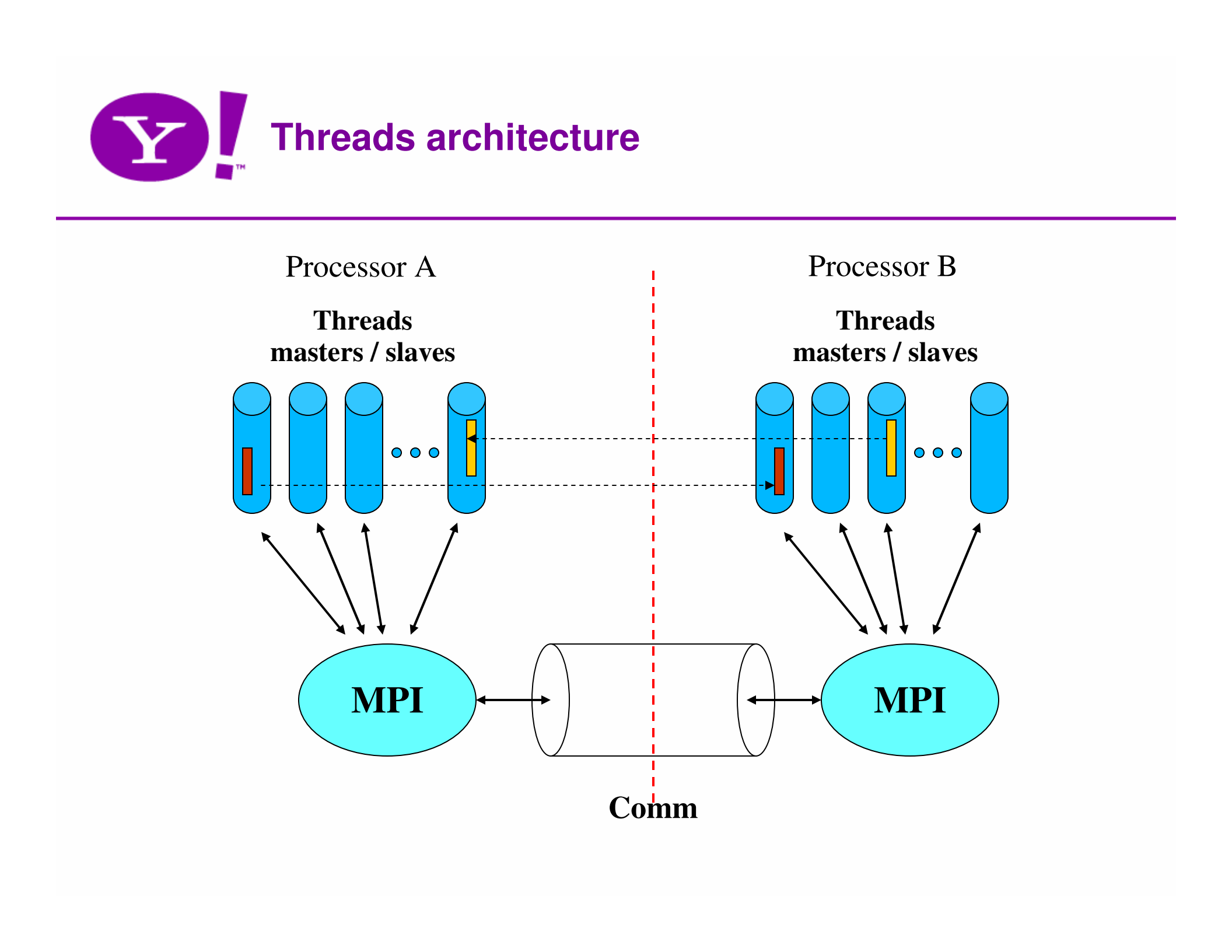


Ranker and fetcher processors

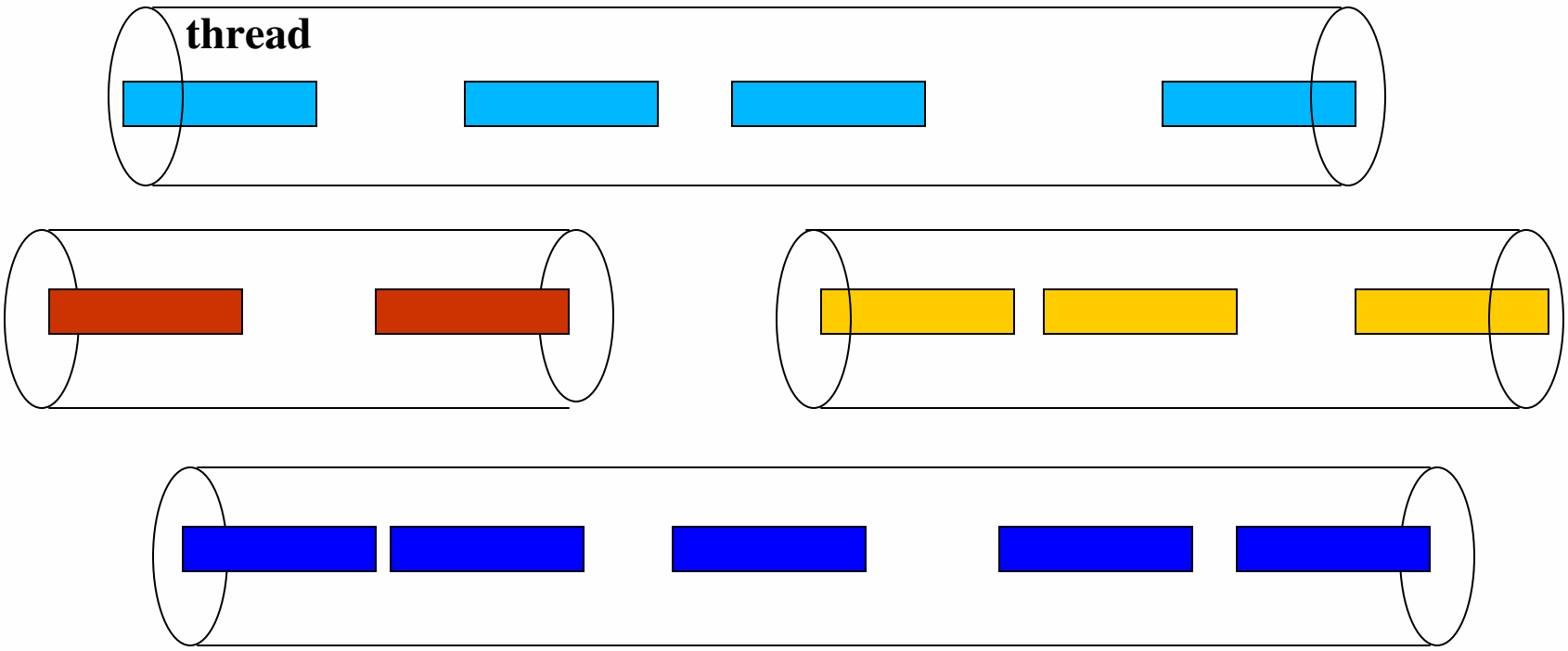


Dealing with high and low query traffic

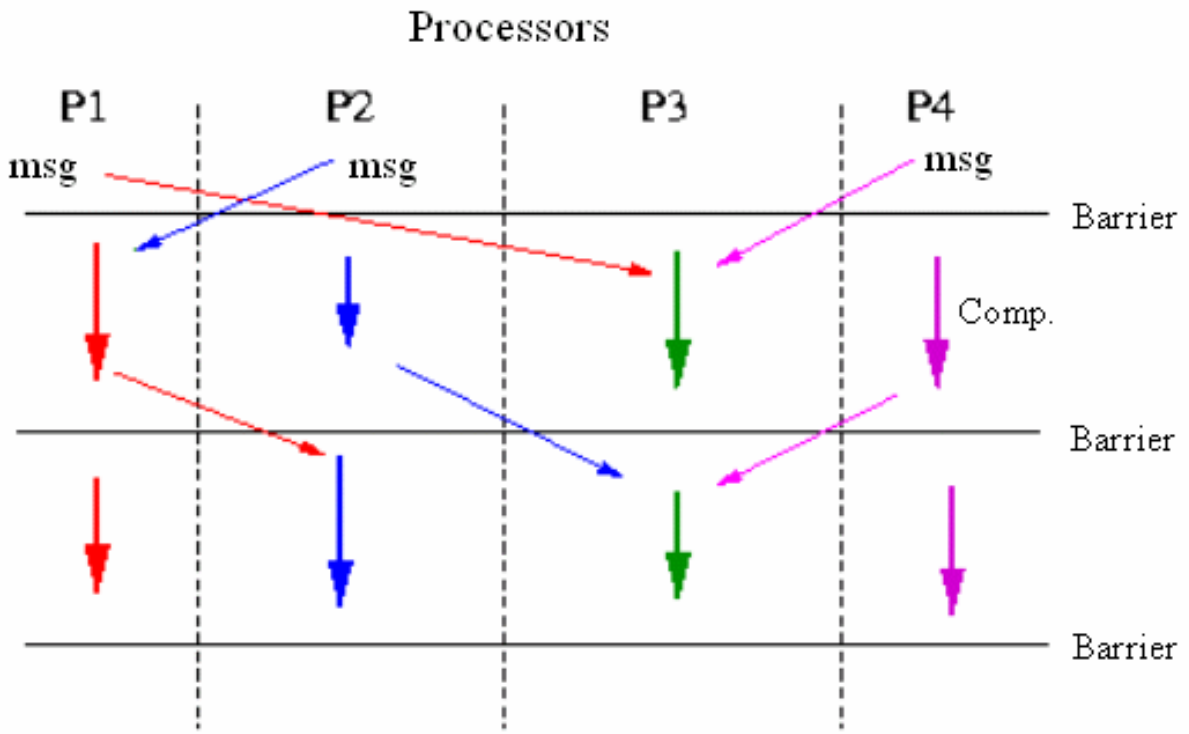



Threads architecture
Processor A
Threads
masters / slaves
Processor B
Threads
masters / slaves
MPI
MPI
Comm

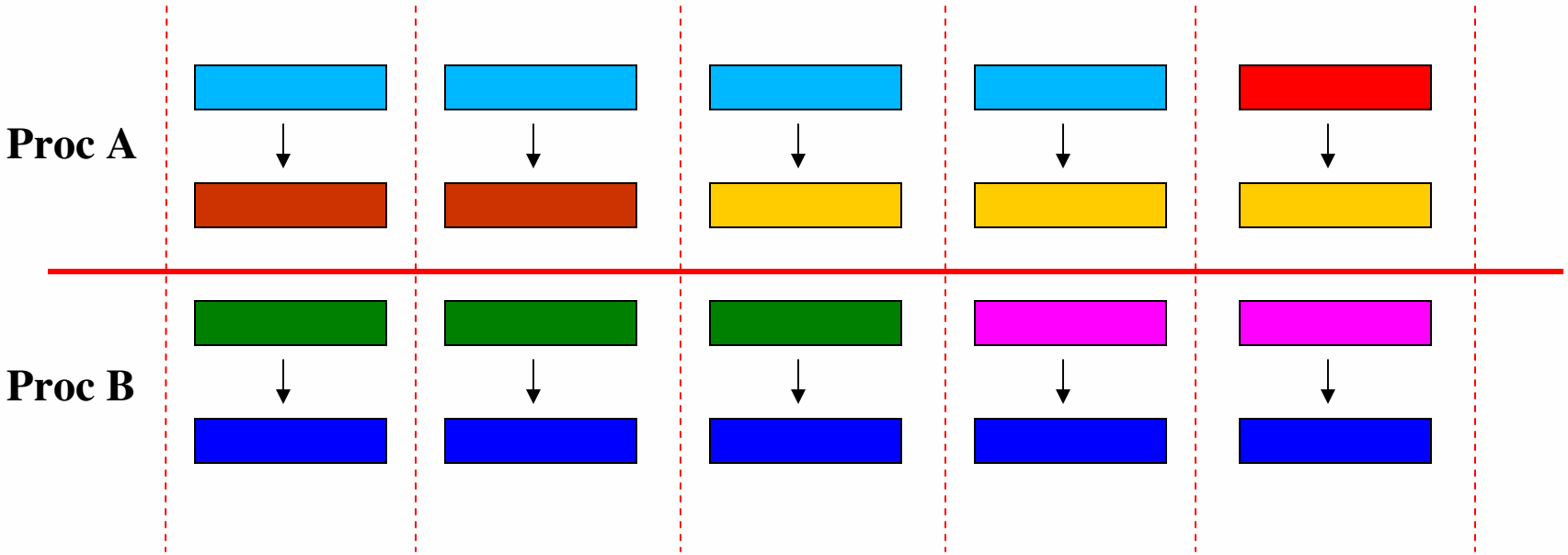
Multi-threaded query processing



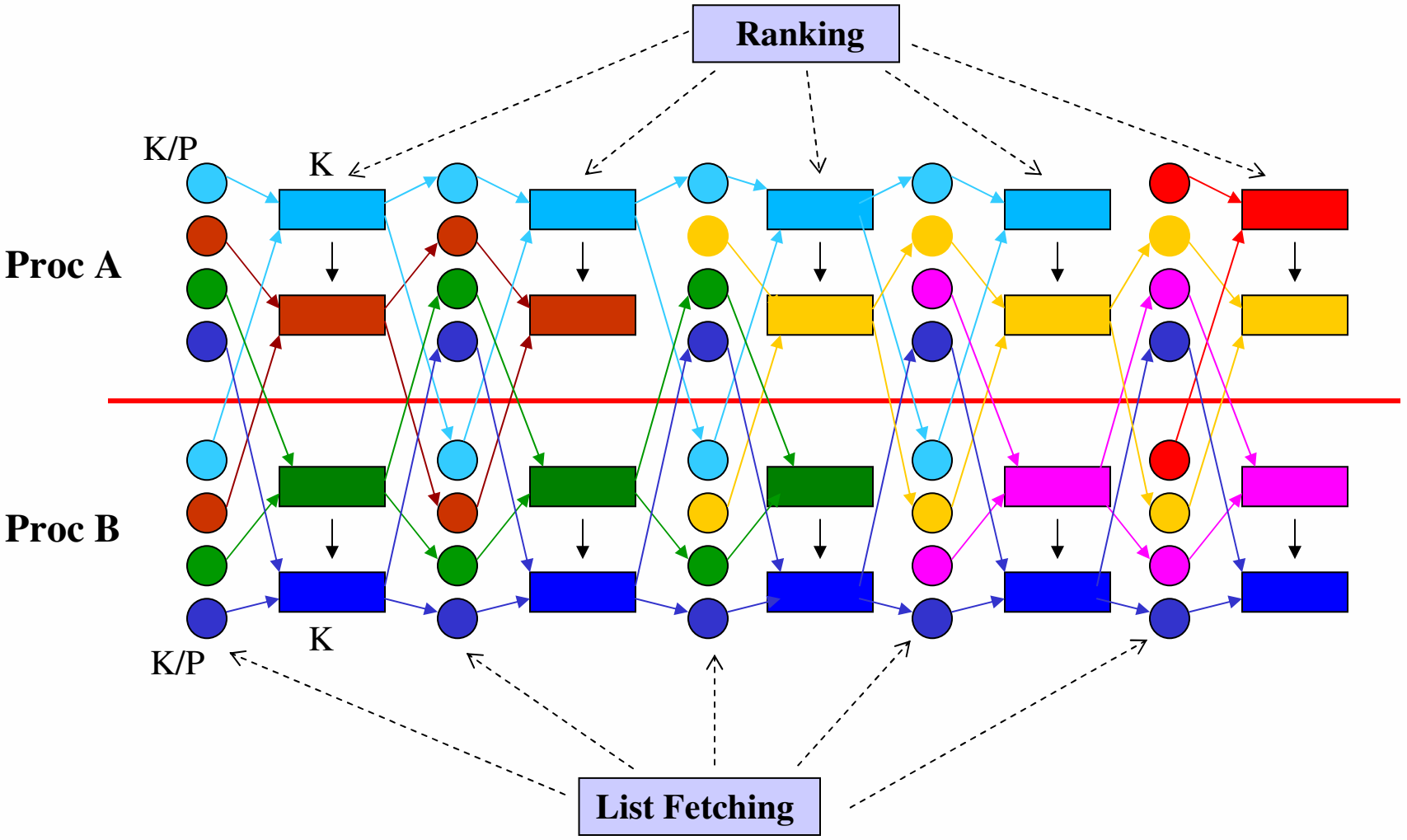
BSP Computing



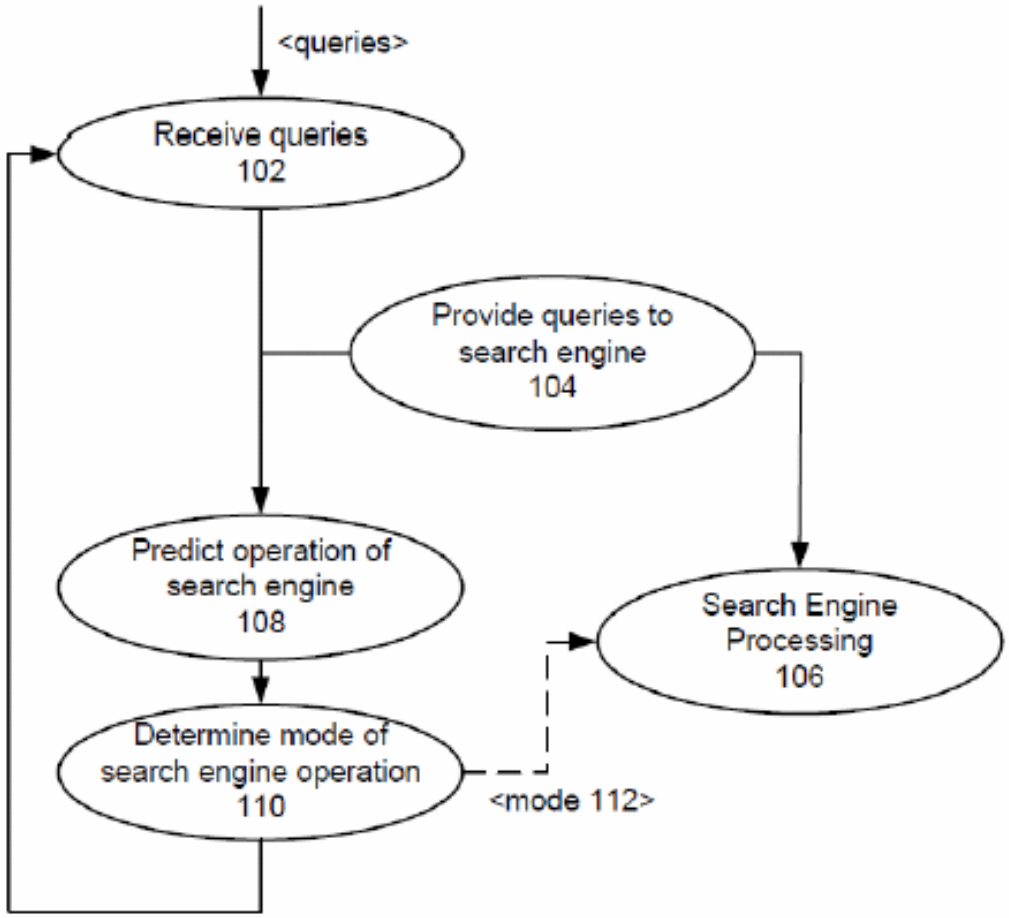
Round-robin query processing



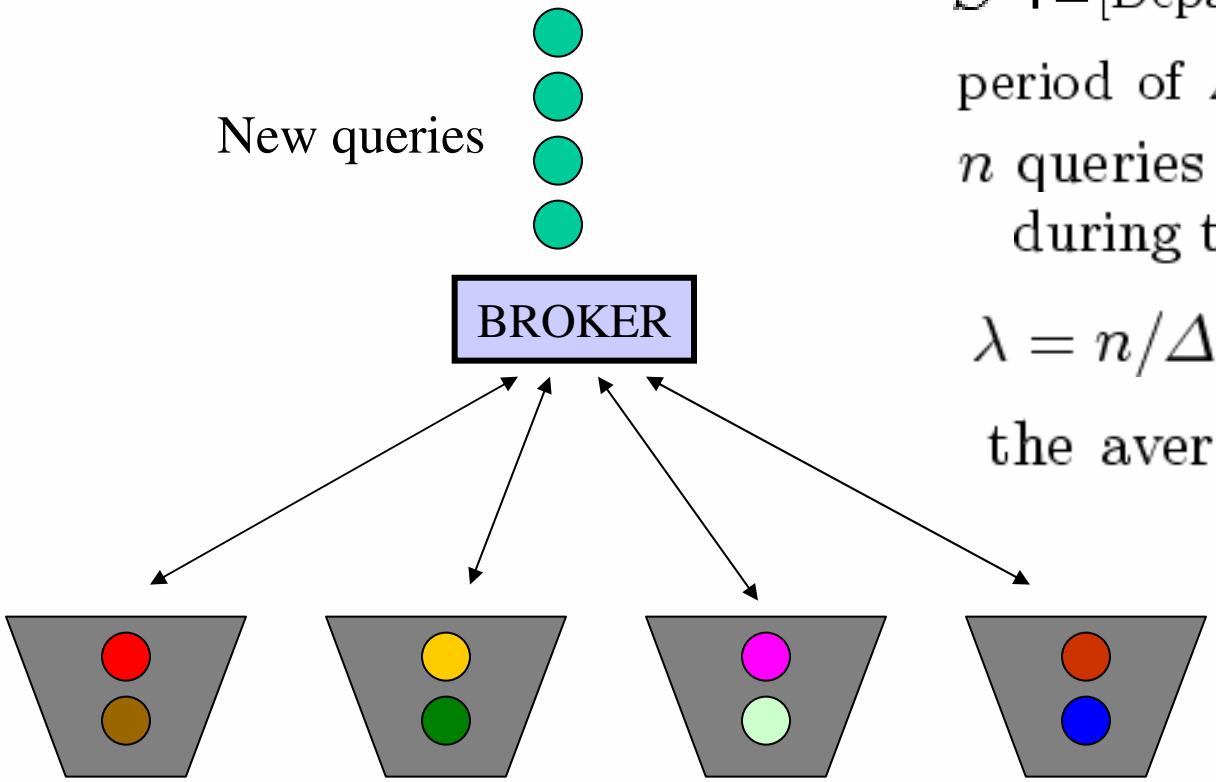
Round-robin query processing



Switching between Sync/Async modes.



Automatic switching → $G/G/\infty$



S += [DepartureTime - ArrivalTime]

period of Δ units of time,

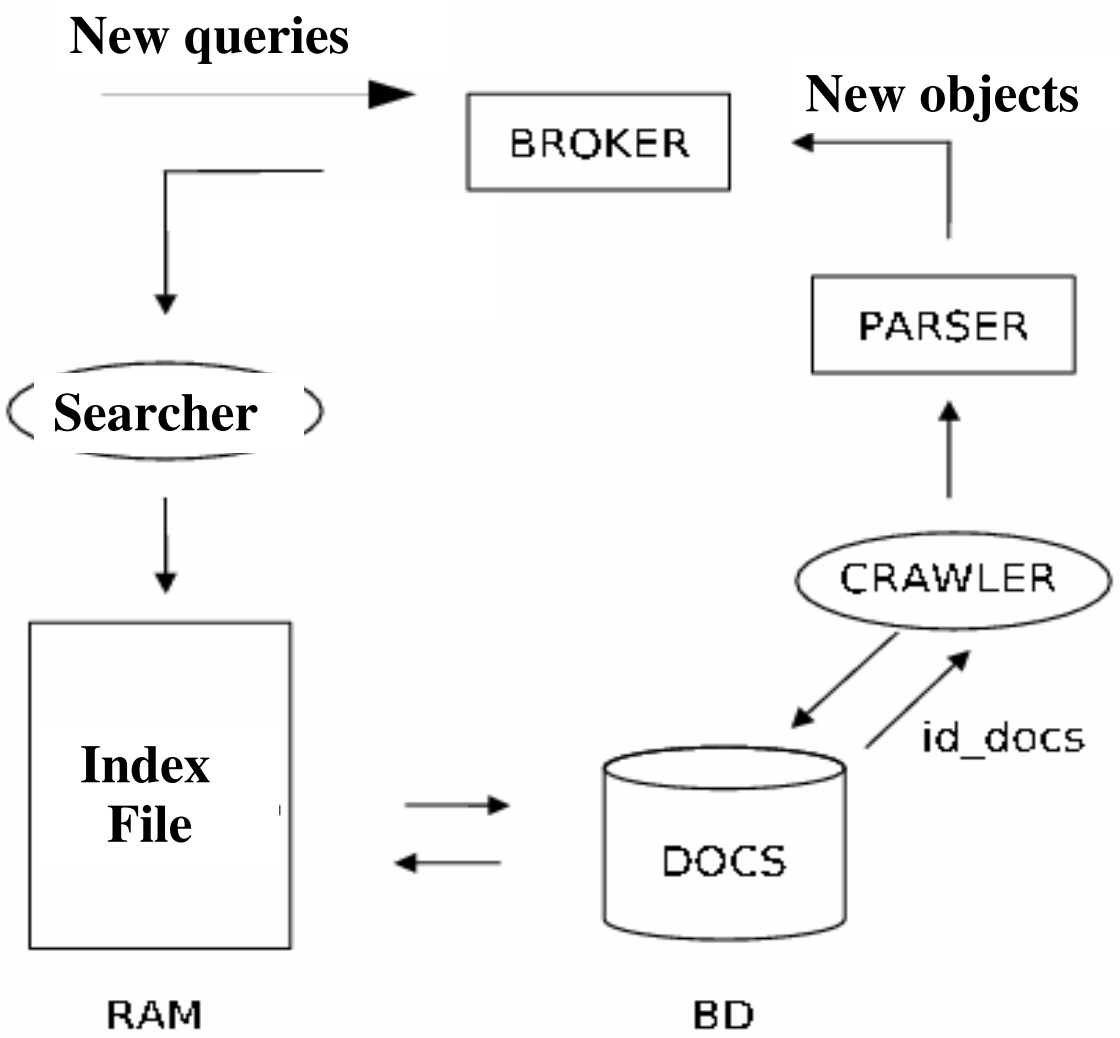
n queries are received during the interval Δ,

$\lambda = n/\Delta \qquad \mu = n/S$

the average q is S/Δ

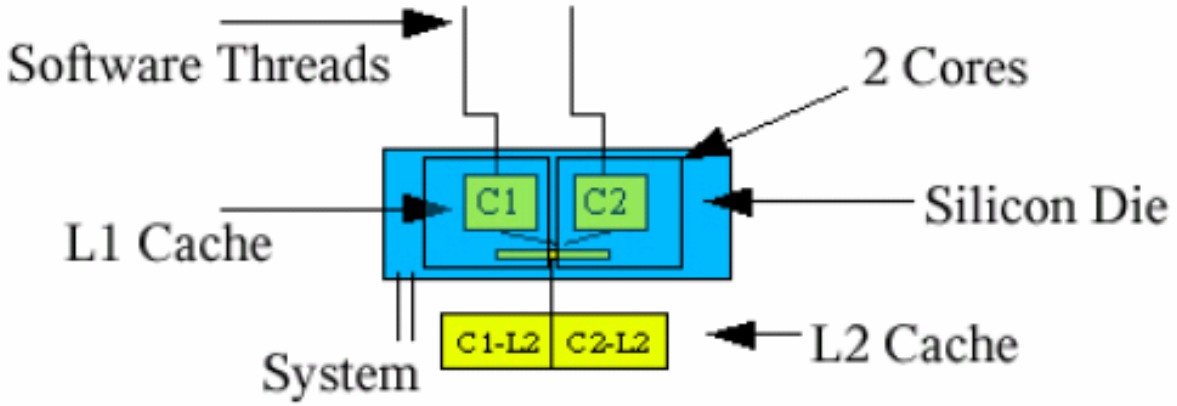
$q =$ Average number of queries being processed.

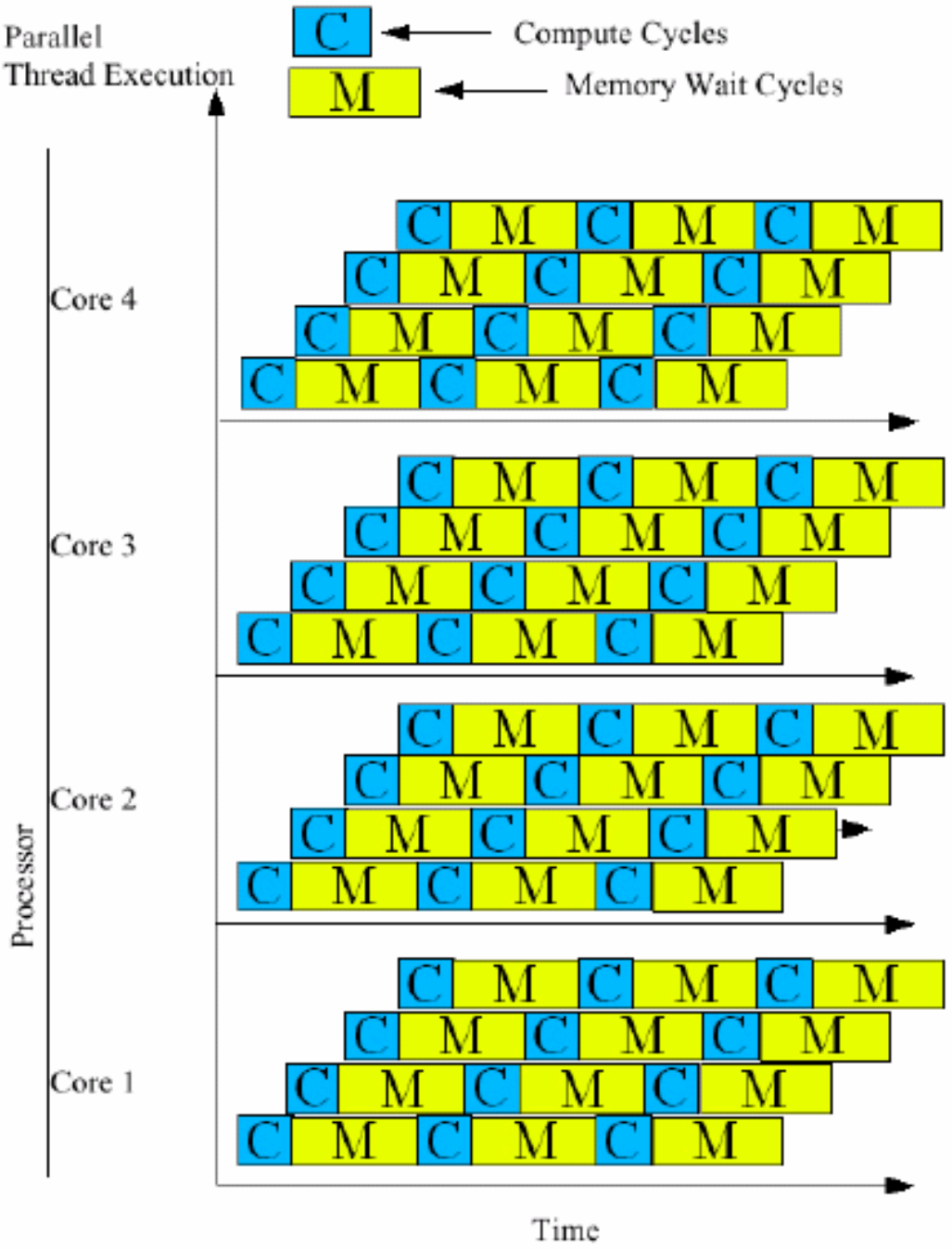
Accepting on-line updates



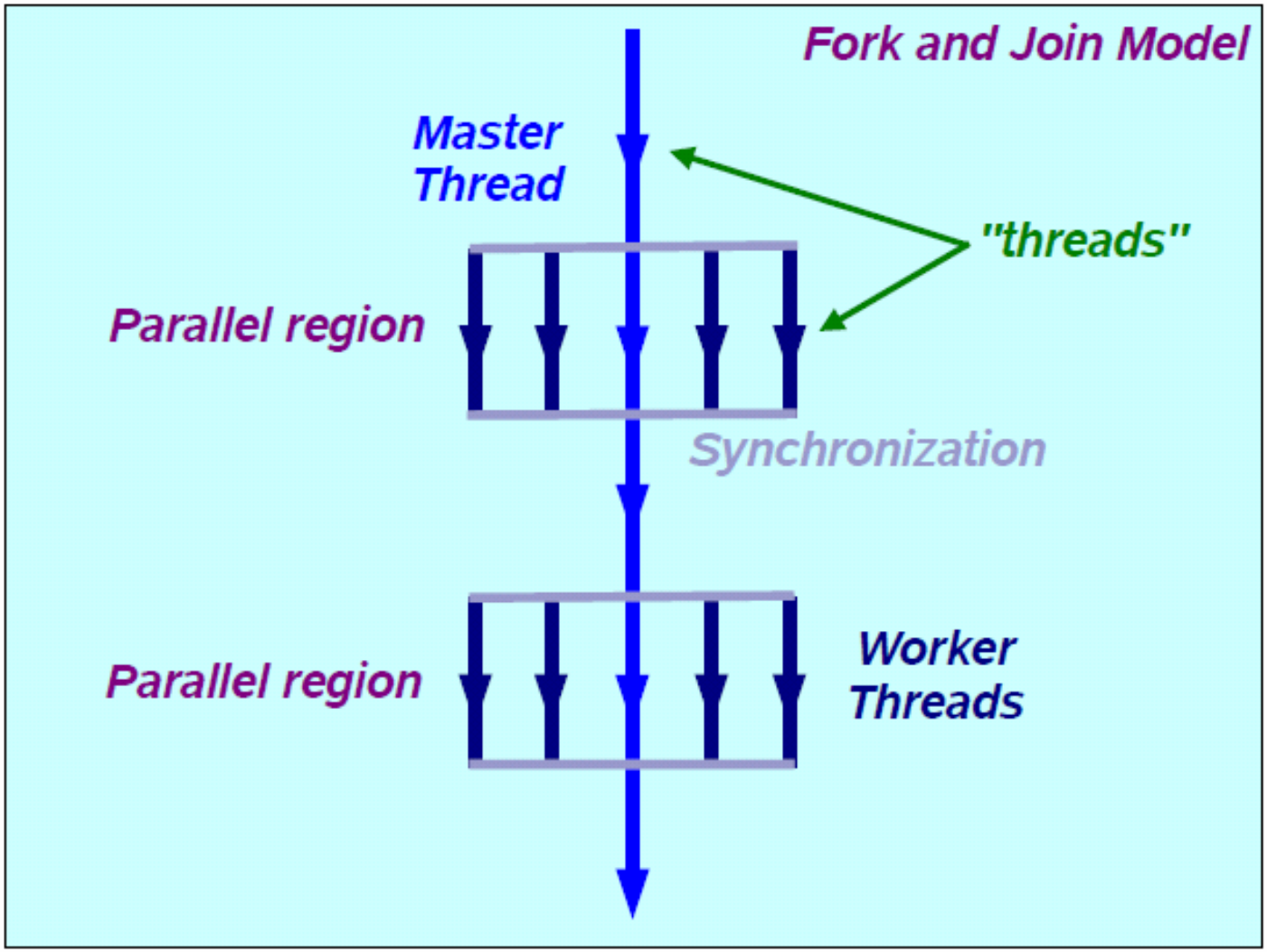
Light multi-threading

Chip Multi-Processing (CMP)

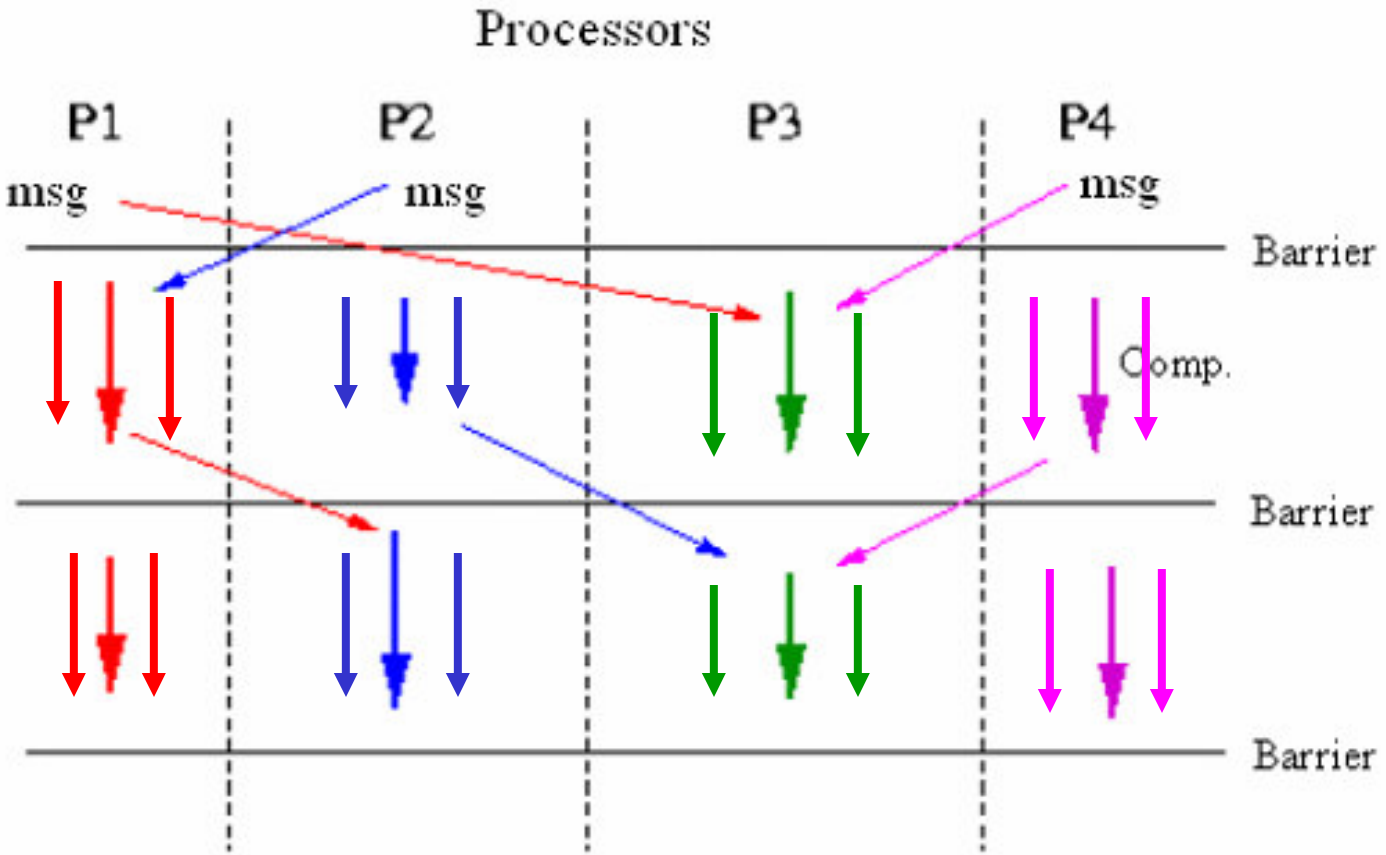



Parallel
Thread Execution
C
Compute Cycles
M
Memory Wait Cycles
Core 4
Core 3
Core 2
Core 1
Processor
Time

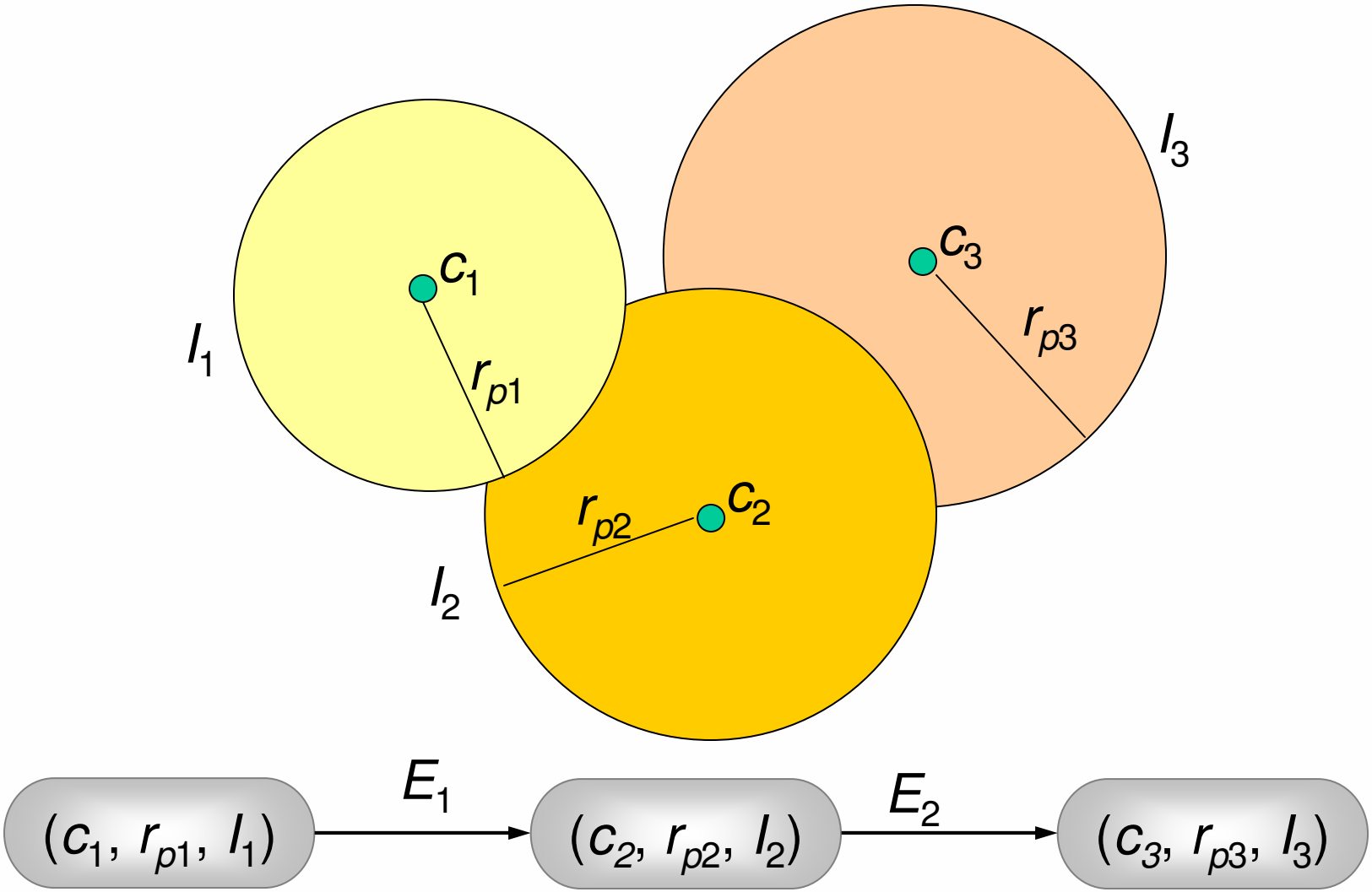
openMP model for in-core multi-threading

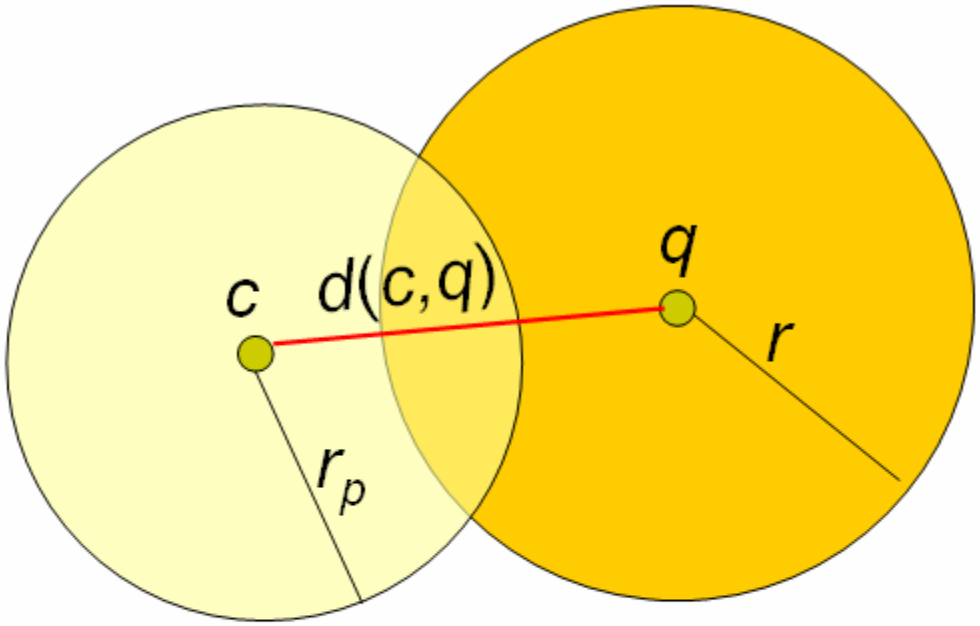


Multi-threading in each superstep and processor

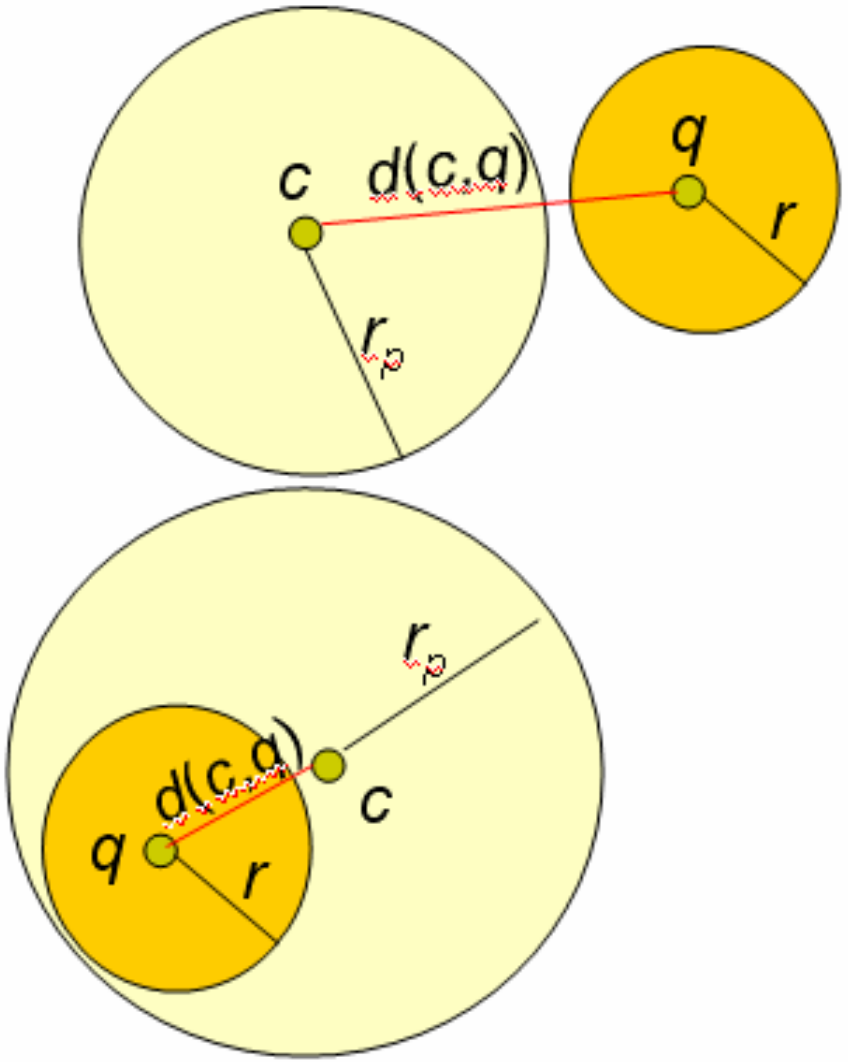


List of Clusters (LC)

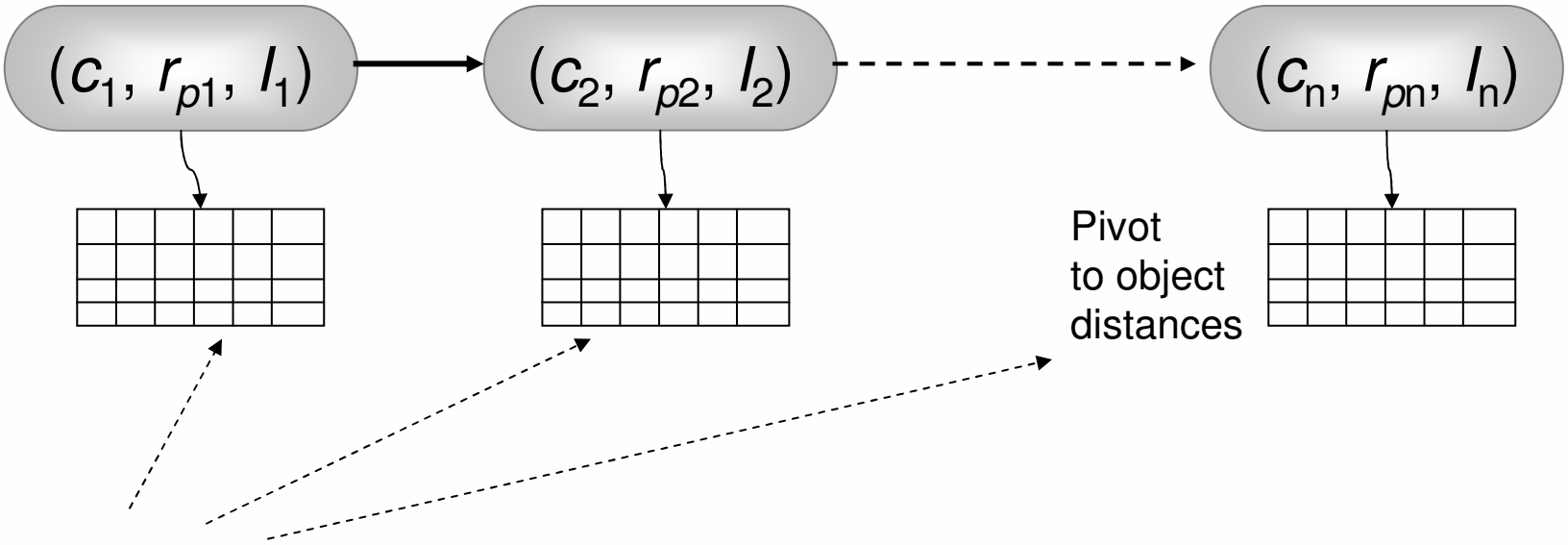




We search exhaustively in *l*, and continue searching in *E*


c
d(c,q)
q
r
r_c
r_c
d(c,q)
c
q
r

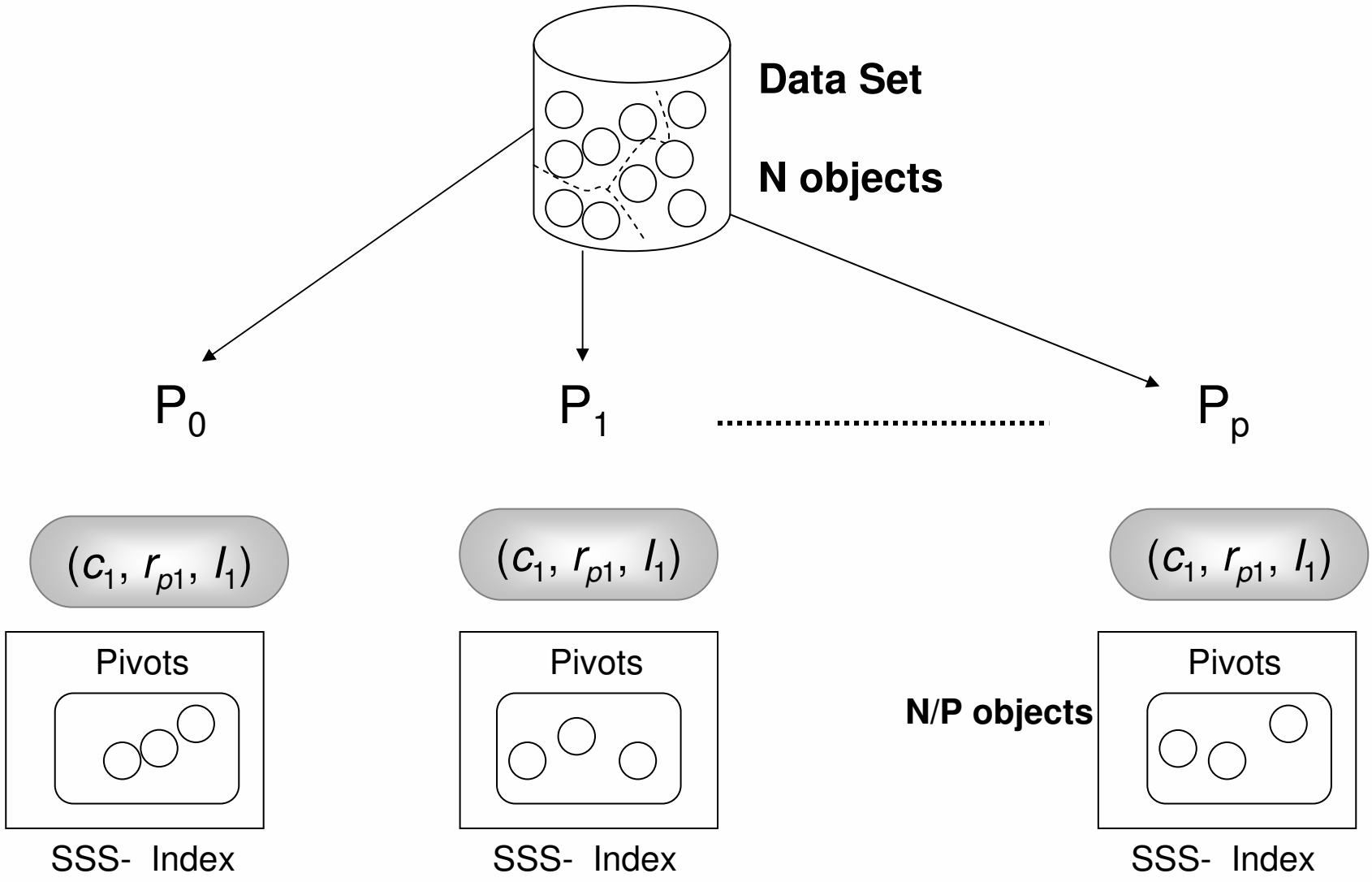
Buckets using Spatial Sparce Selection (SSS)



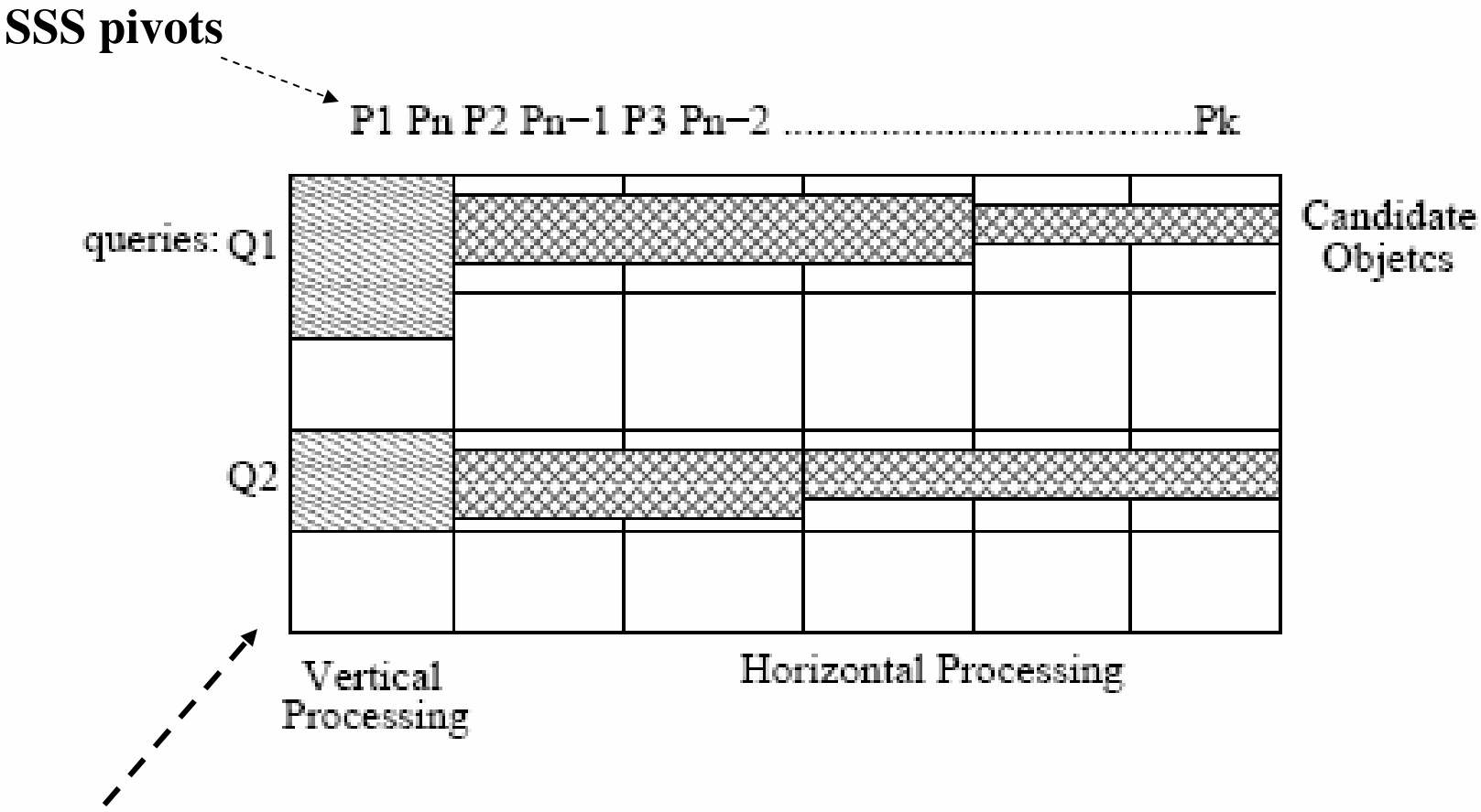
The same SSS pivots in each table (bucket).

Also the same ordering of pivots in the table columns, the first two are the most distant ones, and so on.

Each processor contains the same LC centers and SSS pivots, objects are the distributed ones.



Multi-threaded query processing



The first column is ordered by distance to the first pivot
so two binary searches per query determines the range of rows
where candidate objects to be compared with the query can be found.

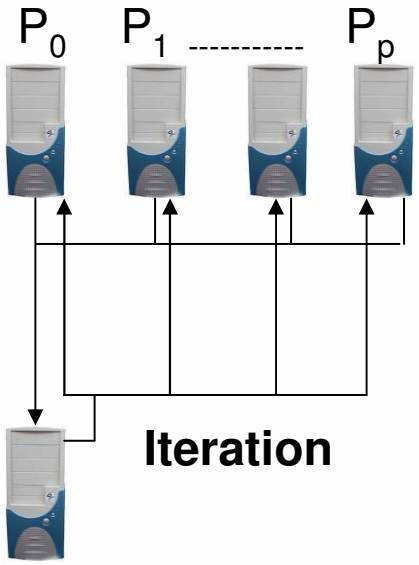
Experiments

We performed experiments using a 32-processors cluster and different data sets and queries. We use two multimedia data sets. The first one is a collection of 47,000 images extracted from the NASA photo and video archives, each of them transformed into a 20-dimensional vector. The Euclidean distance is the distance function used in this data set. The second one is a large set of 900,000 words taken from documents crawled from the Chilean Web.

Index construction

database evenly distributed among processors and the first center selected at random

P1



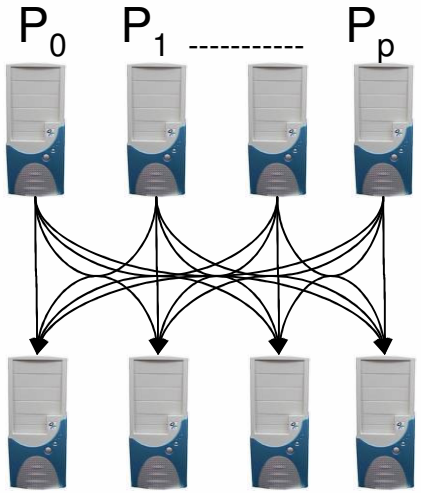
Compute distances to local objects
Selects a new center
Send new center and its sum to P_0

Selects the center maximizing the sum of distance computed

Broadcast the new center

Index construction

P2

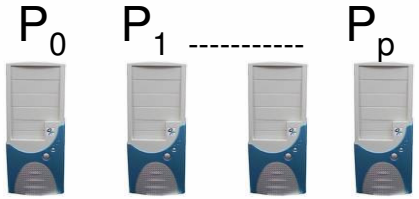


Selects its candidate centers locally then broadcast to all.

Computes distance between the local centers

Selects the ones maximizing the sum of distance

Index construction

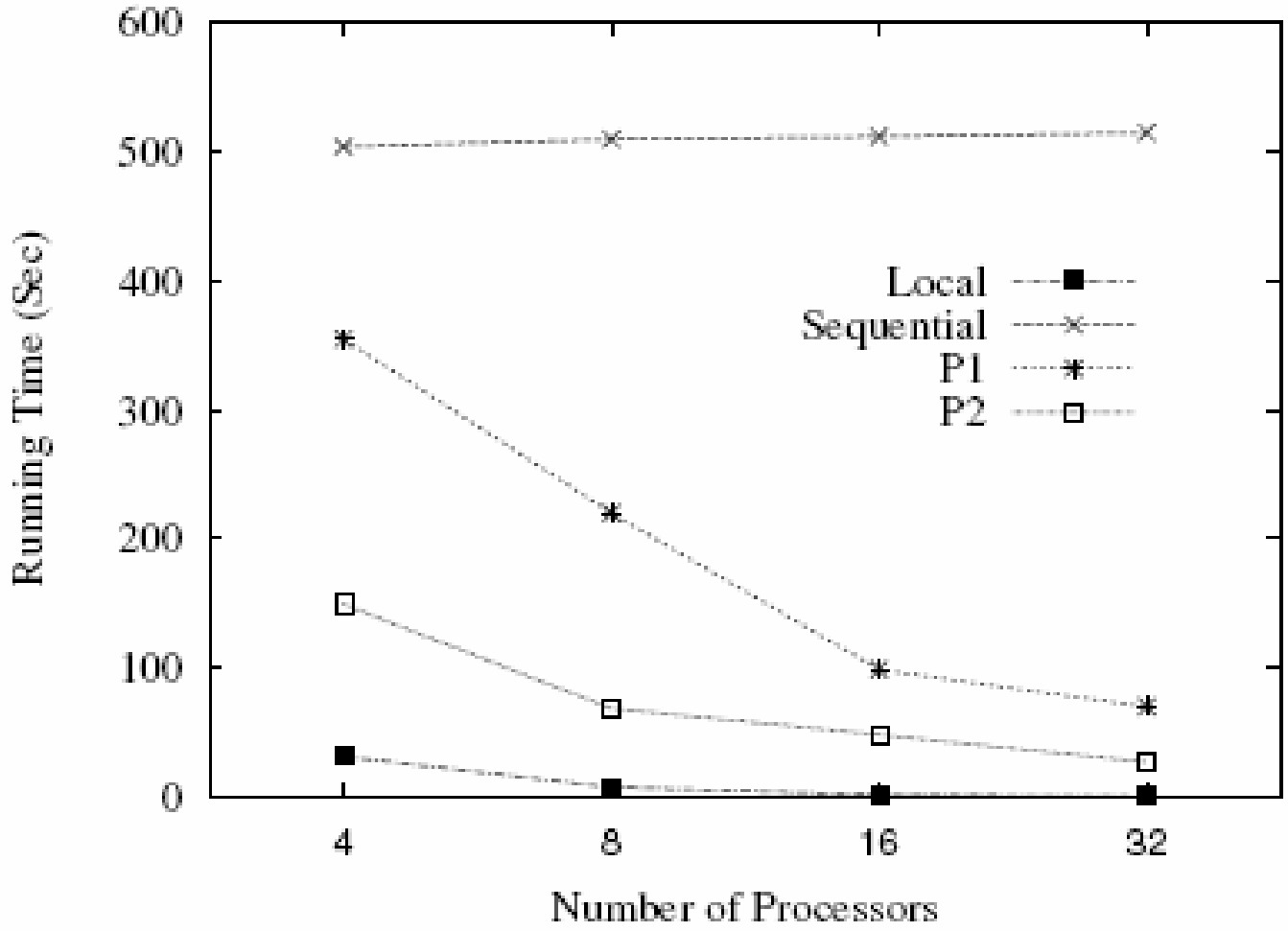


Each processor builds its own local index using its local objects

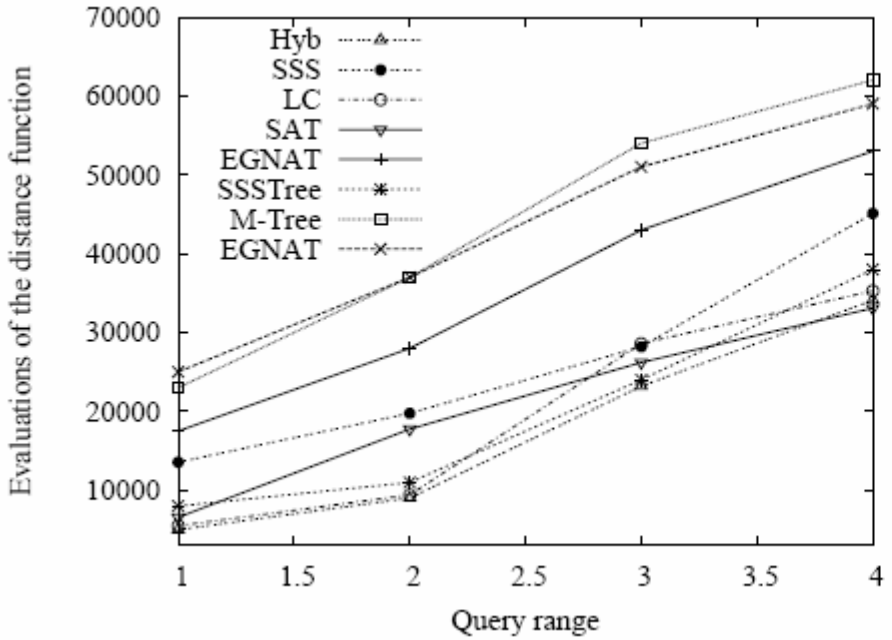
Local

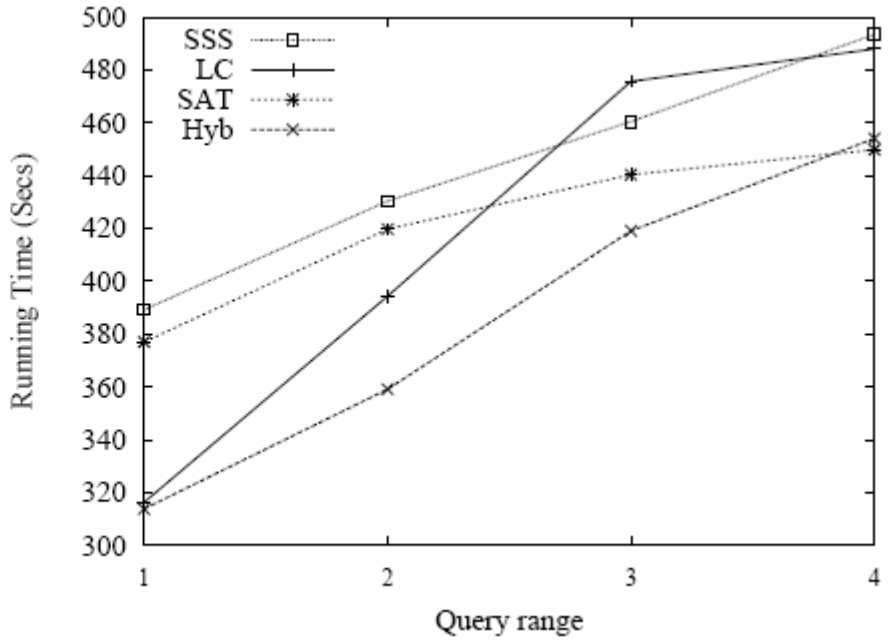
Easy to build and update: No communication is required among processors during these operations.

Index construction



Sequential performance



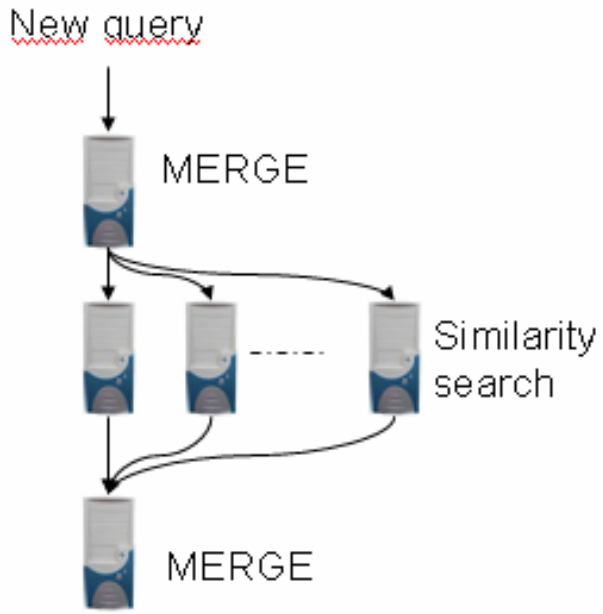


T	1	2	4	6	8	10	12	14	16
$(T=1)/(T\geq 1)$	1.00	1.76	2.12	2.81	4.07	3.52	2.01	0.88	0.93

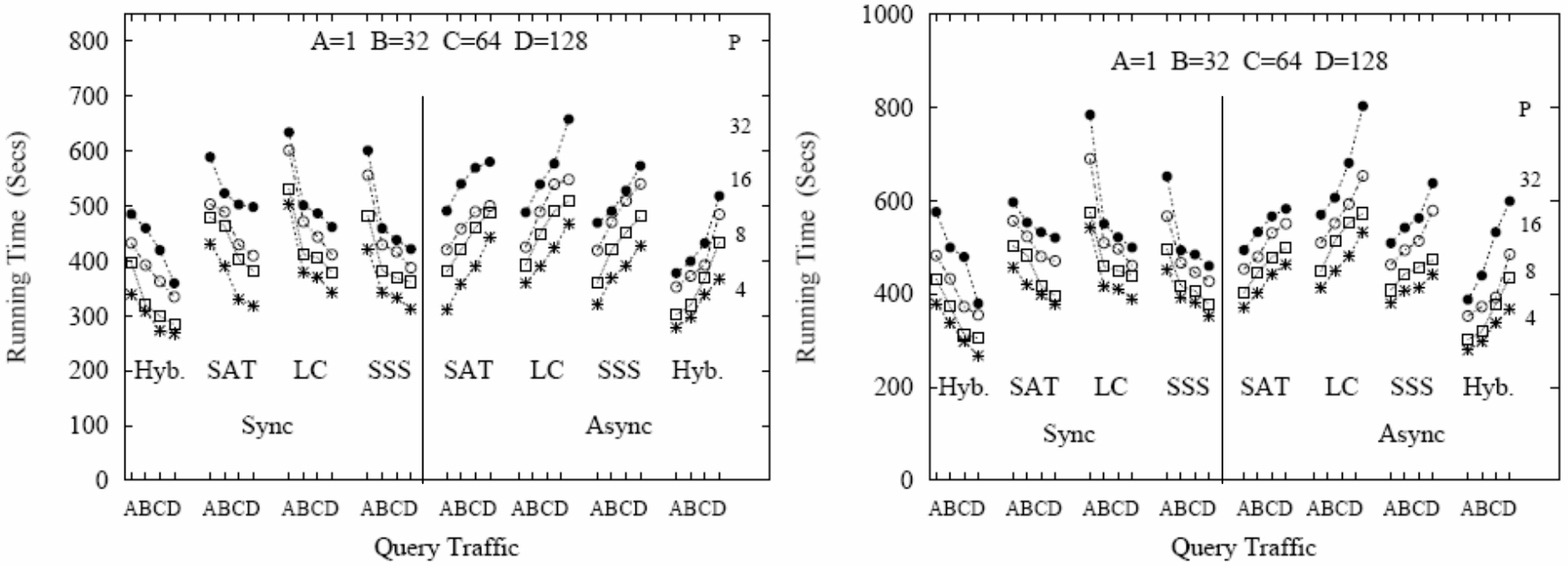
Decreasing running times with openMP threads.

Parallel query processing

- The broker machine selects the MERGE processor and send a new query.
- The MERGE processors performs a broadcast.
- Each processor search similar objects using its local LC index and sends the results to the MERGE machine.
- The MERGE machine collects the results and send them to the broker.



Parallel performance



Running time obtained using two data collections.

Conclusions

Efficient and scalable performance is obtained because of index data structures devised to increase locality in space and time.

The bulk-synchronous organization of parallel query processing allows proper control of hardware resources.

Questions ?