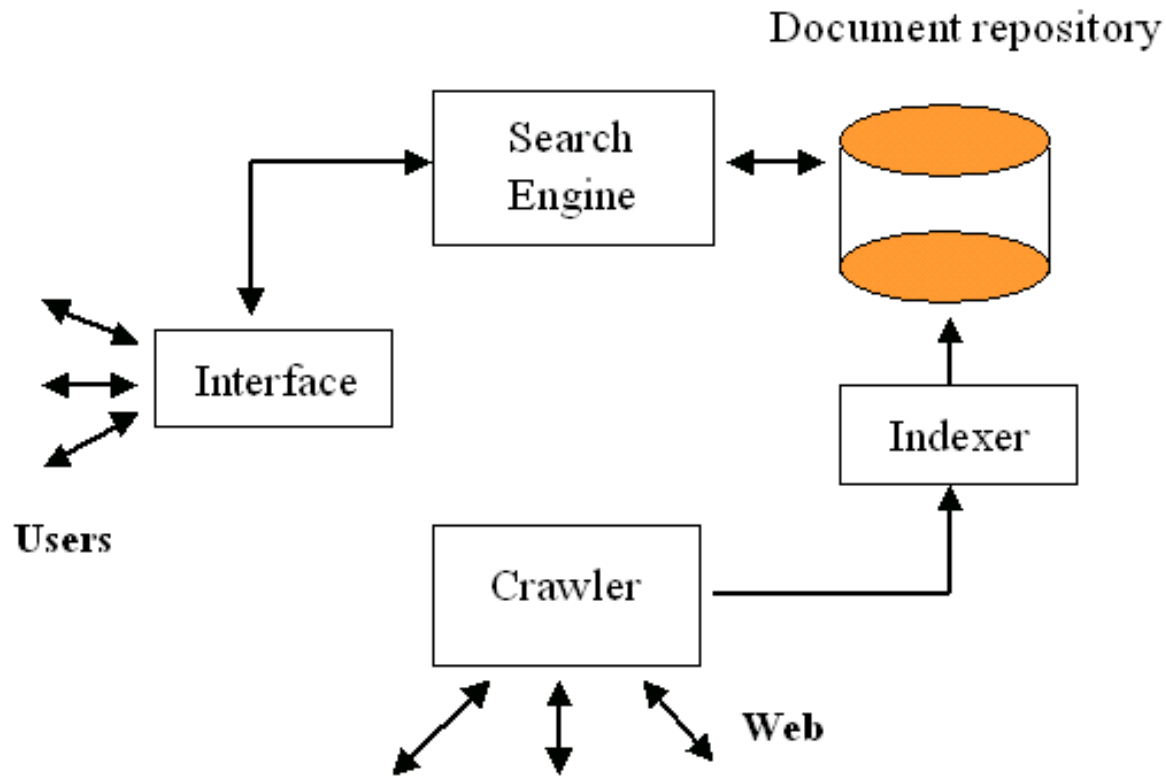


Computación Paralela y Aplicaciones

Primavera 2011

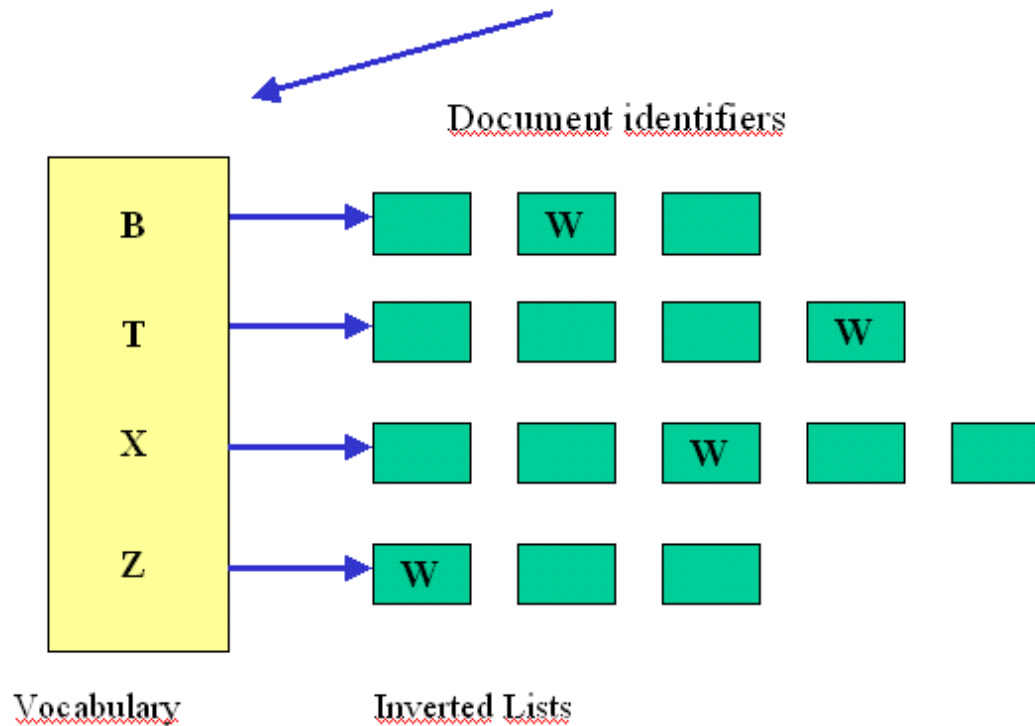
- Arquitectura para motores de búsqueda
- Técnicas de paralelización para sistemas de memoria distribuida
- Procesamiento de consultas.

Arquitectura de Motores de Búsqueda

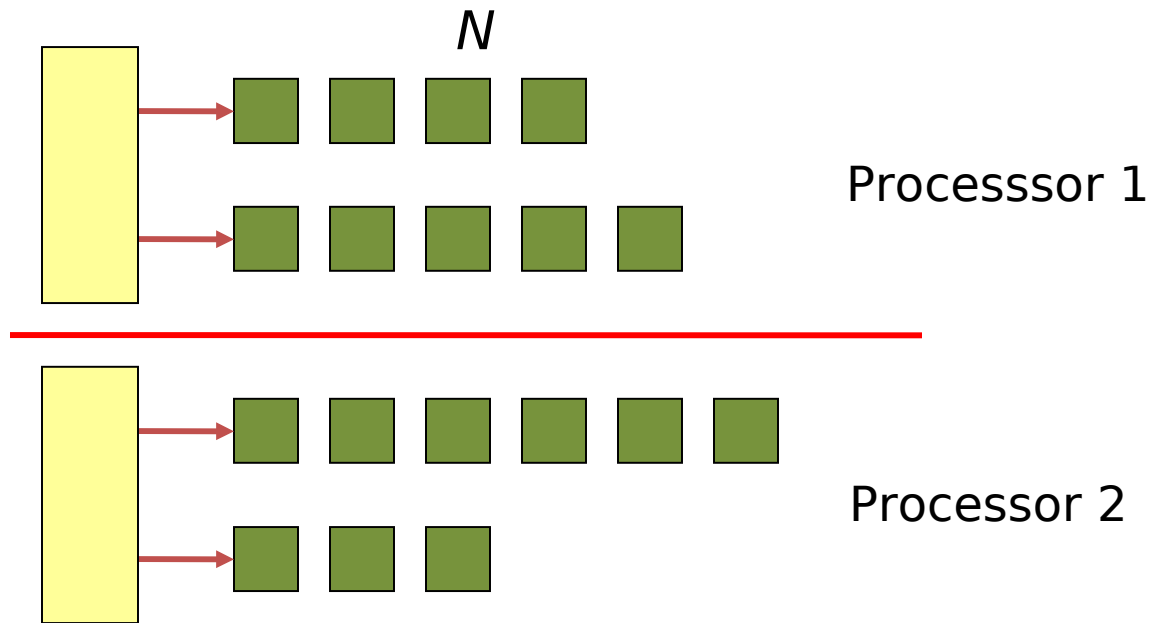


Indice Invertido

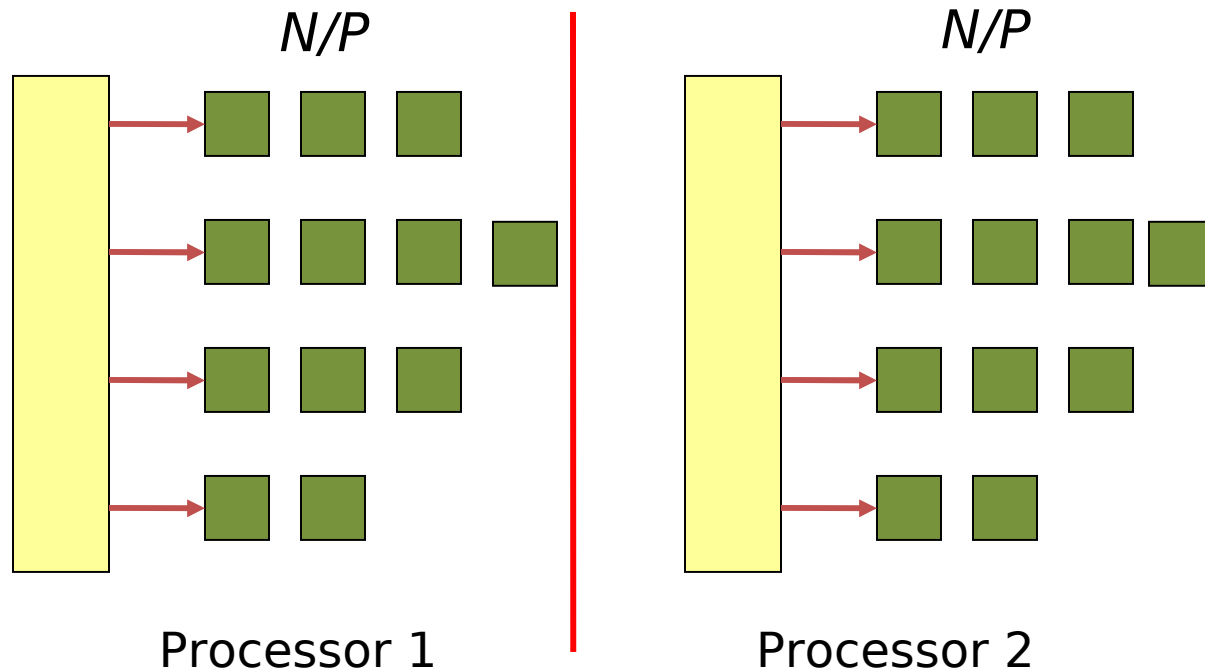
Query = B and T and X and Z

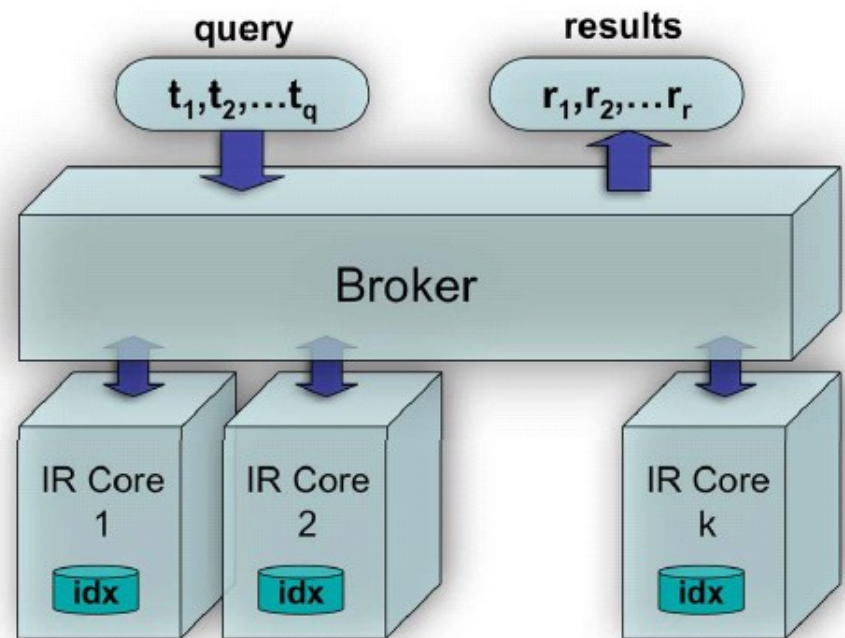
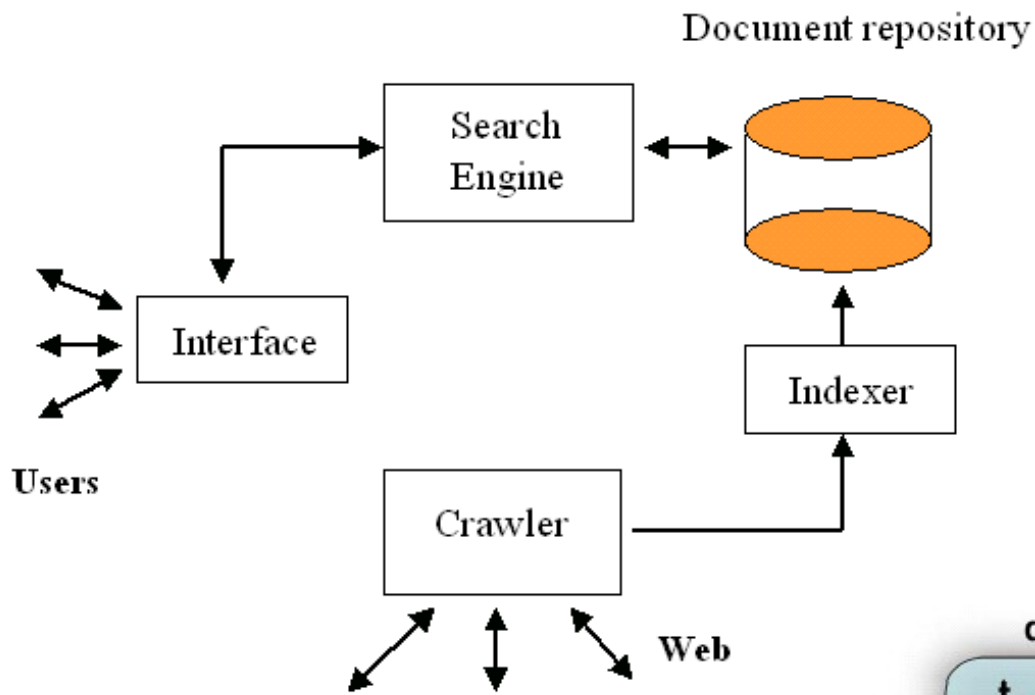


Distribución: índice por términos

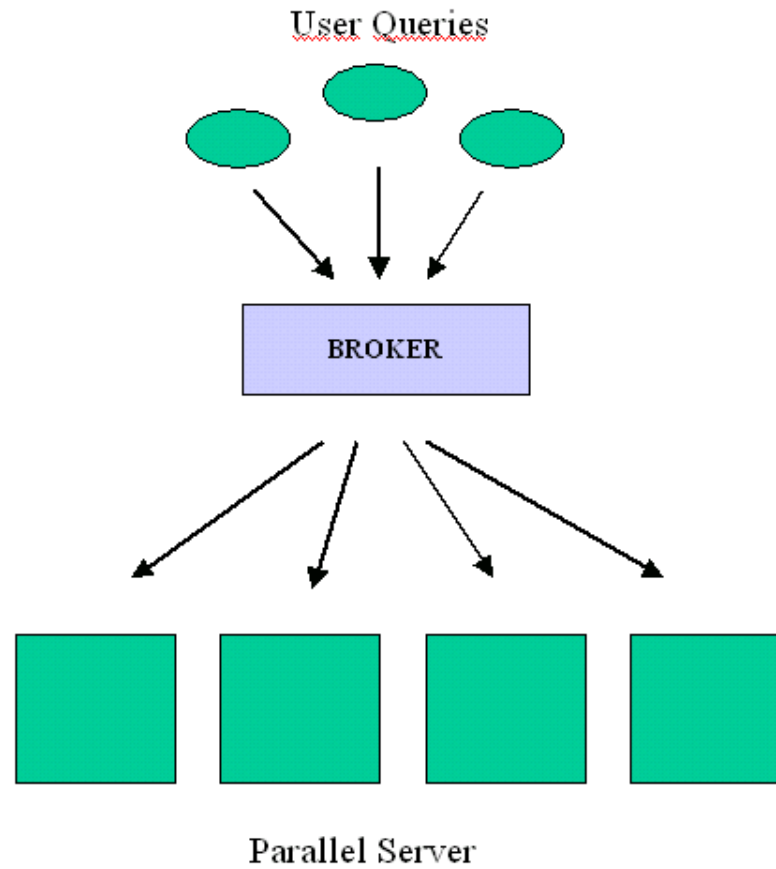


Distribución: índice por documentos

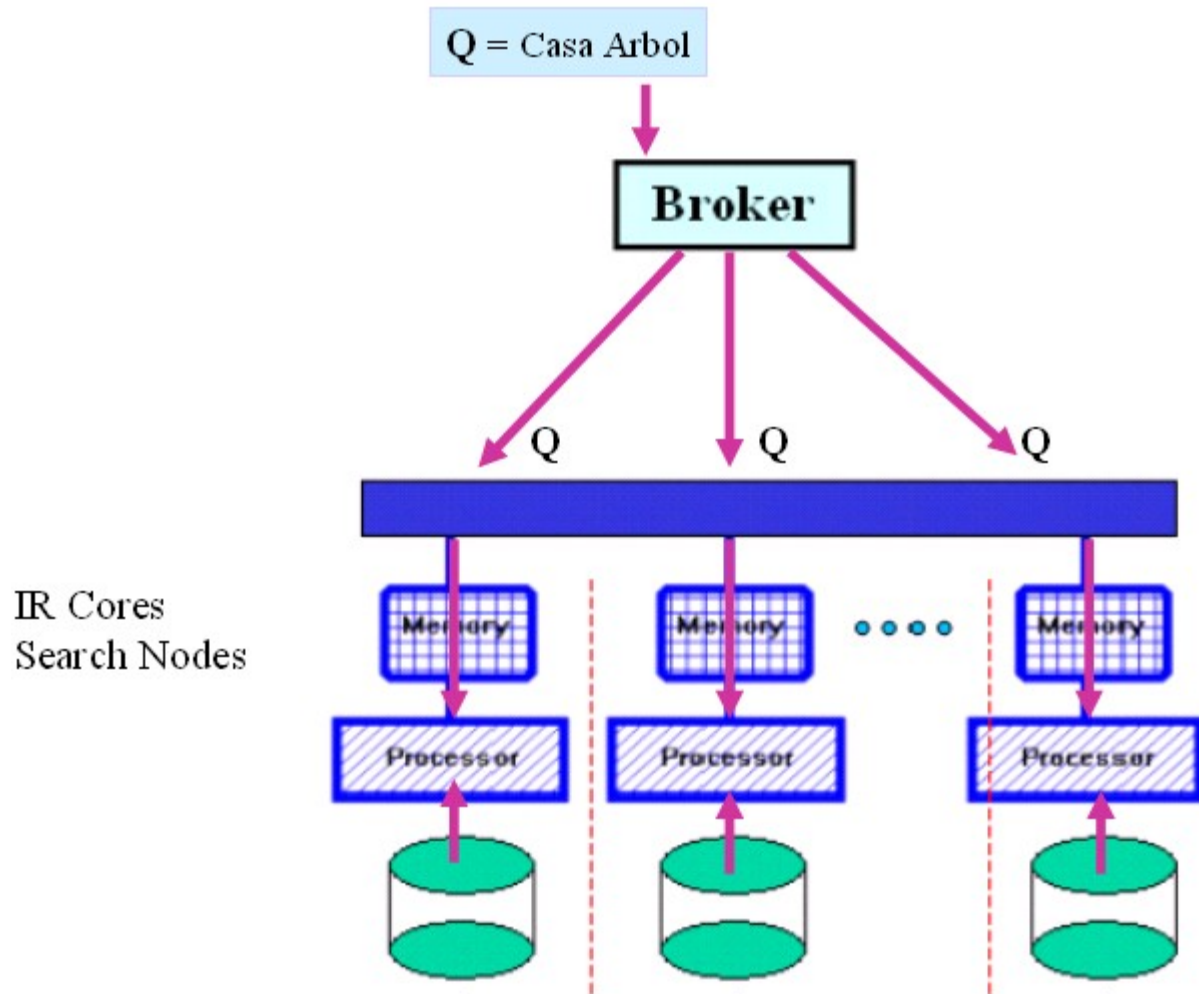




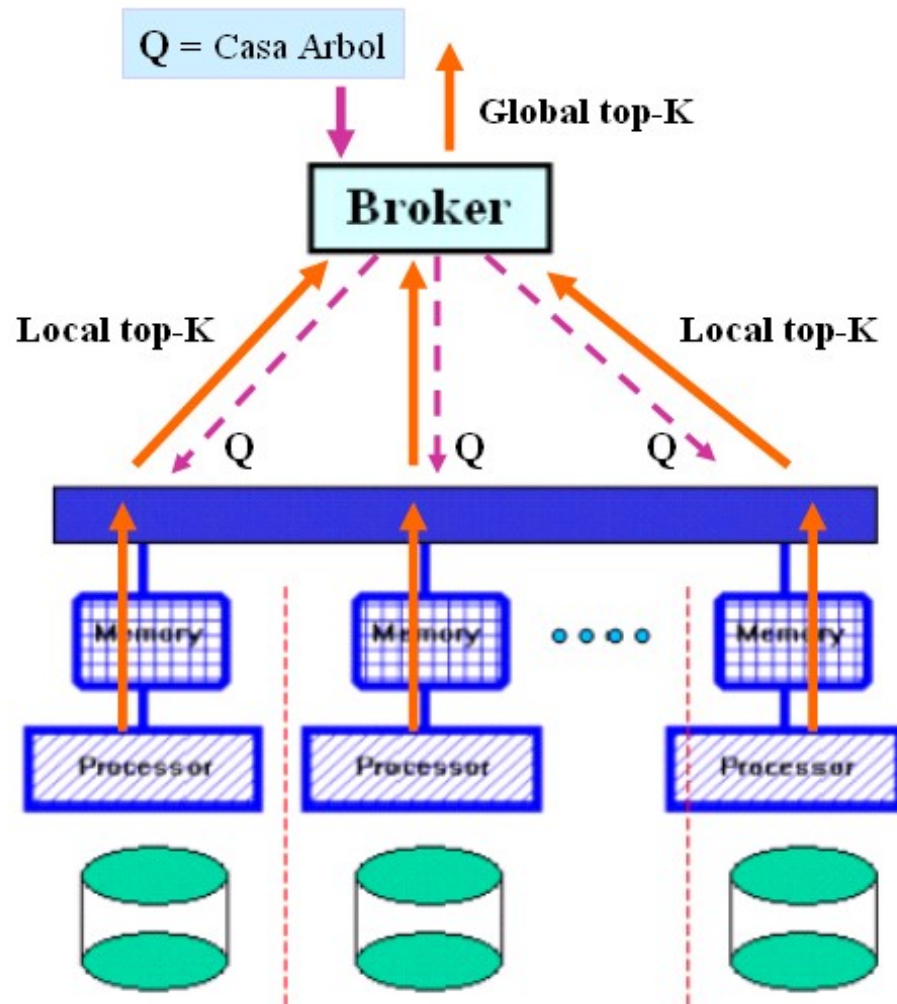
Resolver una consulta



Resolver una consulta

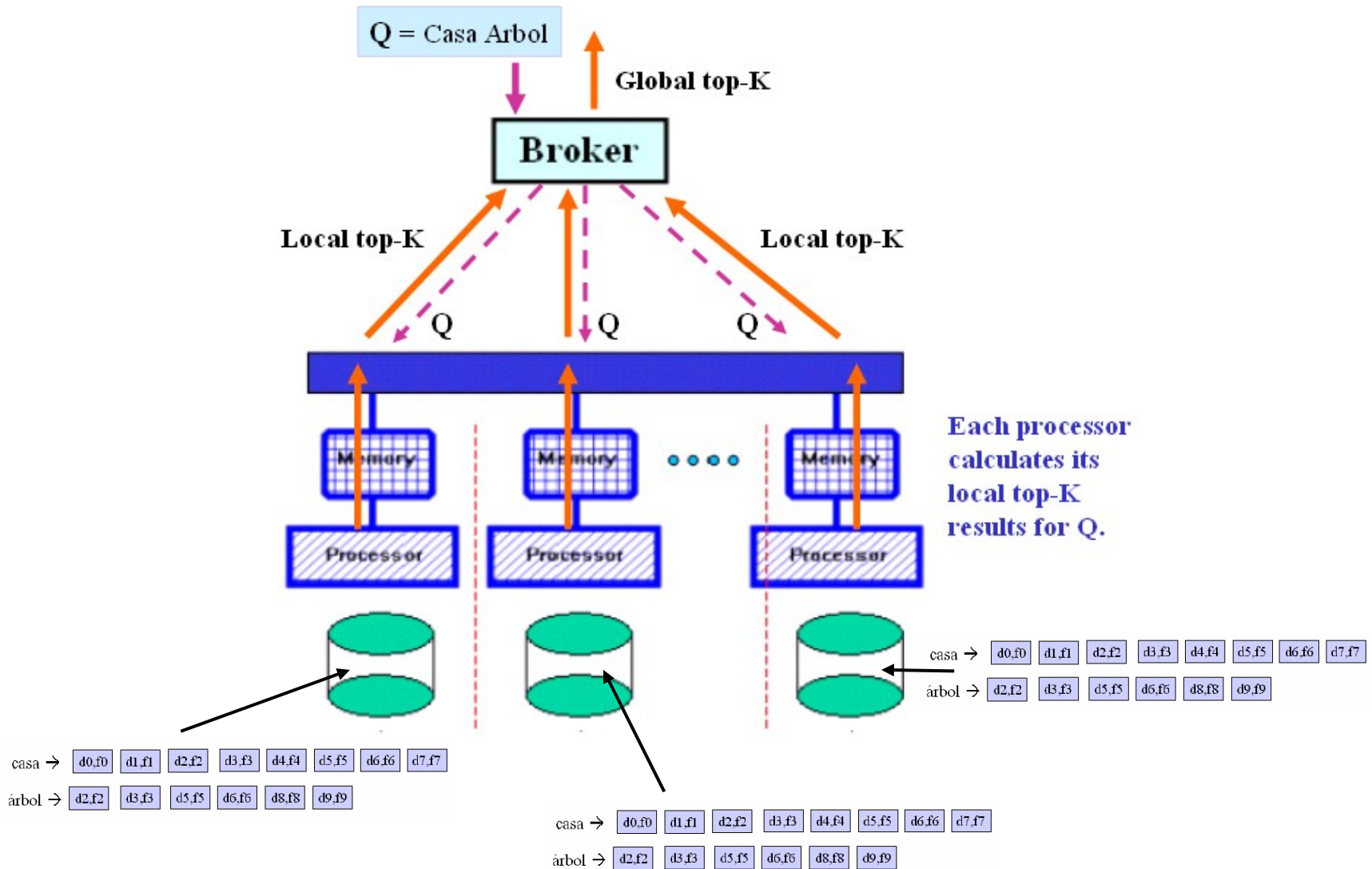


Resolver una consulta

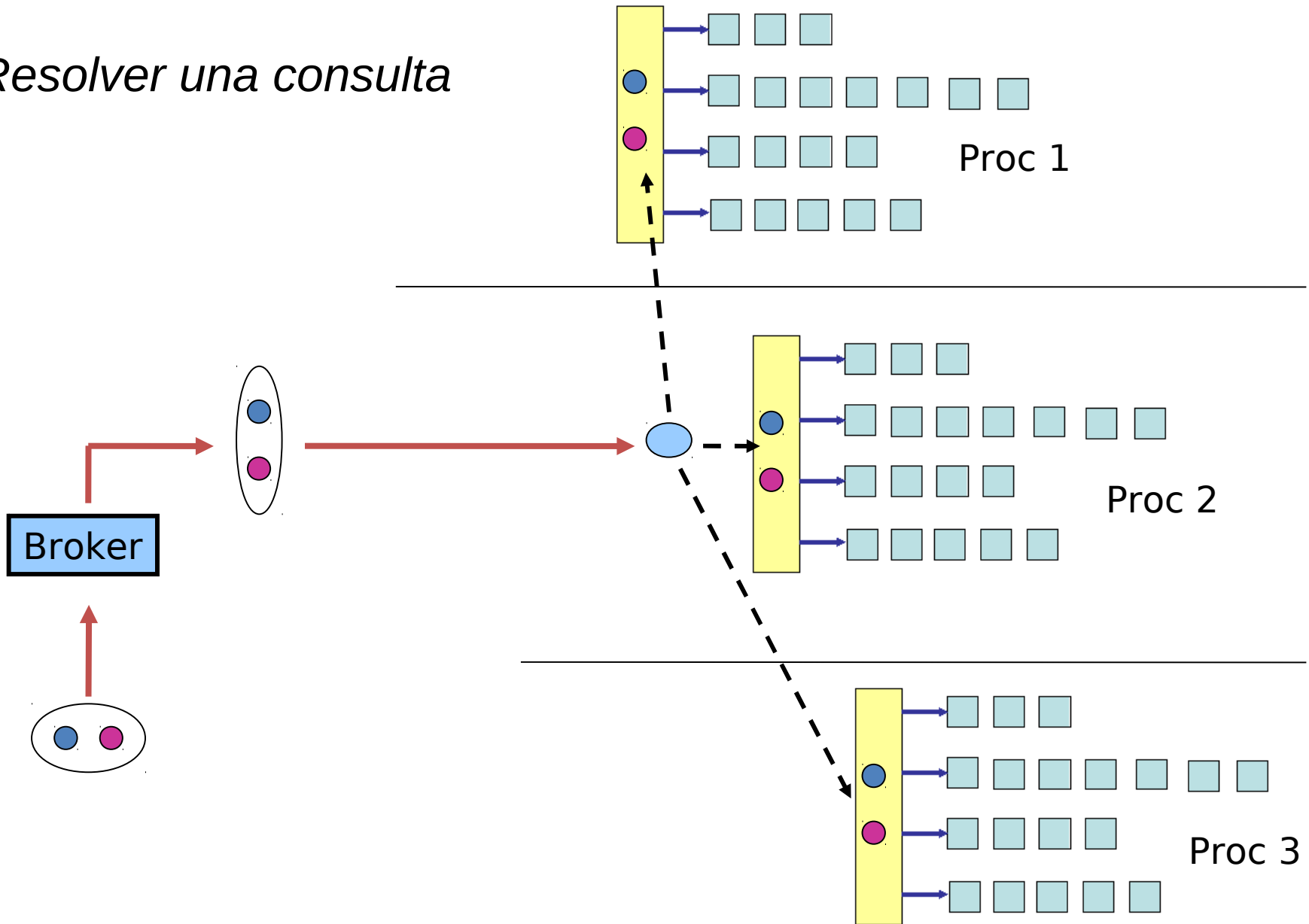


Each processor calculates its local top-K results for Q.

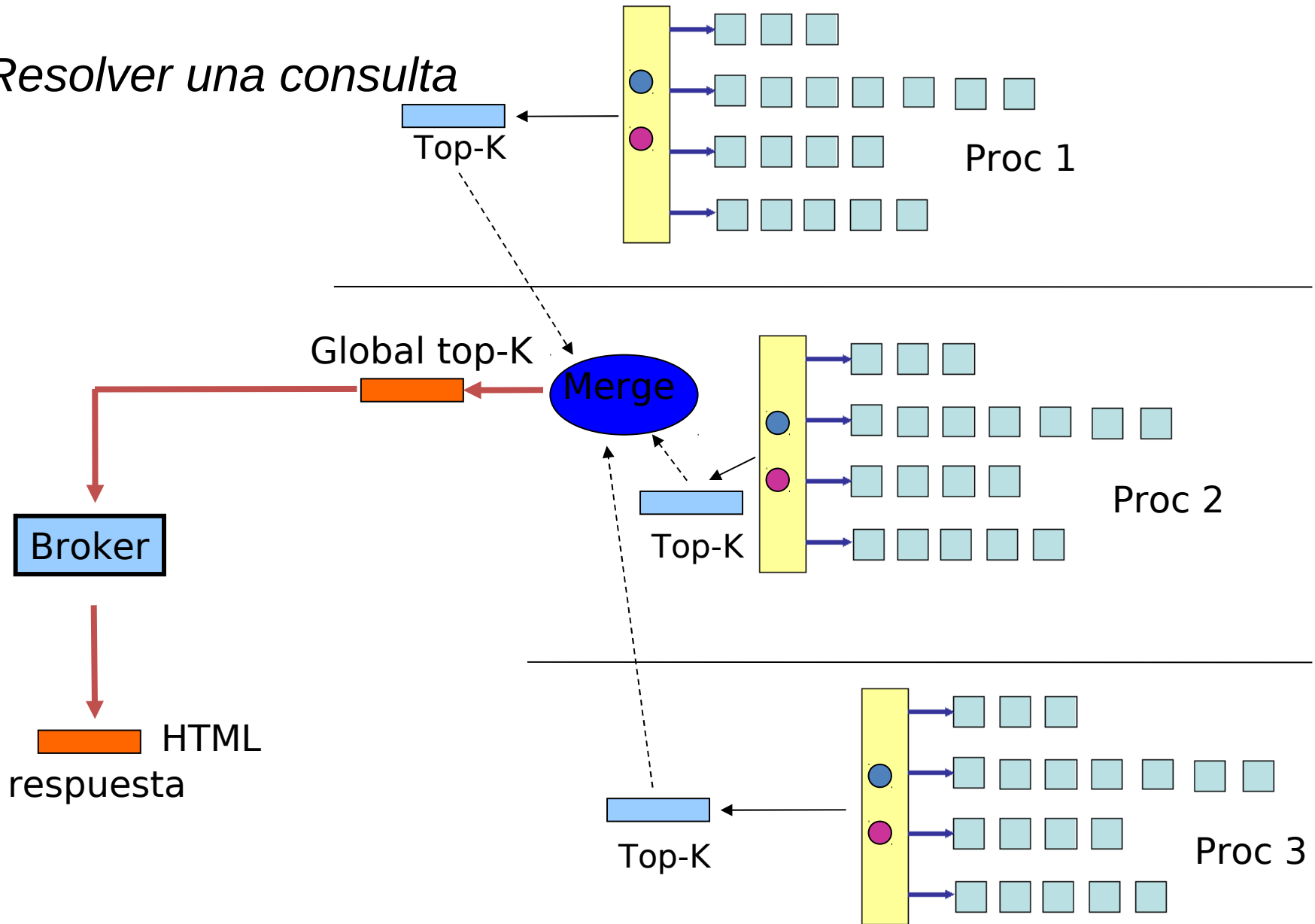
Resolver una consulta



Resolver una consulta

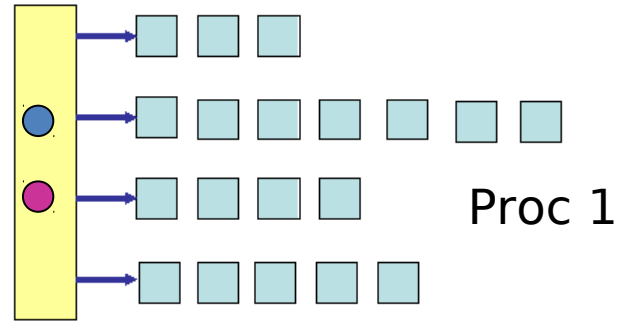


Resolver una consulta



Resolver una consulta

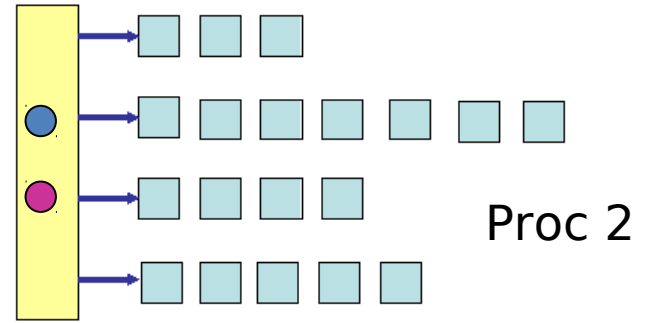
Top-K



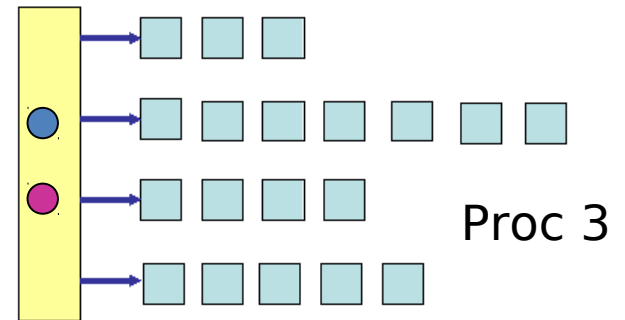
Global top-K

Merge

Top-K



Top-K

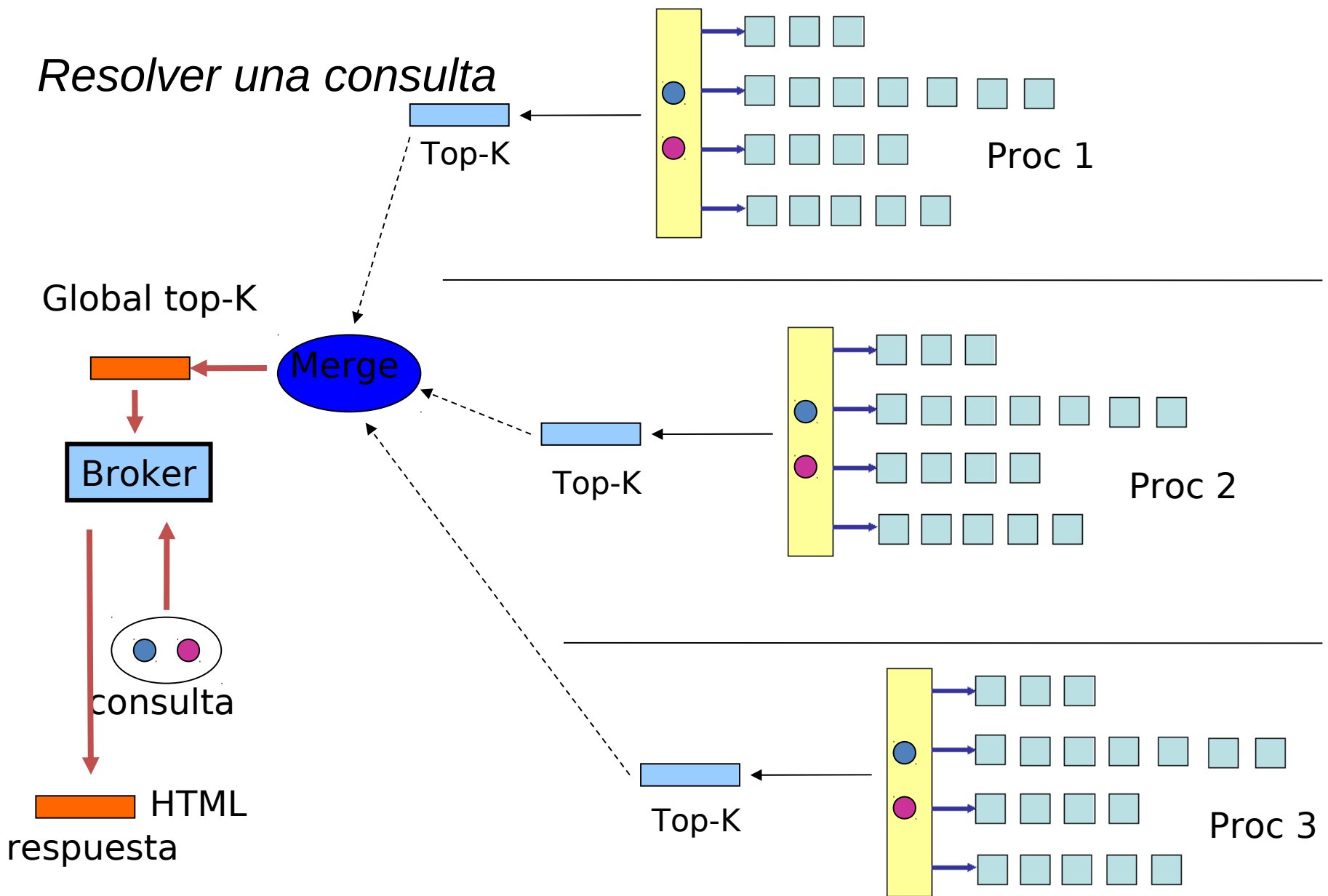


Global top-K

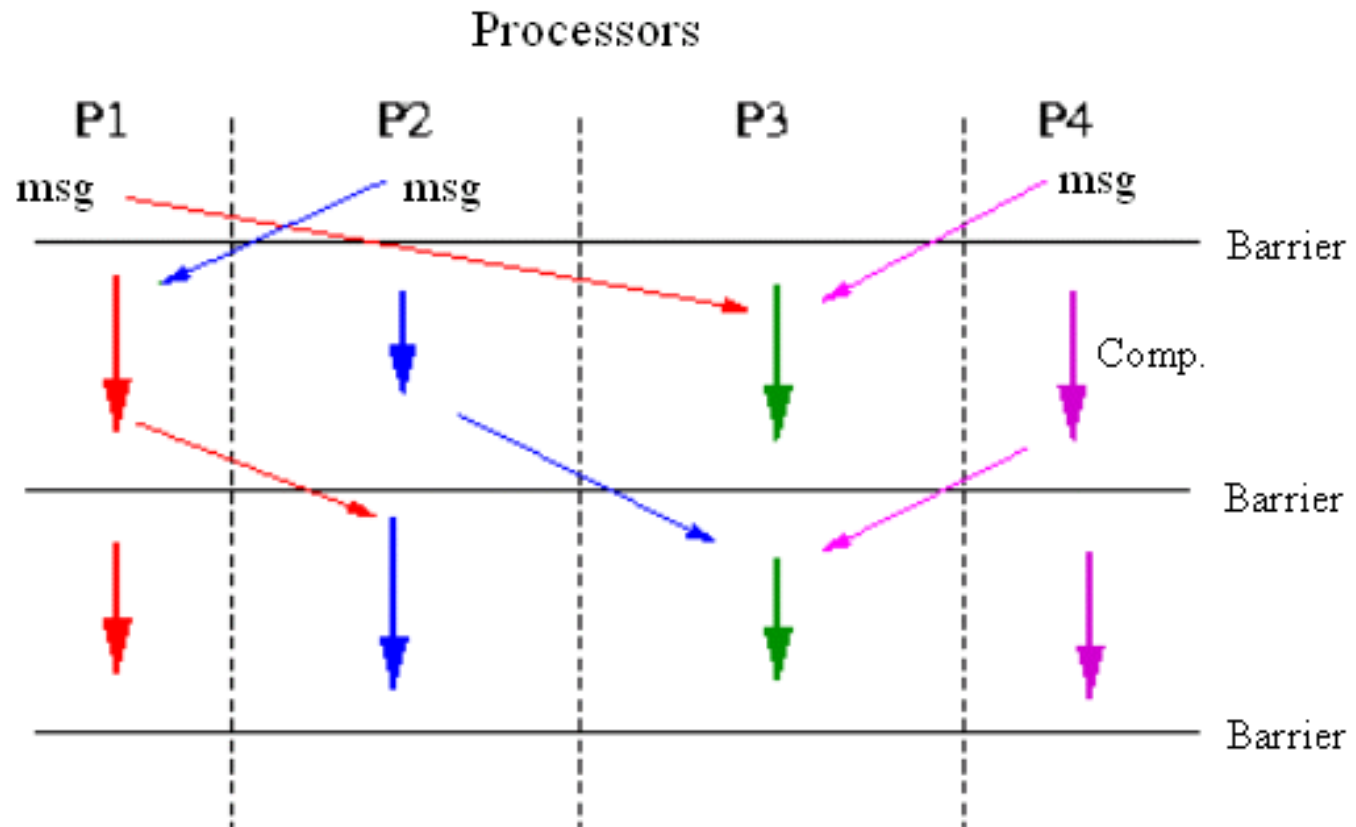
Broker

consulta

HTML
respuesta

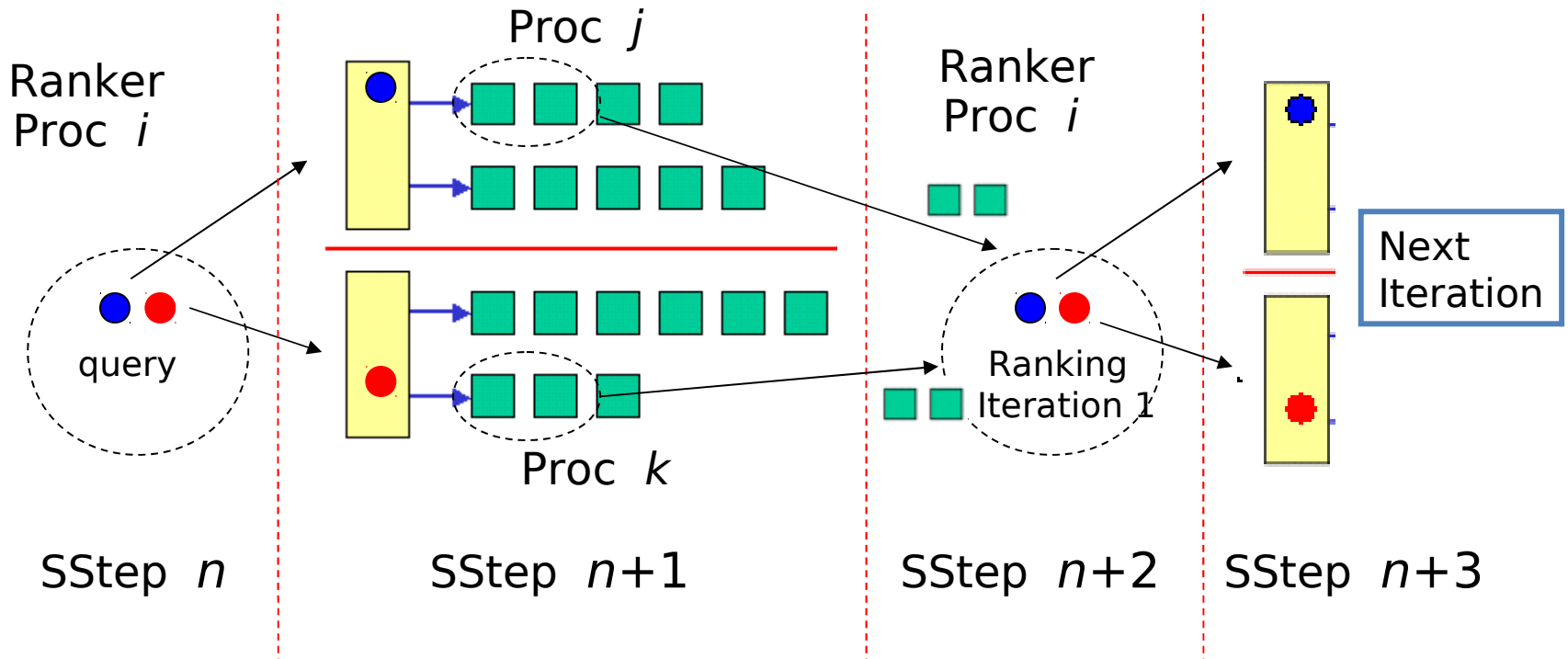


SuperSteps

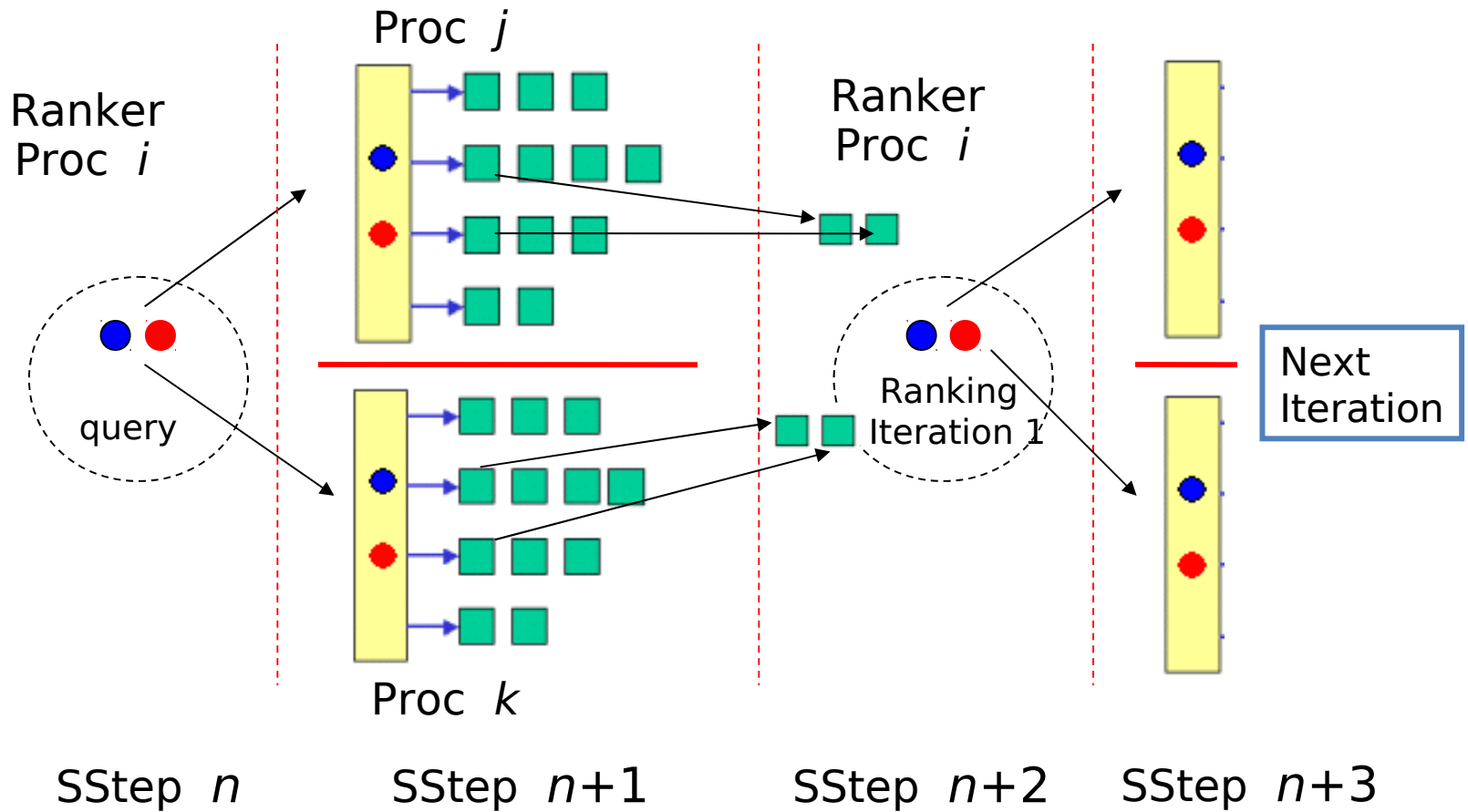


Paradigm of parallel computation (PVM MPI BSPpub BSPLib).

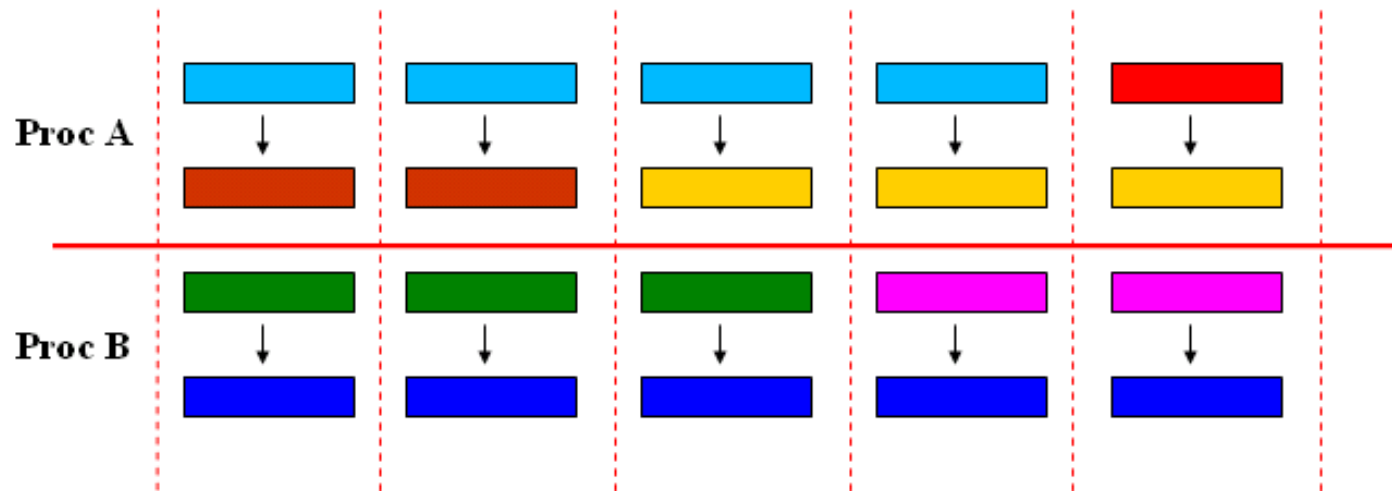
Resolver una consulta, distribución por términos



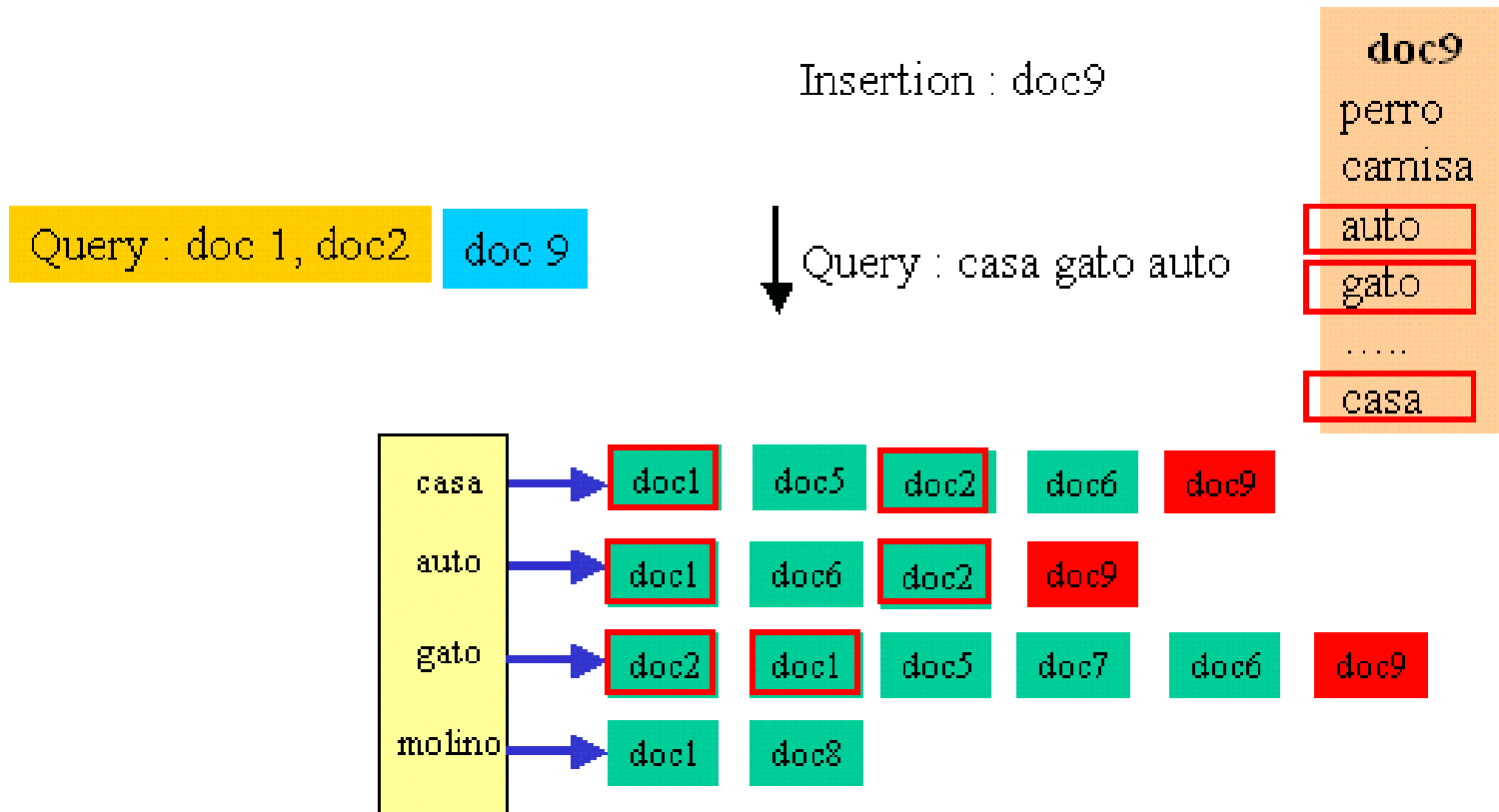
Resolver una consulta, distribución por documentos



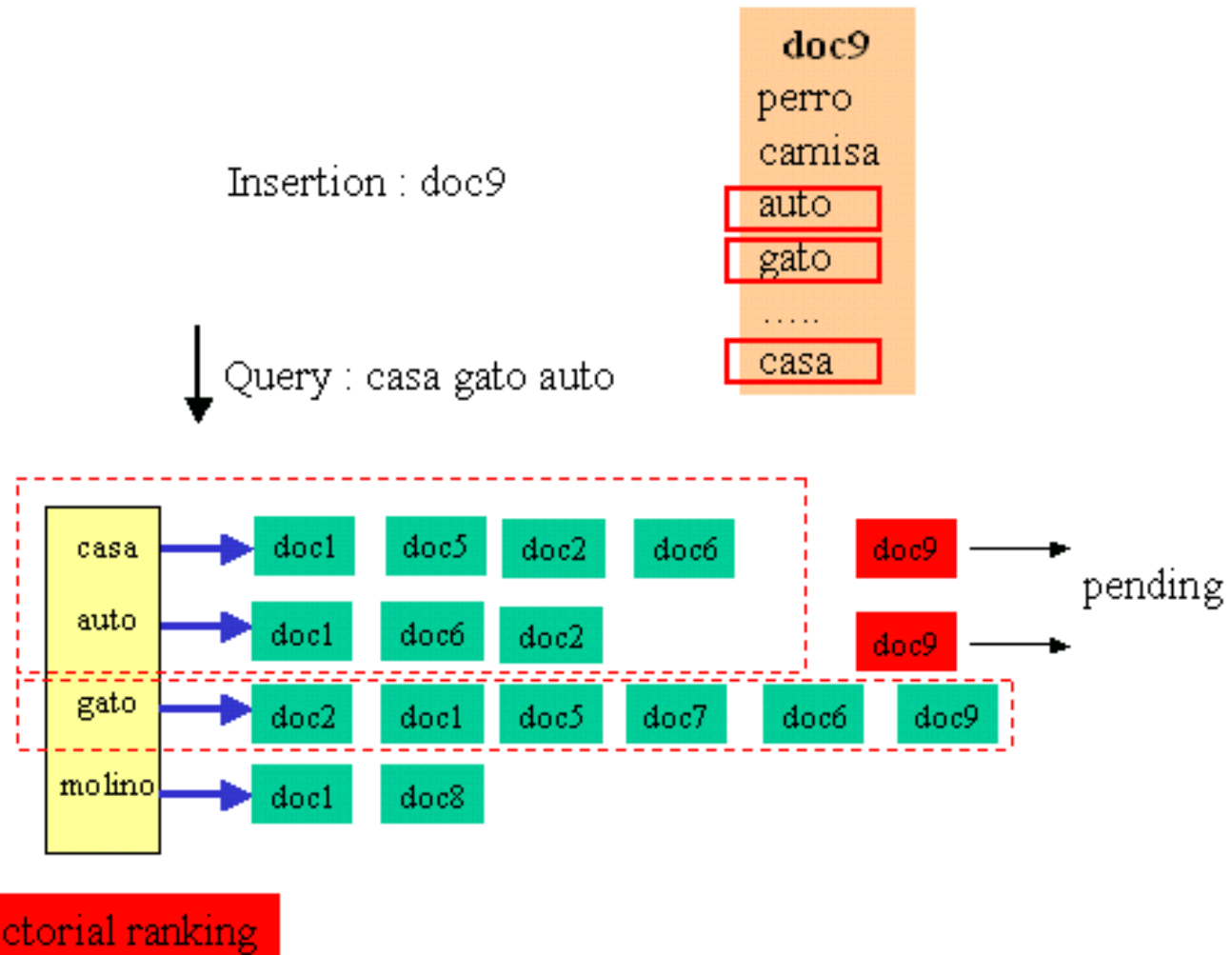
Resolver una consulta, Round-robin



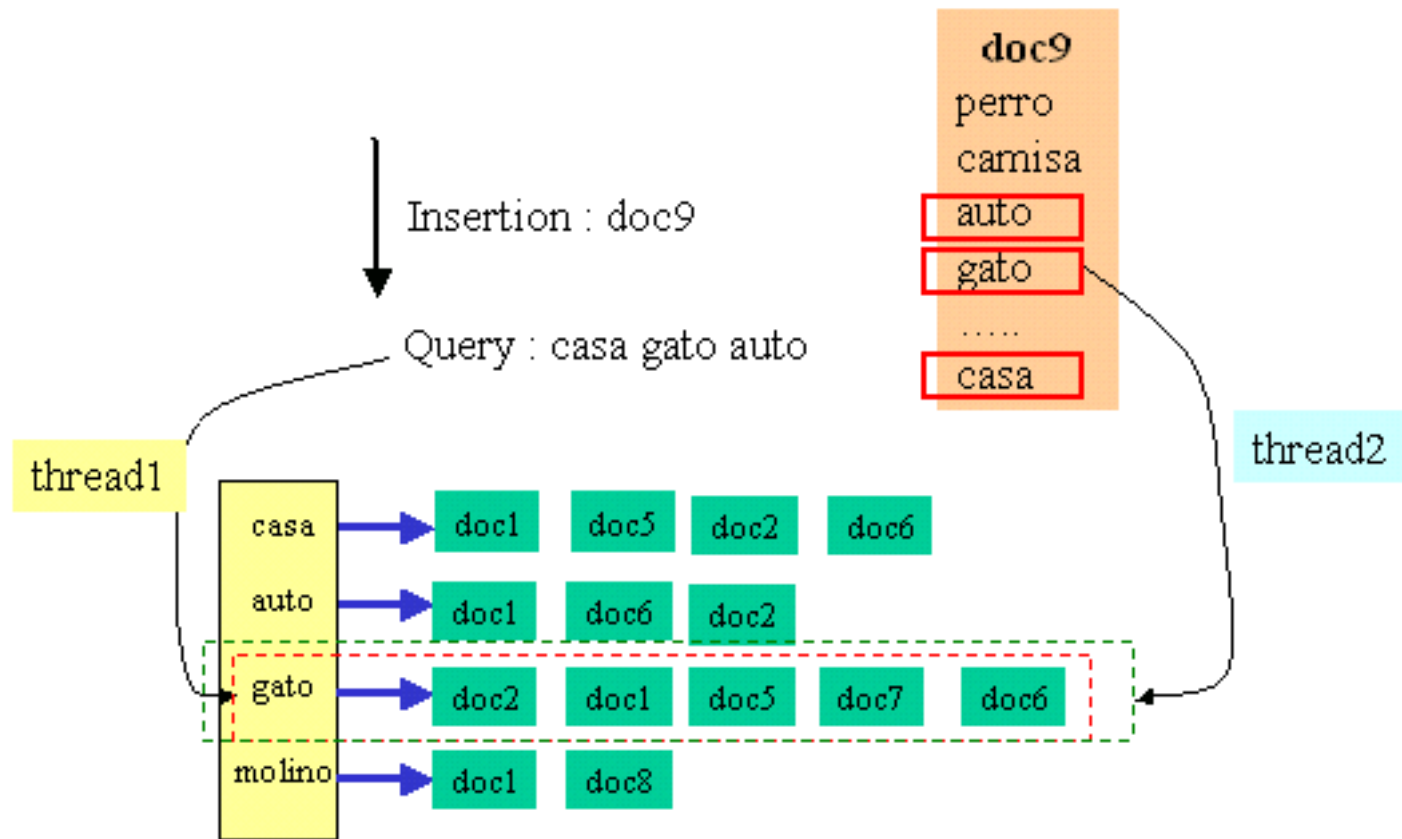
Control de Concurrency



Control de Concurrencia



Control de Concurrency



Solución: Protocolo de locks de dos fases

Thread Q₁

- Solicita lock de lectura para todos sus términos T₁, T₂ y T₃
- Pre-ranking
- Libera locks
- Ranking Final

Colisiones de términos

Thread D₁

- Solicita locks de escritura para todos los términos de D₁
- Actualiza las listas invertidas
- Libera todos los locks

Pide muchos locks

¿Que ocurre si hay iteraciones?
Se debe esperar a resolver toda la consulta antes de liberar locks

Requerimiento de Casualidad

- No siempre es posible, solo para casos en que hay Broker.
- El broker asigna timestamp a las consultas e inserciones de documentos

Modo sincrónico

BSP + PQs de mensajes ordenados por timestamp

En cada paso se atienden en orden de timestamp.(no ocurren problemas de causalidad, ya que al inicio de cada sstep no hay mensajes en tránsito)

Modo asincrónico

El broker asigna uso exclusivo del cluster a los escritores

Detecta cuando han terminado todos los lectores con timestamp inferior y le da el paso al escritor.

ESTRATEGIAS DE CONTROL

TimeStamp Sincrónico (TS)

SuperSteps Exclusivos (SE)

Protocolo de Locks Distribuidos

Protocolo Optimista (óptimo)

TimeStamp Concurrentes

- El broker asigna un timestamp único a cada consulta y documento que recibe.
- Se utiliza como identificador por los threads de la máquina de búsqueda.
- Esta estrategia se usa cuando la máquina de búsqueda opera en modo sincrónico.

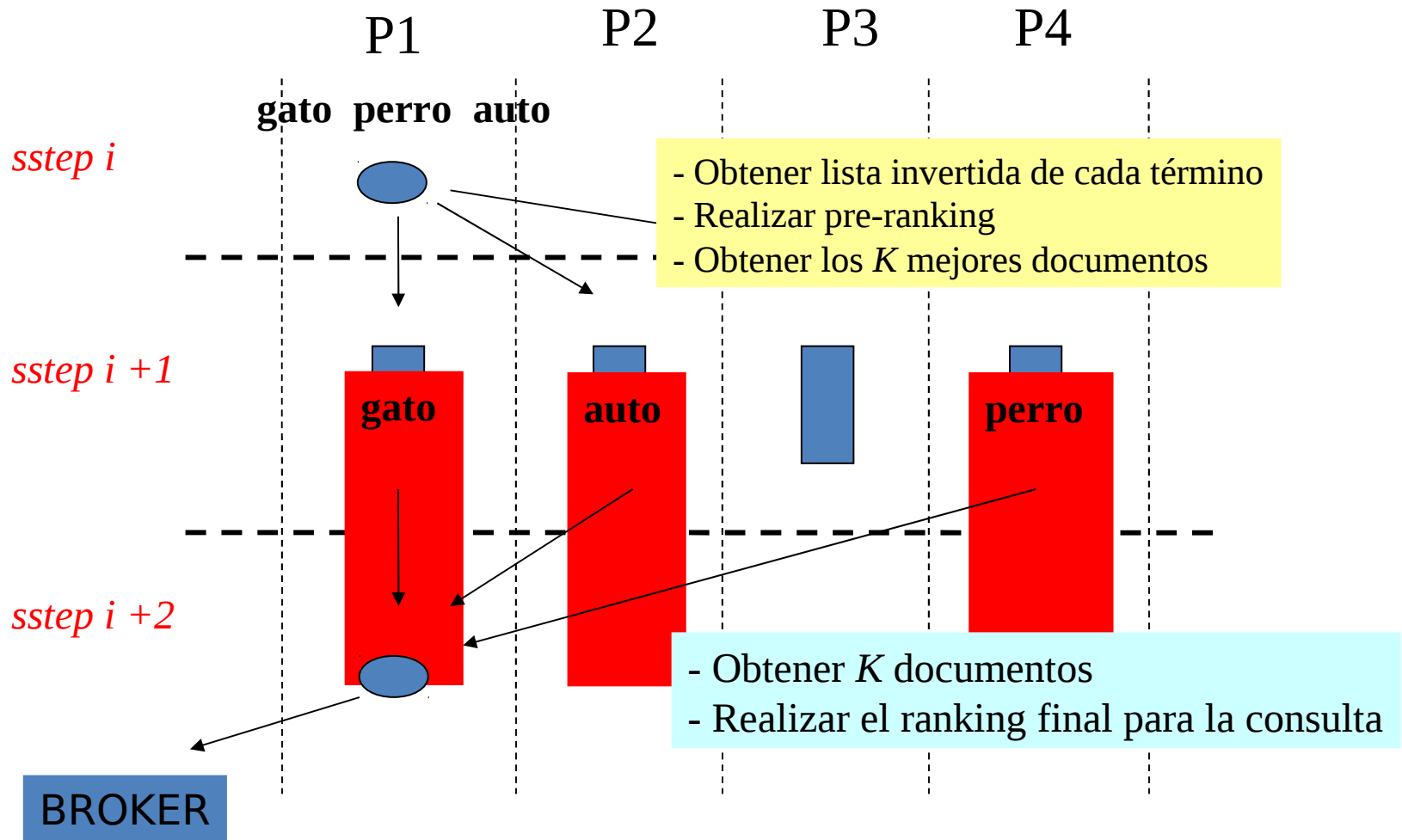
Operación: Consultas

- Se recuperan las listas invertidas de cada término para realizar el ranking parcial.
- Los mejores documentos se envían al procesador *ranker* para hacer el ranking final

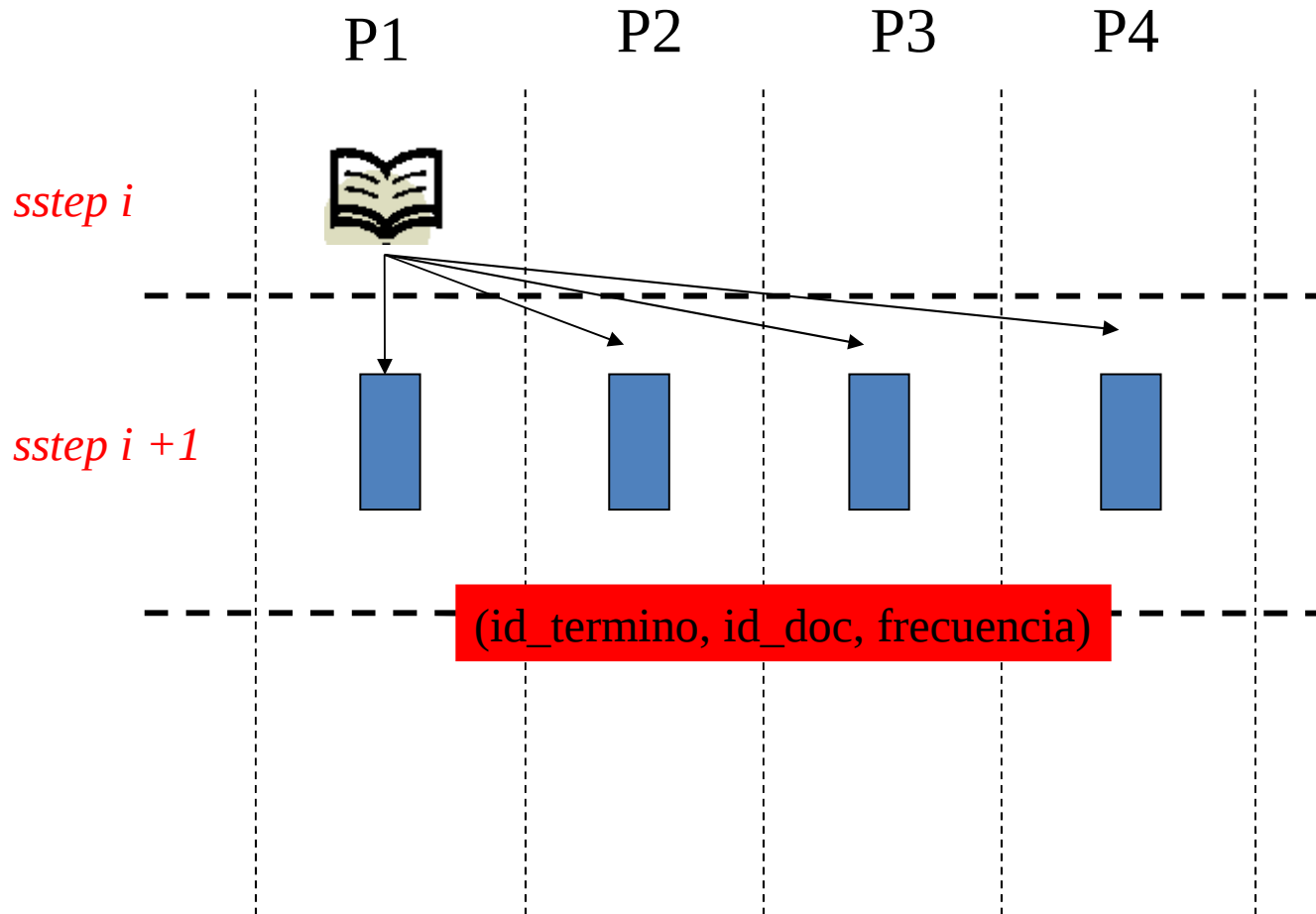
Operación: inserción de nuevos documentos

- Actualizar las listas invertidas correspondientes a los términos del documento.

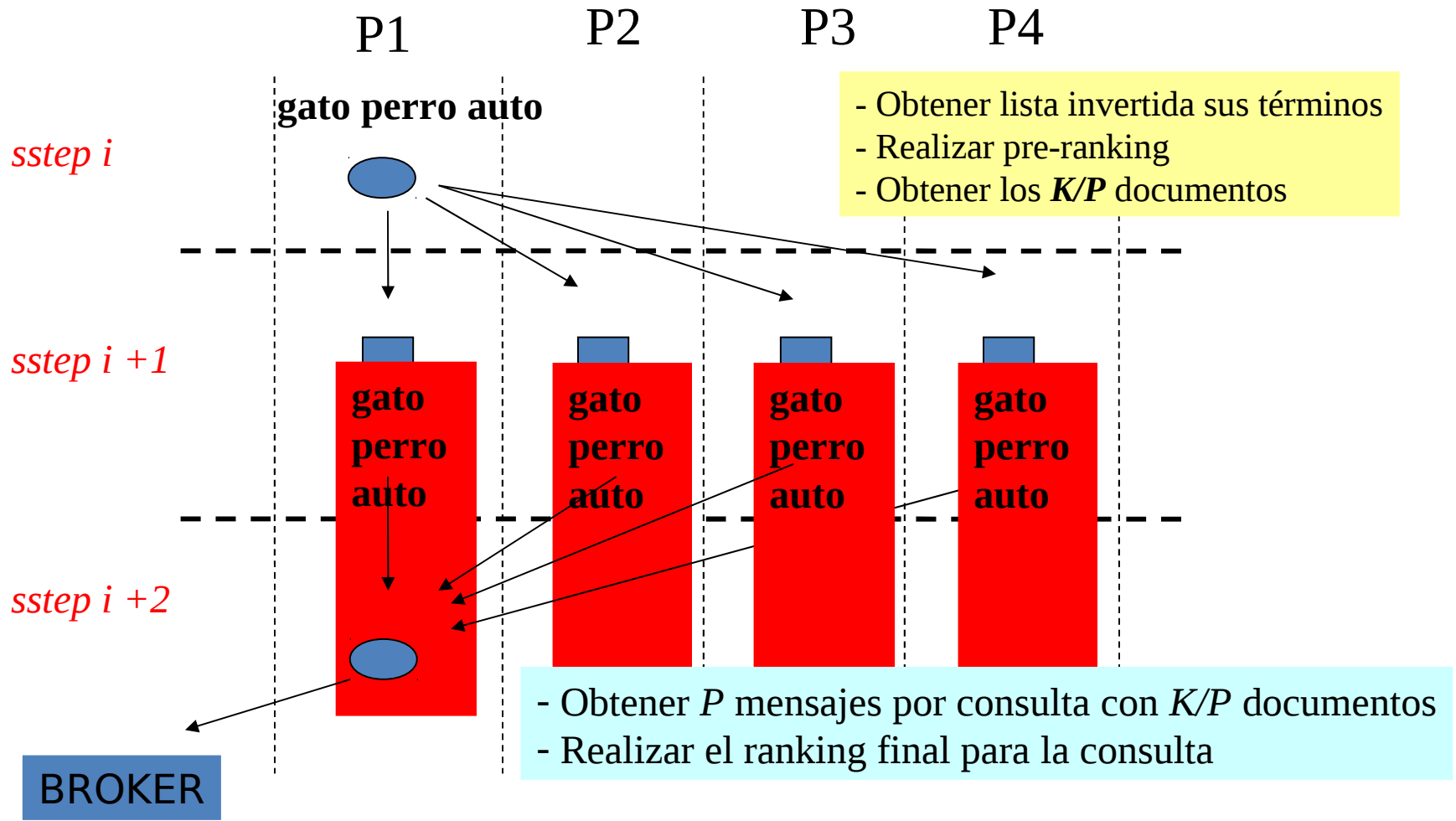
Índice Particionado por Términos: lectura



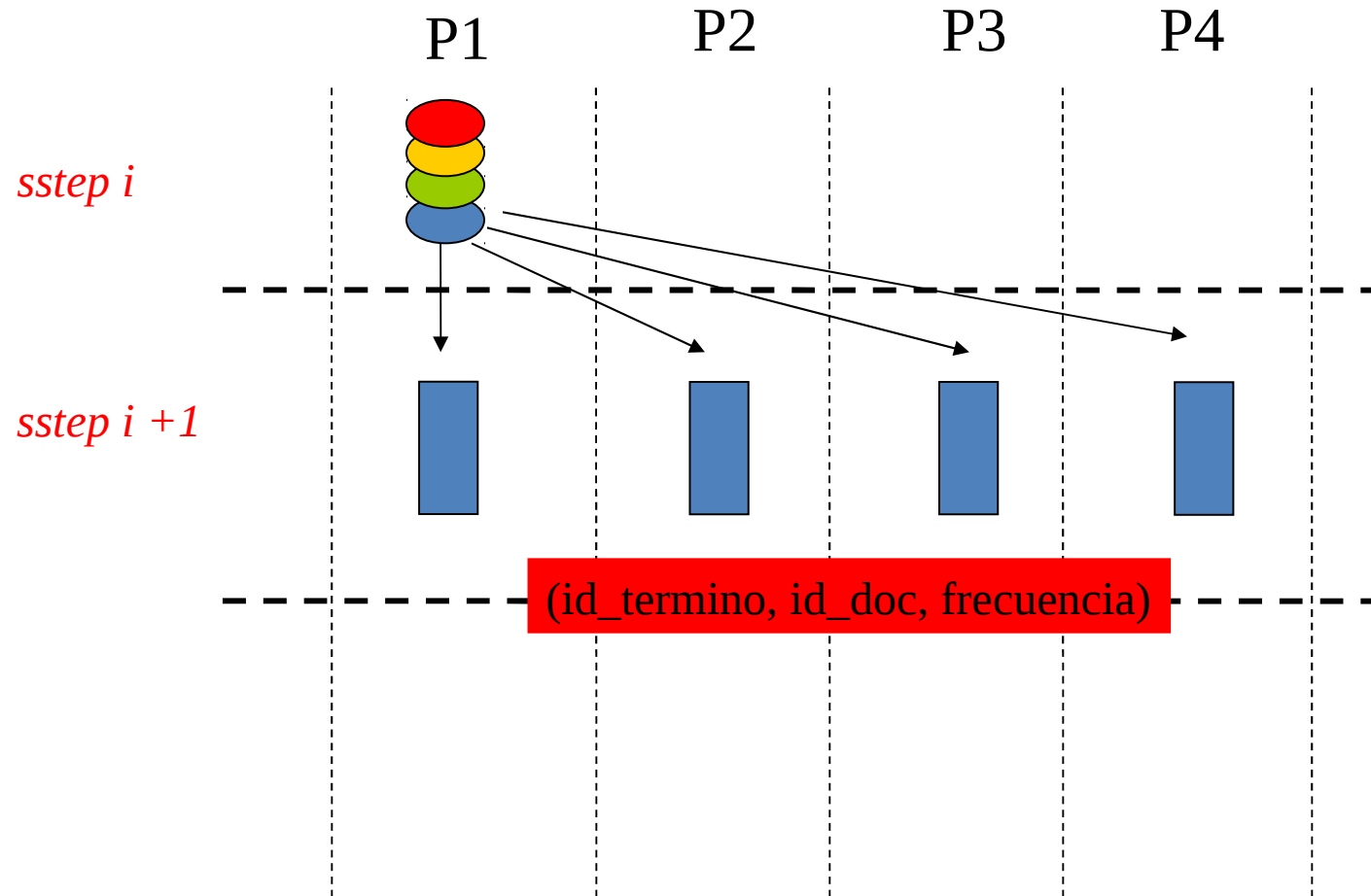
Índice Particionado por Términos: escritura



Índice Particionado por Documentos: lectura



Índice Particionado por Documentos: escritura



- Mejora el throughput del sistema.
- Incrementa la cantidad de operaciones por procesador.
 - Mejora balance de carga
 - Amortiza los costos de comunicación y sincronización
 - **Aumenta el tiempo de respuesta individual de las operaciones**

SuperSteps Exclusivos

- El Broker controla los accesos a los procesadores.
- Espera que las consultas actuales sean *todas* resueltas y luego comienzan las operaciones de actualización/inserción.
- Útil para una máquina de búsqueda que opera en un modo asincrónico.
- También, puede ser implementada sobre el modelo BSP.

Protocolo de Locks Distribuidos

- Cada procesador mantiene un conjunto de *locks* de lectura y escritura.
- Cuando se requiere *insertar un nuevo documento*:
 - *Solicitan un locks de escritura por cada término.*
 - *Con todos los locks asignados, se actualizan las listas invertidas de los respectivos términos.*
- Para las *consultas*:
 - *Solicita locks de lectura por cada términos de la consulta.*
 - *Estos son asignados si el término en particular no tiene un locks de escritura previo.*

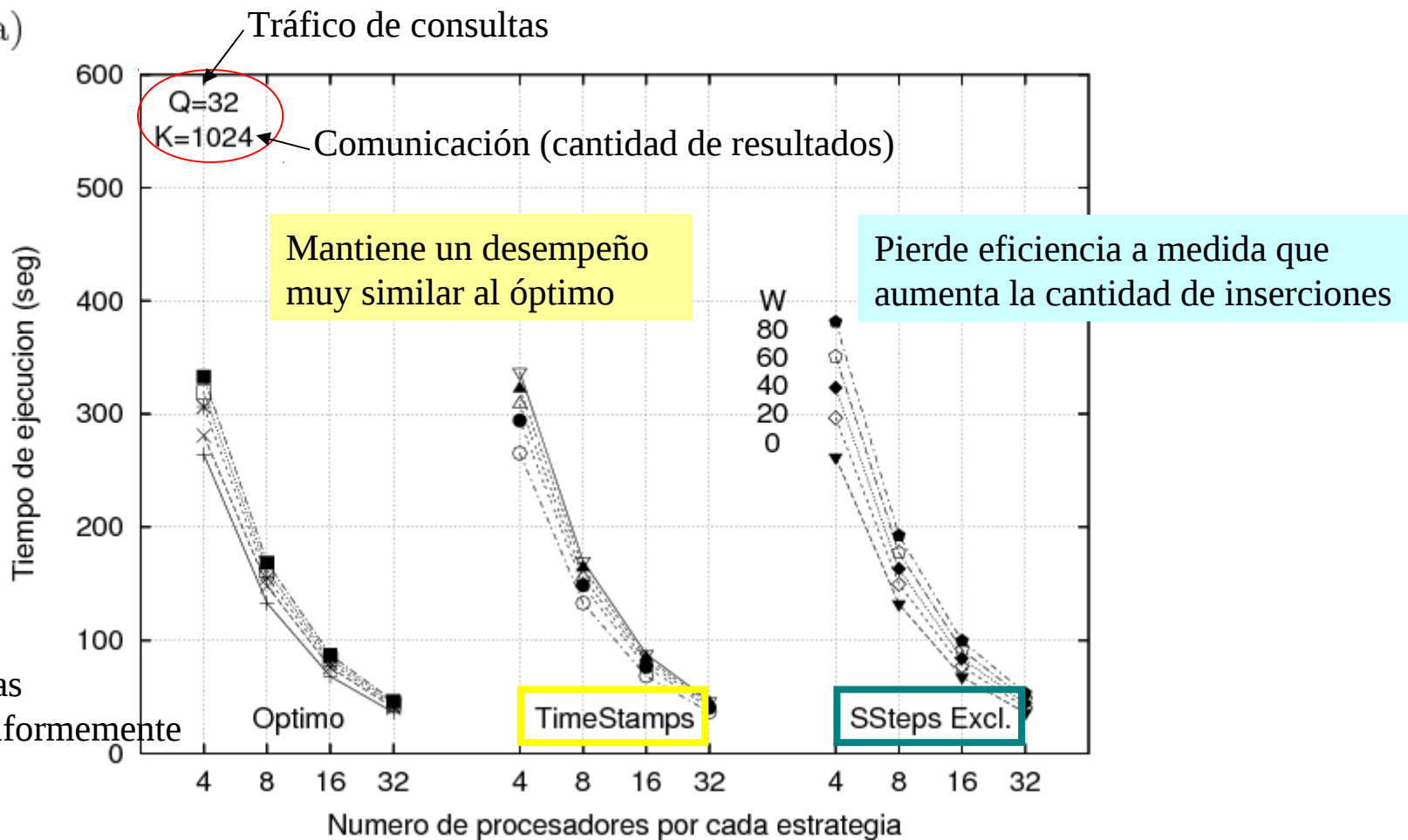
Una vez que finaliza la operación, se hace un *unlock* a los procesadores que tienen dichos términos

Protocolo optimista

- Se modifican las listas invertidas esperando que no van a existir conflictos con operaciones de lectura.
- Inserciones/actualizaciones y consultas son procesadas sin importar los posibles conflictos.
- Representa el mejor caso.
- Cualquier protocolo de control de concurrencia debería realizar más operaciones, por lo tanto debería demandar un tiempo de ejecución mayor que el óptimo.

Tiempos de ejecución para un índice particionado por documentos

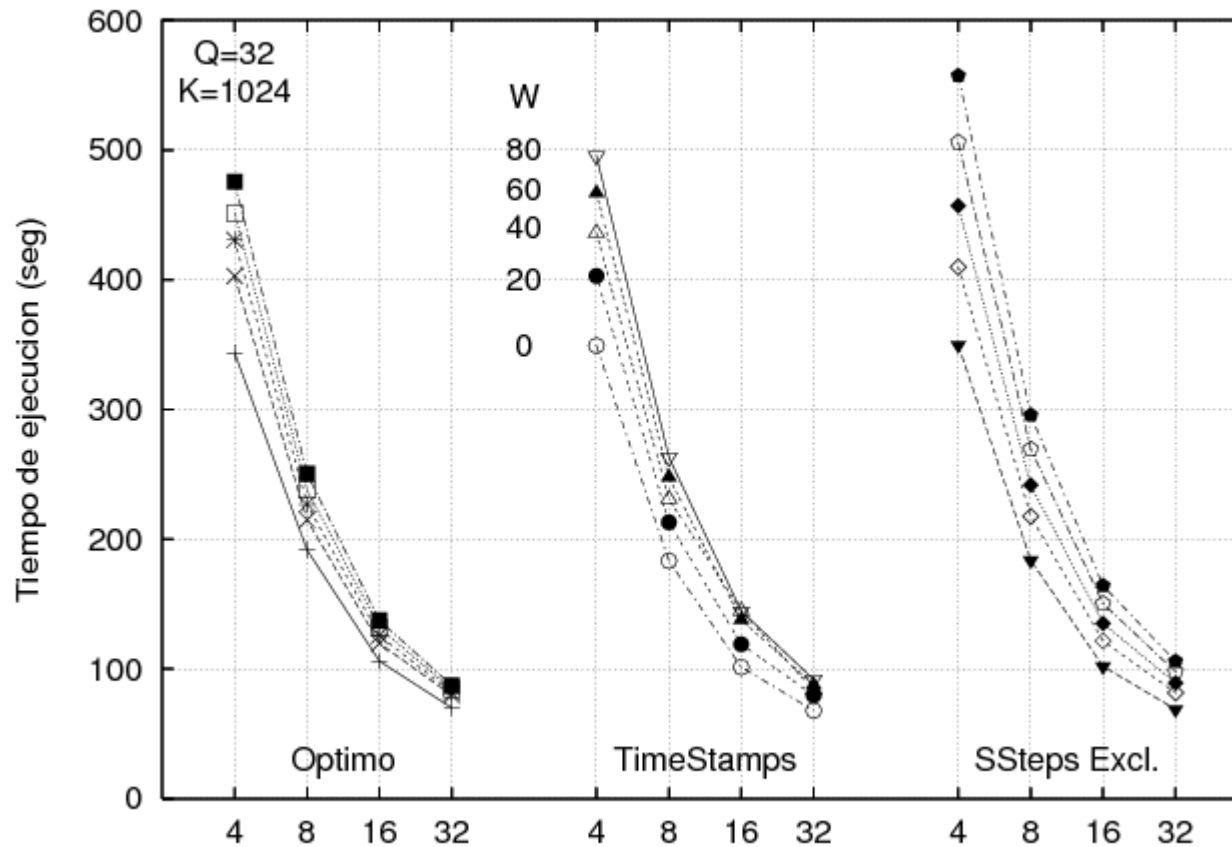
(a)



80.000 consultas distribuidas uniformemente

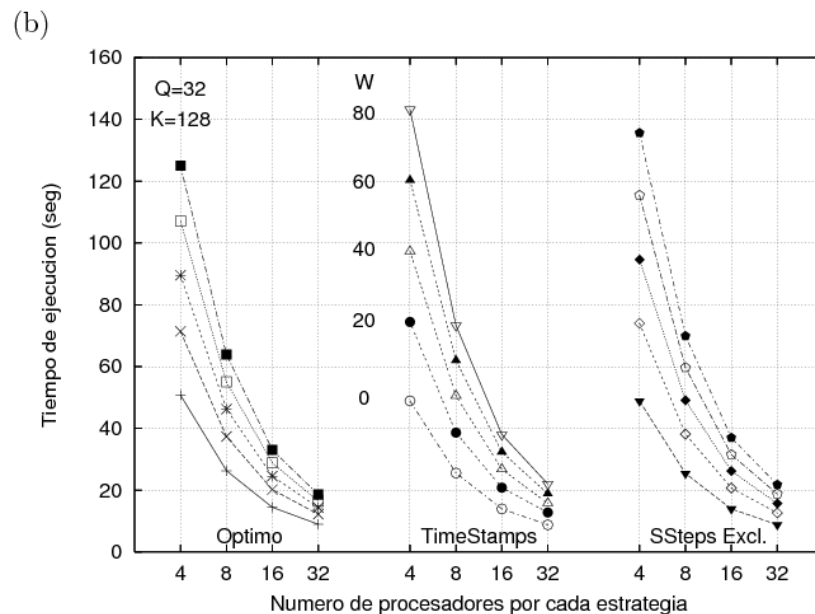
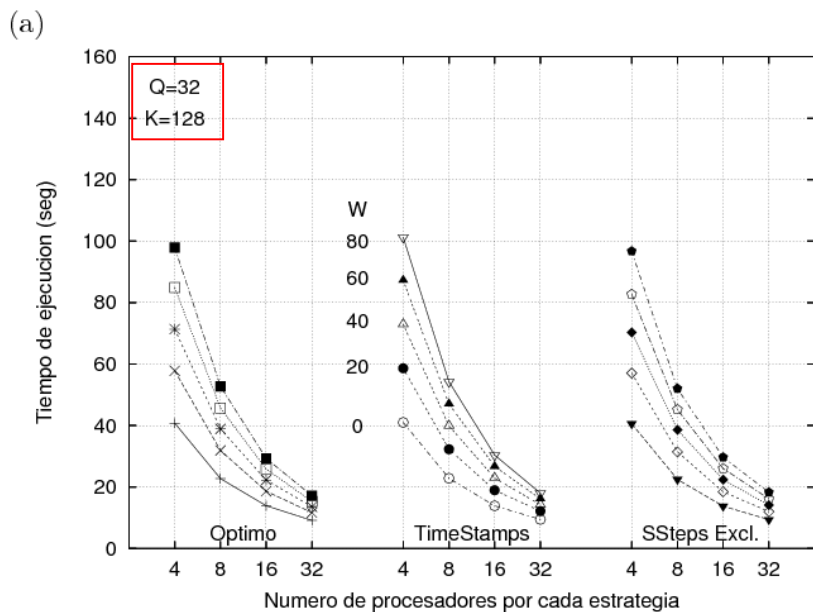
Tiempos de ejecución para un índice particionado por términos

(b)



Los términos y listas invertidas están distribuidas de manera balanceada en los procesadores, pero los términos de las consultas y los documentos a insertar son sesgados, lo cual produce un desbalance.

Tiempos de ejecución para un índice particionado por documentos (a) y por términos (b)



SE presenta mejor balance en el proceso de consultas puesto que las escrituras son atendidas de manera exclusiva en los procesadores.

La mezcla de consultas e inserción de documentos tiende a producir más desbalance cuando el ranking se hace sobre unos pocos documentos.