

8. Complements

1. SGBD
2. Escalabilitat
3. NO SQL
4. Altres "models"

8.2 Escalabilitat

- a. Conceptes i tipus
- b. Sharding
- c. Map-Reduce
- d. Cache
- e. Factors per al disseny de la BD

8.2 Escalabilitat

a. Conceptes i tipus



The ability of a business or system to grow larger

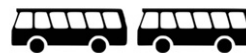
Des del punt de vista del maquinari, pot ser:

-- Vertical

-- Horizontal



Scale Up- Vertical Scaling



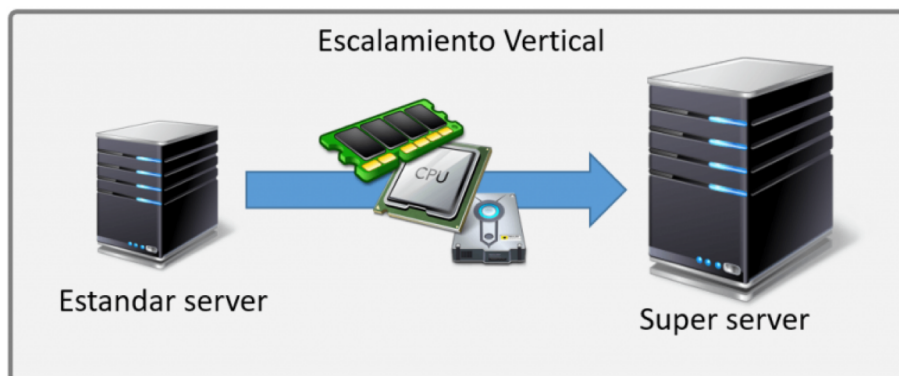
Scale Out- Horizontal Scaling

3

8.2 Escalabilitat

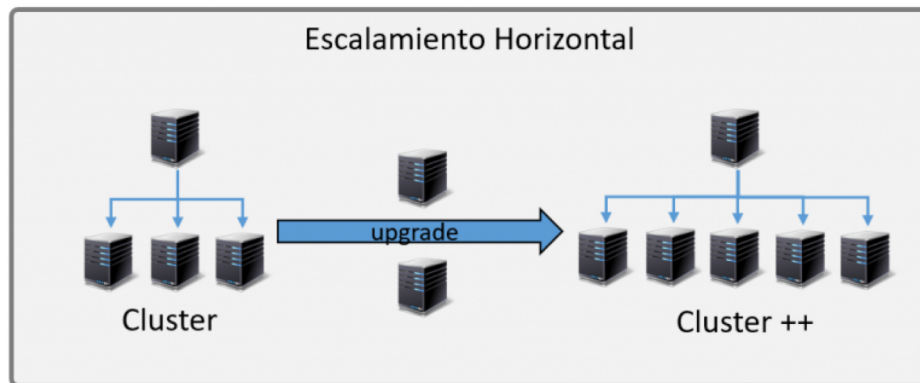
a. Conceptes i tipus

Vertical



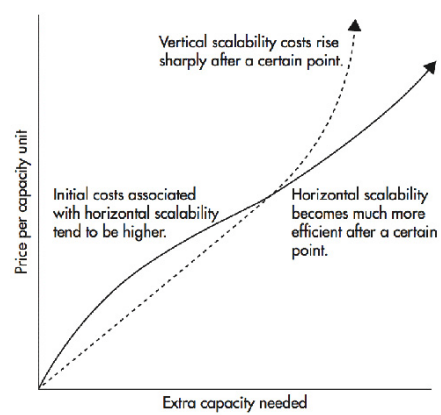
4

Horitzontal



5

En quant a cost:



6

8.2 Escalabilitat a. Conceptes i tipus

Comparació

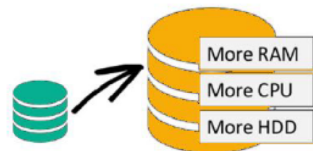
<i>HORIZONTAL</i>	<i>VERTICAL</i>
1. Load balancing required	1. Load balancing unnecessary
2. Resilient to system failure	2. Single point of failure
3. Utilizes Network Calls	3. Inter process communication
4. Data Inconsistency	4. Data consistent
5. Scales well	5. Hardware limit

7

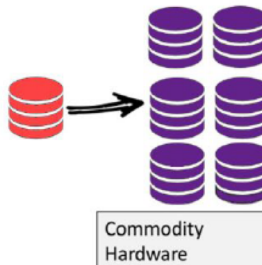
8.2 Escalabilitat a. Conceptes i tipus

Des del punt de vista de la BD

Scale-Up (*vertical scaling*):



Scale-Out (*horizontal scaling*):



8

8.2 Escalabilitat

- a. Conceptes i tipus
- b. Sharding
- c. Map-Reduce
- d. Cache
- e. Factors per al disseny de la BD

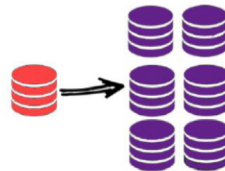
9

8.2 Escalabilitat b. Sharding

Sharding es la tècnica d'optimitzar una BD mitjançant la separació per files o columnes de taules grans a fi de tenir múltiples taules de menor grandària.

Aquestes noves taules s'anomenen particions o *shards*.

És l'*Scale-Out* d'abans:



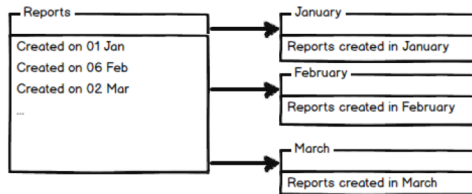
10

8.2 Escalabilitat b. Sharding

Particionament de taules:

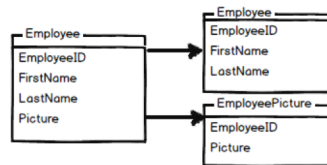
Horitzontal

- Es divideix per files.
- Totes les particions tenen les mateixes columnes.
- La divisió és per conceptes i es gestiona per sw



Vertical

Es divideix per columnes



11

8.2 Escalabilitat b. Sharding

Com a particions, per crear el shards es pot fer:

- **Horitzontal**: Efectiu quan el resultat habitual de consultes està agrupat per files (per rangs)
- **Vertical**: Quan el resultat habitual es per columnes, o hi ha columnes que es consulten molt poc, o columnes LOB

Original Table

CUSTOMER ID	FIRST NAME	LAST NAME	CITY
1	Alice	Anderson	Austin
2	Bob	Best	Boston
3	Carrie	Conway	Chicago
4	David	Doe	Denver

Vertical Shards

VS1			VS2	
CUSTOMER ID	FIRST NAME	LAST NAME	CUSTOMER ID	CITY
1	Alice	Anderson	1	Austin
2	Bob	Best	2	Boston
3	Carrie	Conway	3	Chicago
4	David	Doe	4	Denver

Horizontal Shards

HS1			
CUSTOMER ID	FIRST NAME	LAST NAME	CITY
1	Alice	Anderson	Austin
2	Bob	Best	Boston
HS2			
CUSTOMER ID	FIRST NAME	LAST NAME	CITY
3	Carrie	Conway	Chicago
4	David	Doe	Denver

12

8.2 Escalabilitat b. Sharding

Diferencia entre particions o shards d'una taula:

Conceptualment es el mateix, però els shards son particions que físicament estan ubicades/repartides entre diferents màquines

****Shard lògic vs shard físic, no es el mateix!!**

Pros

- Sistemes escalables
- Millora el processament paral·lel
- Cerques sobre parts petites
- Disminueix la grandària dels índex

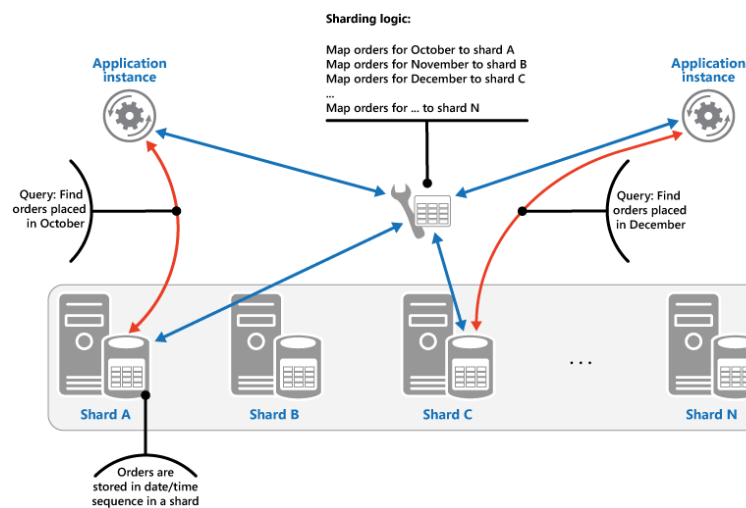
Contres

- Programació complexa
- Problemes de comunicació

Tolerància a fallades

13

8.2 Escalabilitat b. Sharding

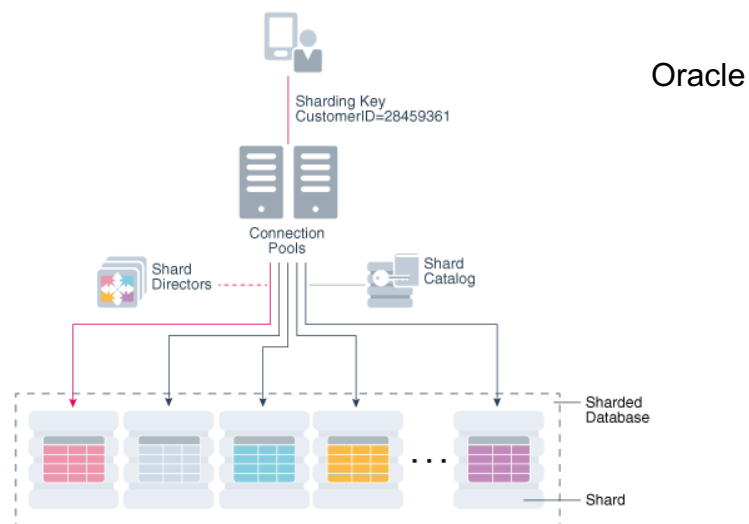


MS

14

8.2 Escalabilitat

b. Sharding



15

8.2 Escalabilitat

- a. Conceptes i tipus
- b. Sharding
- c. Map-Reduce
- d. Cache
- e. Factors per al disseny de la BD

16

Map-reduce is a data processing paradigm for condensing large volumes of data into useful aggregated results.



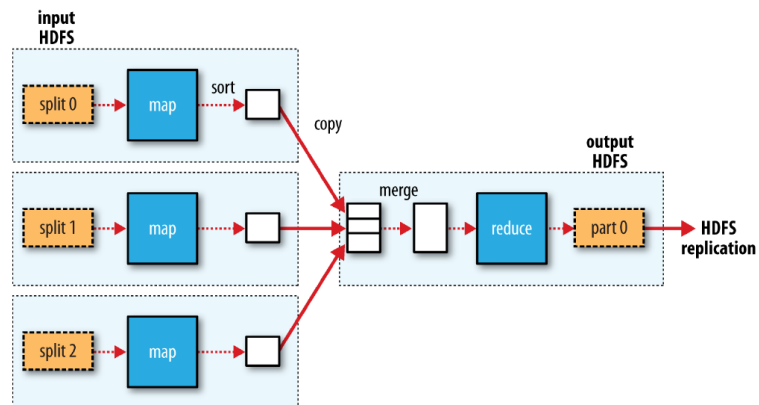
MapReduce és un model de programació i d'implementació per a processar i generar jocs de dades grans, amb un [algorisme](#) paral·lel i distribuït, en un [clúster](#).



Es compon de dos procediments:

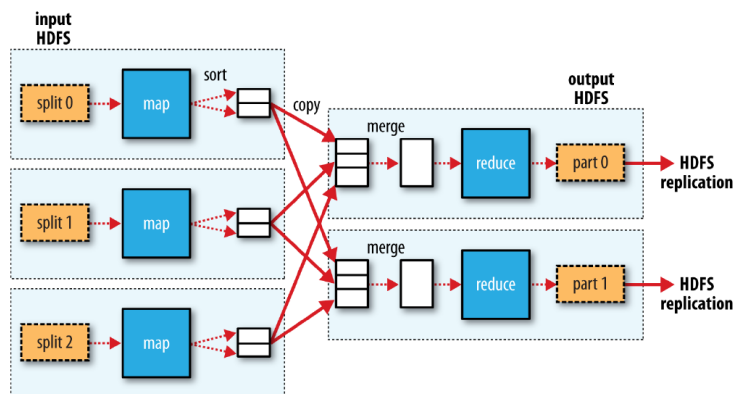
- **Map()** que efectua el filtrat i ordenat (per ex. ordenar estudiants pel primer cognom en cues, amb una cua per cognom)
- **Reduce()**, que fa l'operació d'agregació (per ex. comptar el nombre d'estudiants a cada cua, obtenint-ne la freqüència dels cognoms).

Esquema Map-Reduce



19

Successius Map-Reduce



20

8.2 Escalabilitat

- a. Conceptes i tipus
- b. Sharding
- c. Map-Reduce
- d. Cache
- e. Factors per al disseny de la BD

21

8.2 Escalabilitat d. Cache



La **memòria cau**, o memòria *cache* és una memòria d'alta velocitat instal·lada en el mateix processador i en la qual s'emmagatzemen les dades que el processador necessita utilitzar immediatament i aquelles que són llegides amb més freqüència.

22

8.2 Escalabilitat d. Cache

Caching is a buffering technique that stores frequently-queried data in a temporary memory. It makes data easier to be accessed and reduces workloads for databases.

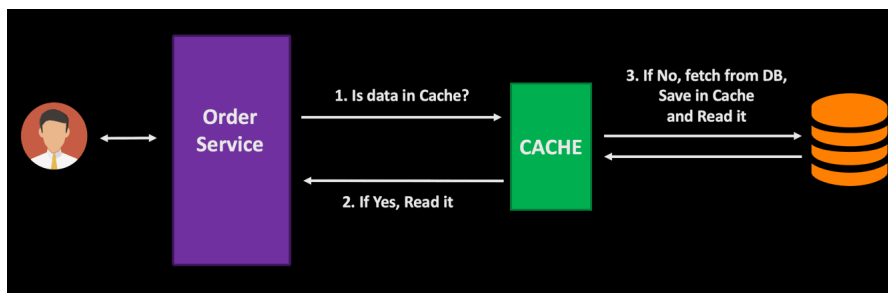
For example, you need to retrieve a user's profile from the database and you need to go from a server to server. After the first time, the user profile is stored next (or much nearer) to you. Therefore, it greatly reduces the time to read the profile when you need it again.



23

8.2 Escalabilitat d. Cache

En resum:



24

Beneficis:

- **Performance** — Performance is improved by making data easier to be accessed through the cache and reduces workloads for database.
- **Scalability** — Workload of backend query is distributed to the cache system which is lower costs and allow more flexibility in processing of data.
- **Availability** — If backend database server is unavailable, cache can still provide continuous service to the application, making the system more resilient to failures.

Estratègia de *caching*:

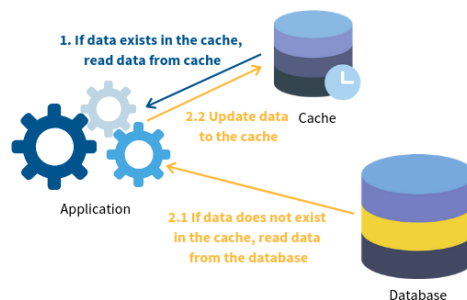
És la relació que hi ha entre la font de dades i el sistema de cache i com s'accedeix a aquestes dades.

Hi ha diversos tipus d'estratègies, tots influeixen en el disseny del sistema i en el seu rendiment global. Els principals son:

- *Cache Aside*
- *Read Through*
- *Write Through*
- *Write Back*
- *Write Around*

Cache Aside:

La cache està
apart de la BD

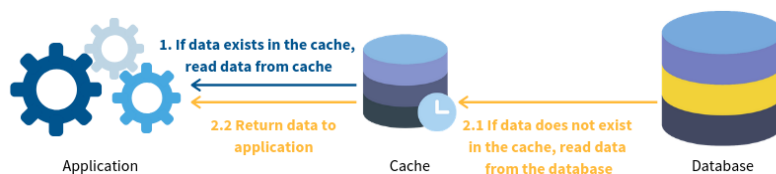


27

Read Through:

La cache està entre l'aplicatiu i la BD.

L'aplicatiu just treballa amb la cache que es la responsable de llegir les dades de la BD i actualitzar-se

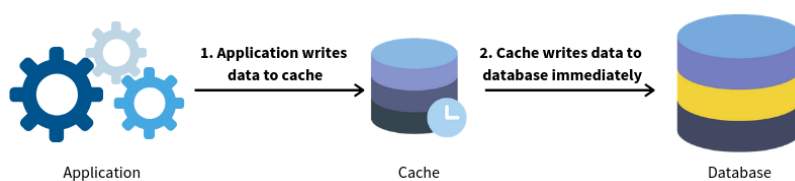


28

Write Through:

La cache està entre l'aplicatiu i la BD.

Tota escriptura que fa l'aplicatiu ha d'anar immediatament a la BD passant per la cache

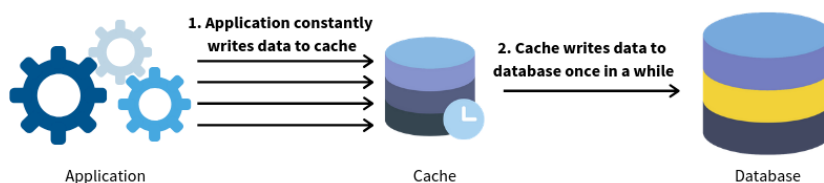


29

Write Back (Write Behind):

La cache està entre l'aplicatiu i la BD.

Tota escriptura es fa a la cache que la passarà a la BD cada cert temps (produeix *delay*)

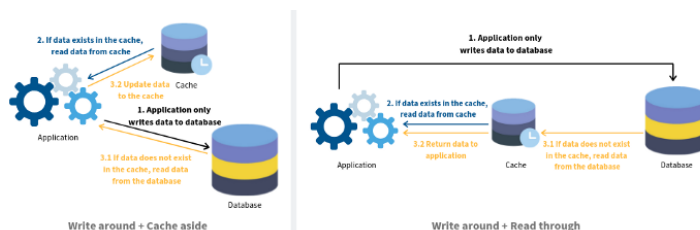


30

Write Around:

Combina amb *Cache Aside* o amb *Read Through*.

Les escriptures es fan directament a la BD, únicament els resultats de lectures van a la cache



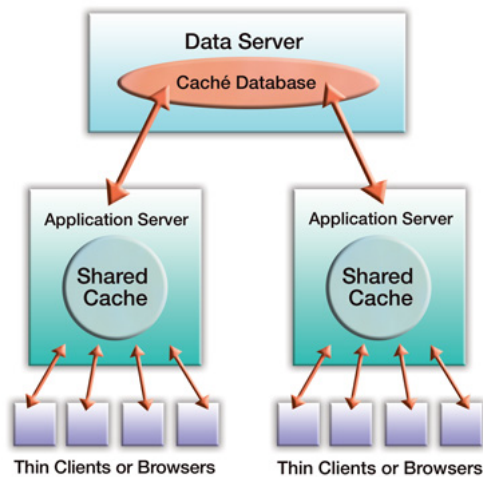
31

Estratègia	Pros	Contres
C Aside	<ul style="list-style-type: none"> Resistent a fallades quan cau la BD Implementació senzilla El model de dades de la cache pot ser diferent del de la BD En general millor per molta càrrega de lectures (<i>read-heavy</i>) 	<ul style="list-style-type: none"> Es pot produir inconsistència entre la cache i la BD Latència inicial considerable
R Through	<ul style="list-style-type: none"> Programació de l'aplicatiu mes simple Millor per <i>read-heavy</i> on les lectures es repeteixen molt 	<ul style="list-style-type: none"> El <i>plugin</i> de la cache es mes complex Latència inicial considerable
W Through	<ul style="list-style-type: none"> Garanteix la consistència si s'empra juntament amb <i>Read Through</i> 	<ul style="list-style-type: none"> Incrementa la latència d'escriptura
W Back	<ul style="list-style-type: none"> Bona per sistemes <i>write-heavy</i> Relativament tolerant a fallades de BD Redueix escriptures a la BD 	<ul style="list-style-type: none"> Si falla la cache es poden perdre les dades de forma definitiva

32

8.2 Escalabilidad d. Cache

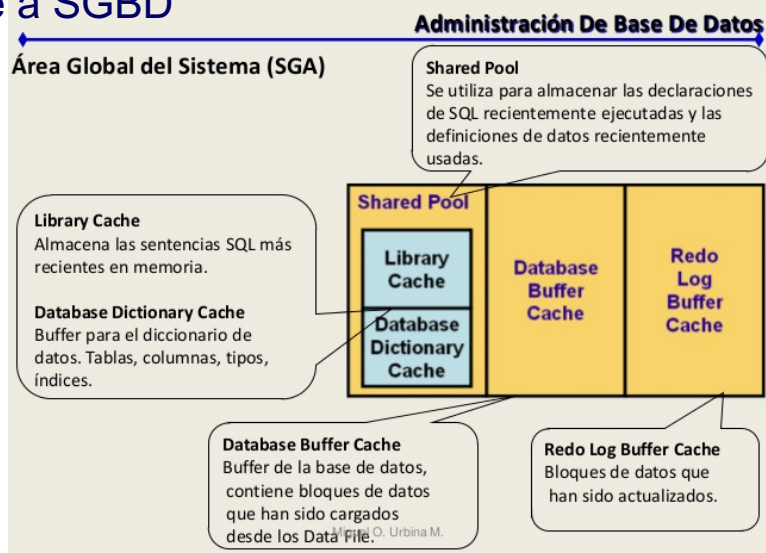
Tot servidor te caches!!



33

8.2 Escalabilitat d. Cache

Cache a SGBD



34

8.2 Escalabilitat

- a. Conceptes i tipus
- b. Sharding
- c. Map-Reduce
- d. Cache
- e. Factors per al disseny de la BD

35

8.2 Escalabilitat e. Factors per al disseny de la BD

Disseny de la Base de Dades:
Evitar colls de botella

- Requisits
- Taules massa grans
- Estructures en estrella.
- Factors “geogràfics”

36

Estructures en estrella

