Apunts1BigDataIntroduccio.pdf

Aridevi

Desenvolupament d'Aplicacions de Dades Massives

3º Grado en Ingeniería de Datos

Escuela de Ingeniería
Universidad Autónoma de Barcelona

BIG DATA

DESENVOLUPAMENT D'APLICACIONS DE DADES MASSIVES

1. INTRODUCCIÓ

Un sistema de dades és un conjunt de programes que permeten l'emmagatzemament, modificació i extracció de la informació en una base de dades.

Per tal de fer-lo possible, son necessàries:

- Noves eines d'emmagatzematge i tractament de dades.
- Noves aplicacions amb una àmplia gamma d'exigències de tractament de dades.
- Cap eina pot satisfer les necessitats de processament i emmagatzematge de dades.
- El processament es divideix en tasques per a una única eina.
- Les diferents eines estan vinculades amb el codi de l'aplicació.

Existeixen aplicacions de dades intensives son aplicacions que estan vinculades a E/S o amb la necessitat de processar grans volums de dades.

- Aquestes, han de controlar:
  - ✓ No limitat per la potència de la CPU.
  - ✓ Quantitat rellevant de dades.
  - ✓ Complexitat de les dades.
  - ✓ Velocitat de canvi de dades.

- Aquestes, necessiten:
  - ✓ **Base de dades:** emmagatzema dades per al seu posterior tractament.
  - ✓ **Cache:** recordeu el resultat d'una operació costosa.
  - ✓ **Índex de cerca:** cerca dades amb paraules clau.
  - ✓ **Stream processing:** envia missatges per ser gestionats de manera asíncrona.
  - ✓ **Batch processing:** processa una gran quantitat de dades acumulades.

- Application as a service:
  - ✓ Utilitzar com a components petites eines d'ús general.
  - ✓ Construir un sistema de dades amb finalitats especials.
  - ✓ Ara ets un dissenyador de sistemes de dades!
  - ✓ El vostre sistema estarà disponible per als programadors mitjançant API.

- Coses a fer amb un sistema de dades:
  - ✓ Assegurar-se que les dades segueixen sent correctes i completes.
  - ✓ Proporcionar un bon rendiment als clients.
  - ✓ Ser capaç d'escalar quan augmenta la càrrega.
  - ✓ Proporcionar una bona API per al servei.

- Tres conceptes importants: #Pregunta examen
  - ✓ **Fiabilitat:** El sistema ha de funcionar correctament quan es produeixen errors.
    - Faluts: El component comença a funcionar sota les especificacions.
    - Failures: El sistema deixa de prestar el servei requerit.
    - Mecanismes de tolerància d'errors: Prevenir falles per produir avaries.
  - ✓ **Escalabilitat:** Com afrontar el creixement (dades, trànsit, complexitat).
    - Capacitat del sistema per fer front a l'augment de càrrega.
    - La càrrega d'un sistema depèn de:
      1. L'arquitectura del sistema.
      2. Sol·licituds per segon al servidor web.
      3. Relació de sol·licituds de lectura/escriptura a una base de dades.
      4. Nombre simultània d'usuaris en videoconferència.
  - ✓ **Manteniment:** Mantenir i millorar el comportament d'un sistema, tasca realitzable/productiva.

Ariadna De Vicente Viladesau

L'activitat d'un usuari a Twitter: #Pregunta examen

Anem a plantejar una situació i a analitzar el comportament del sistema. Quan parlem de **tweet**, ens referim a la publicació que fa un usuari pel seus seguidors, mentre que quan parlem de **Home Timeline**, fa referència als tweets publicats que pot visualitzar l'usuari de la gent que segueix.

Què significa el concepte fan-out quan estem analitzant de l'activitat d'un usuari a Twitter?

Fan-out: Un usuari segueix a N altres usuaris i cada usuari és seguit per M altres usuaris. Cada missatge publicat té un potencial de N*M número de missatges a gestionar.

Per emmagatzemar els tweets, tindríem dues opcions:

1. Inserir cada tweet nou en una col·lecció global de tweets.

   Quan l'usuari sol·liciti mirar el home timeline. Realitzaríem els passos següents: Buscar tots els usuaris que segueix, buscar tots els tweets de tots aquests usuaris i ordenar tots els tweets per temps.

   Quan tenim N usuaris i M seguidors per a cada usuari, quin és el principal problema que hem de gestionar? Principal problema: Creixement quadràtic de missatges amb un número limitat de recursos (capacitat d'emmagatzematge, temps de processament). Hem d'evitar haver de processar N*M missatges per construir la visió dels missatges dels usuaris (timeline quèries).

2. Mantenir al cache de cada usuari timeline.

   Quan l'usuari publiqui un tweet. Realitzaríem els passos següents: Buscar totes les persones que segueixen aquest usuari i inserir el nou tweet a cadascuna de les memòries cau de la cronologia d'inici. D'aquesta manera, el Timeline de lectura ara és més senzill, ja que, **ja està calculat**.

   Quines solucions es poden aplicar?

   Solució principal: Passa per aprofitar el fet de que la ràtio de missatges publicats és 100 vegades menor que la ràtio de lectura del home timeline. Per tan podem construir una cache amb el timeline de cada usuari quan van arribar els missatges nous. Quan l'usuari vol llegir el seu timeline, ja està preparat.

S'haurà de tenir en compte que per aquells usuaris que siguin famosos i tinguin molts seguidors, un simple tweet pot arribar a generar 100 milions d'escriptures al home timelines (processat en 5 segons). "Celebrities latency challenge".

La majoria dels usuaris tenen la seva timeline preprocessada amb memòria cau, els tweets de famosos s'obtenen i es combinen quan es llegeix el home timeline i quan se sol·licita, els tweets de celebritats es fusionen amb el home timeline de l'usuari seguint l'enfocament 1.

Rendiment del sistema

Com es veu afectat el rendiment en augmentar el paràmetre de càrrega al mateixos recursos (CPU, memòria, ample de banda de xarxa).

En augmentar un paràmetre de càrrega, quants més recursos tenim necessiteu mantenir el rendiment sense canvis?

- Com descriure el rendiment?
  - ✓ **Rendiment:** Nombre de registres a processar per segon. Temps per processar un treball en un conjunt de dades.
  - ✓ **Temps de resposta:** Temps entre un client que envia una sol·licitud i rep una resposta.
  - ✓ **Latència:** Durada que una sol·licitud està esperant per ser gestionada.
- Afrontar la càrrega:
  - ✓ **Ampliació/escala vertical:** Moveu-vos a un servidor més potent. És car i té limitacions de la tecnologia.
  - ✓ **Scale-out/escalada horitzontal:** Distribueix la càrrega entre múltiples més petits màquines. Difícil de mantenir una vista unificada/dades amb estat.
  - ✓ **Sistemes elàstics:** Afegeixen recursos quan augmenta la càrrega. Bons per a esdeveniments impredictibles.

Ariadna De Vicente Viladesau