11.2 Key-Value Stores

- slide 1

Penyimpanan nilai kunci ini dirancang sebagai koleksi pasangan nilai dan kunci, di mana setiap kunci memiliki sifat unik. Model penyimpanan ini tidak terikat pada schema tertentu, memberikan kebebasan untuk menyimpan nilai apa pun sesuai kebutuhan. Fleksibilitas dan skalabilitas yang tinggi menjadi keunggulan utama.

Dalam penyimpanan ini, kita dapat dengan mudah menambahkan, mengambil, atau menghapus data menggunakan antarmuka yang sederhana.

- slide 2

Salah satu penyimpanan nilai value yang yang popular di sini adalah amazon dynmoDB.

DynamoDB, digunakan oleh layanan inti Amazon yang memerlukan ketersediaan tinggi dan akses data berbasis kunci, merupakan solusi yang menawarkan skalabilitas dan ketersediaan yang optimal. Layanan-layanan seperti keranjang belanja, daftar penjual, preferensi pelanggan, dan katalog produk mengandalkan DynamoDB untuk memenuhi kebutuhan ini.

-slide 3

DynamoDB menyimpan data dalam bentuk tabel basis data, di mana setiap tabel terdiri dari kumpulan item. Masing-masing item merupakan daftar pasangan atribut-nilai, dengan nilai atribut dapat berupa tipe skalar, set, atau JSON. Konsep ini menggambarkan item sebagai baris dalam tabel relasional, namun dengan fleksibilitas tambahan karena atribut dapat bersifat heterogen (atribut-atribut dalam sebuah item tidak harus memiliki tipe data yang sama atau mematuhi skema relasional yang kaku).

Meskipun DynamoDB mencapai ketersediaan dan skalabilitas tinggi, terdapat pengorbanan konsistensi dalam beberapa skenario kegagalan. Hal ini diatasi dengan menggunakan teknik sintesis P2P dalam sebuah cluster.

-slide 4

Hash Konsisten:

* Hash konsisten adalah teknik pengaturan data di dalam sistem terdistribusi. Ide dasarnya adalah menggunakan fungsi hash untuk memetakan data ke dalam sejumlah node atau server.
* Fungsi hash adalah algoritma matematis yang mengonversi data (seperti nama atau angka) menjadi nilai hash, yang kemudian digunakan untuk menentukan di mana data tersebut akan disimpan.

Node Bertanggung Jawab:

* Setiap node atau server dalam sistem terdistribusi memiliki tanggung jawab untuk menyimpan sebagian data.
* Dalam contoh ini, kita punya Node A dan Node B.

Interval Nilai Hash:

* Data di-hash, dan hasil hashnya adalah angka atau nilai tertentu.
* Interval nilai hash (A, B] berarti bahwa data yang di-hash dan nilainya berada di antara A (tidak termasuk A) dan B (termasuk B) akan ditangani oleh Node B.

Contoh Item (c, v):

* Misalnya, kita memiliki item (c, v), di mana c adalah kunci dan v adalah nilainya.
* Jika hasil hash dari c berada di antara A dan B (termasuk B), maka item ini akan ditugaskan atau disimpan di Node B.

-slide 5