

Samenvatting Databanken III - NoSQL

TIN 3 - HoGent

Lorenz Verschingel

31 oktober 2015

1 Distributed Systems

Een gedistribueerd systeem bestaat uit verschillende computers en software componenten. Deze communiceren met elkaar via het netwerk en delen de resources. Het systeem kan mainframes, workstations... bevatten.

1.1 Voordelen van het gedistribueerd systeem

1. Reliability:

Als één machine crashed wordt de rest hierdoor niet beïnvloed.

2. Scalability:

Men kan makkelijk machines toevoegen als daar nood aan is.

3. Performance:

De verzameling aan processoren van de verschillende machines kan een hoger performantie bieden dan een centralized computer.

4. Open system:

Iedere service is toegankelijk voor iedere client.

5. Sharing of Resources

6. Flexibility

7. Speed

1.2 Nadelen van het gedistribueerd systeem

1. Software:

Er is minder software support → Dit is het grootste probleem

2. Security:

Het delen van data tussen de machines verhoogt het risico op het vlak van security.

3. Troubleshooting

4. Networking

1.3 Scalability

1.3.1 Verticaal schalen

Resources toevoegen aan een bestaande logische eenheid om de capaciteit te verhogen.

1.3.2 Horizontaal schalen

Meer machines (nodes) toevoegen aan het systeem om de capaciteit te verhogen.

2 NoSQL

2.1 Wat?

NoSQL = Not Only SQL

- Geen relationeel model
- Draait goed op clusters
- Meestal open-source
- Schema-less

2.2 Waarom?

NoSQL laat developers toe om met objecten onmiddellijk in de database te stoppen, zonder ze eerste te converteren naar het relationeel model.

NoSQL databanken zijn goedkoper dan relationele databanken. De laatste zijn ook niet bedoeld om foto's en dergelijke in op te slaan terwijl dit nu de snelst groeiende data is.

→ Gebruik de juiste databank voor de juiste toepassing

2.3 CAP theorem

1. Consistency: Alle nodes in het systeem zien dezelfde data op hetzelfde moment.
2. Availability: Wanneer één van de nodes in het systeem uitvalt beperkt het de andere nodes niet in hun functioneren.
3. Partition Tolerance: Het systeem blijft functioneren bij het willekeurig verliezen van berichten.

Je kan maar 2 tegelijk garanderen:

CA Single site cluster, hierdoor zijn de nodes altijd met elkaar in contact. Als een partitie bijkomt, blokkeert het systeem.

CP Niet alle data is beschikbaar. Wat beschikbaar is, is consistent.

AP Het systeem blijft altijd functioneren, maar de data die geretourneerd wordt is niet altijd consistent.

2.3.1 Voorbeeld: Riak

Riak is een distributed key-value database. In essentie zijn er drie variabelen:

1. r : het aantal nodes dat moet antwoorden alvorens een read request succesvol wordt beschouwd.
2. w : het aantal nodes dat moet antwoorden alvorens een write request succesvol wordt beschouwd.
3. n : de replicatie factor (aantal nodes waarop de data gekopieerd wordt).

Als een Riak cluster bijvoorbeeld vijf nodes heeft dan kan Riak zorgen voor hoge beschikbaarheid of hoge consistentie door r en w aan te passen.

- Hoge beschikbaarheid: r en w zijn beide laag
- Hoge consistentie: r en w zijn beide hoog

2.4 Verschillende soorten NoSQL databases

Er zijn vier soorten NoSQL databases:

1. Key Value: Kan alles ingestopt worden
2. Document: Werkt makkelijk met objecten (JSON, XML)
3. Wide Column: Aggregaties berekenen
4. Graph: Relaties zien