NoSQL ve Distribution Modelleri

Distribution Models

- Veri miktarı arttıkça verinin tek bir sunucuda yönetilmesi zorlaşmaktadır
- Dikey ölçekleme (scale up) de veri artış hızına yeterince cevap veremeyebilir
- Yatay ölçekleme (scale out) yani cluster çözümler daha verimli olabilir
- Verinin cluster ortamda farklı node'lara dağıtılmasında iki ana model vardır
  - Sharding
  - Replication

Distribution Models

- Sharding verinin bölümlere ayrılarak, farklı node'lara dağıtılmasına odaklanır
- Replication ise aynı verinin birden fazla node'a kopyalanması üzerine kuruludur
- Replication'da kendi içinde iki farklı topolojiye sahiptir
  - Master-slave
  - Peer-to-peer

Distribution Models

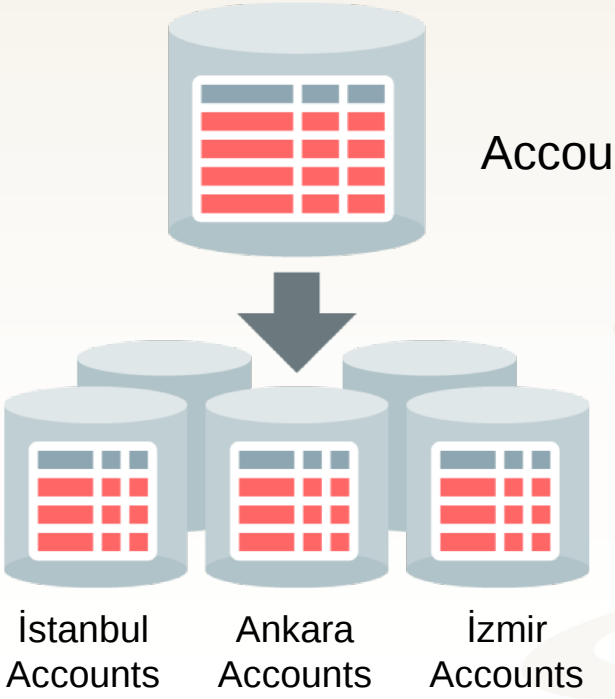
- Sharding ve replication birbirinden bağımsız (ortogonal)'dır
- Dolayısı ile aynı sistem içinde birlikte kullanılabilirler

Distribution Models

- NoSQL çözümler cluster çözümleri daha ön plana çıkarmaktadır
- Ancak cluster çözümlerin uygulamada transparan biçimde uygulanması mümkün değildir
- Cluster mimari, verinin yapısı ve saklanması ile ilgili kurallarda değişiklik tetiklemiştir (aggregate oriented)
- Cluster mimari, hesaplama ile ilgili kurallarda da değişikliğe gidilmesine neden olmuştur (map reduce)

Sharding

- Verinin bölümlere ayrılarak, her bir bölümün ayrı bir node üzerinden yönetilmesini ve erişilmesini sağlar



Sharding

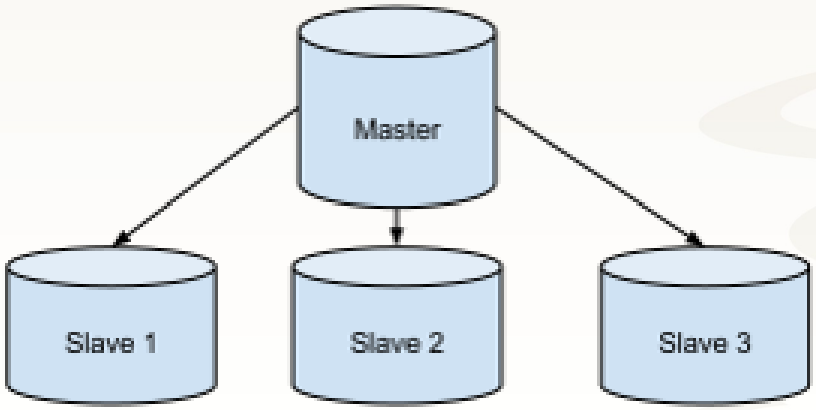
- Aggregate, sharding için iyi bir atomik birimdir
- Gruplama coğrafi lokasyona, yükün node'lar arasında eşit olarak dağıtılmasına göre yapılabilir
- Gruplama mantığı uygulama düzeyinde ele alınması gerekebilir
- NoSQL çözümler sharding'i tamamen DB düzeyinde halletmeye (auto-sharding) odaklanırlar

Sharding

- Sharding, hem okuma hem de yazma performansını iyileştirir
- Ancak resilience açısından çok bir iyileşme sunamaz
- Hatta cluster sistemler daha az maliyetli olup sorun çıkarma olasılıkları daha yüksek olduğundan hata oranlarının artmasına da yol açabilir

Master-Slave Replication

- Veri master'a yazılır ve bütün slave node'lara kopyalanır
- Okuma işlemleri slave node'lardan yapılabilir



Master-Slave Replication

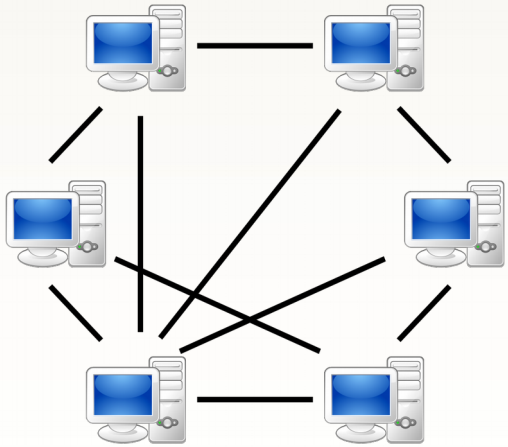
- Okuma senaryolarının yoğun olduğu sistemler için daha uygundur (read scalibility)
- Yazma senaryolarının yoğun olduğu senaryolarda çok avantajlı değildir (write scalibility)

Master-Slave Replication

- Resilience'a da olumlu etkisi vardır
- Master node erişilmez olsa bile okuma senaryoları için slave node'lar cevap vermeye devam (read resilience) ederler
- Dezavantajlarından birisi inconsistency ihtimalini artırmasıdır (read inconsistency)
- Diğer problem ise master'ın single point of failure olmasıdır

Peer-to-Peer Replication

- Master kavramı yoktur, bütün replica node'lar eşittir
- Yazma ve okuma işlemleri bütün node'larda gerçekleştirilebilir



Peer-to-Peer Replication

- Master'ın single point of failure problemi ve write darboğaz problemlerini çözmektedir
- Ancak inconsistency problemi devam etmektedir
- Read inconsistency'ler genelde geçicidir
- Ancak aynı kayıt üzerinde iki farklı node'a yapılabilecek yazma işlemlerinden doğacak write inconsistency kalıcı bir problemdir

Sharding & Replication

- Columnar store'larda sharding ve peer to peer replication birlikte kullanılmaktadır
- Cluster sistemde birden fazla peer node yer alır ve veri bu node'lar üzerinde aggregate model'e göre dağıtılır
- Replication Factor N = 3 peer to peer replication için iyi bir başlangıç noktasıdır
- Bir node'un fail etmesi durumunda availability'de herhangi bir sorun olmayacaktır

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com