

NoSQL ve Map Reduce Algoritması



Map-Reduce

- Hesaplama işlemlerinin mümkün olduğunca cluster sistemlerdeki mevcut makinalardan yararlanmasını sağlayan algoritmadır
- Her bir node'un işlem gücünden (CPU) maksimum yararlanıp ve verinin mümkün olduğunca aynı makinada tutularak hesaplamanın dağıtık biçimde yapılmasını hedefler

Map-Reduce

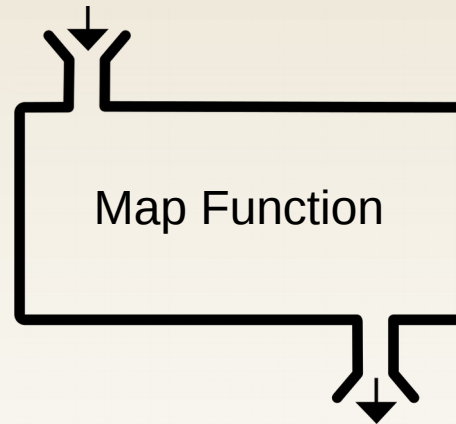
- İlk adım map işlemidir, burada input bir aggregate olabilir, örneğin Order
- Output ise bu aggregate'den elde edilen bir grup key-value ikilisidir, örneğin her ayrı Product için, satılan product sayısı ve ödenen ücret
- Map fonksiyonunun her bir aggregate'e ayrı ayrı uygulanması söz konusudur, örneğin satılan belirli bir Product sayısının ve fiyatının tespit edilmesi

Map-Reduce

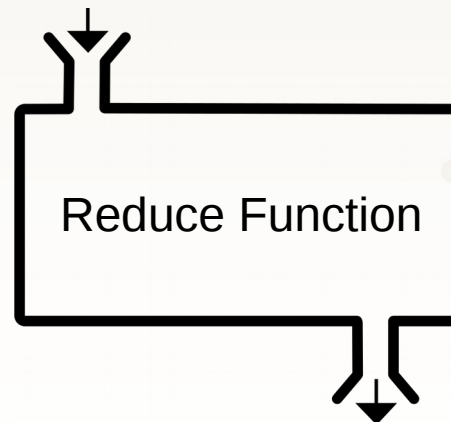
- İkinci aşamada reduce işlemi gerçekleşir
- Map işlemi sadece tek bir kayıt üzerinde işlem yapar
- Reduce ise aynı key değerine sahip birden fazla map işleminin çıktısı üzerinde çalışır
- Input olarak bu map çıktılarını alır, output olarak da bu input'un birleştirilmesinden elde edilen değeri döner, örneğin satılan toplam Product sayısı, ödenen toplam ücret

Map-Reduce

Aggregate



Key-Value Pairs



Summarize to
Single output

Map-Reduce ve Parallelism

- Her bir map operasyonu ayrı bir aggregate üzerinde çalışır
- Dolayısı ile map operasyonları aggregate düzeyinde paralel koşturulabilir
- Reduce operasyonları ise key düzeyinde ayrışırlar
- Dolayısı ile reducer'lar ise key bazında gruplanarak paralel koşturulabilirler

Map-Reduce Pipeline

- Aynı yapıda input ve output'a sahip reducer'lar pipe & filter örüntüsü ile bir pipeline oluşturabilirler
- Bir reducer'ın çıktısı, diğerinin girdisi olur
- Bu sayede paralel koşturma imkanı artar, network üzerinde veri transferi azalır

Map-Reduce ve Materialized View

- Map-reduce işlemlerinin hesaplamaları zaman alan işlemlerdir ve sürekli tekrarlanmaları gerekir
- Sıklıkla kullanılan kısımları performans açısından materialized view şeklinde saklanabilir

Incremental Map-Reduce

- Veri sürekli yenilendiği için map-reduce işlemlerinin de sürekli tekrarlanması gerekir
- Map-reduce işlemlerinin ilk aşaması verinin en çok olduğu ve işlemin de yoğunlaştığı kısımdır
- Yeni veri geldiğinde veya veri değiştiğinde sadece değişen kısımlar için map-reduce işlemlerinin çalıştırılması performans açısından yararlıdır

Incremental Map-Reduce

- Bunun için map-reduce işleminin yapısı incremental güncellemelere uygun hale getirilmelidir
- Map işlemlerinde aggregate veri birbirinden izole olduğu için incremental süreçte bu bölüm daha kolay ele alınır
- Reduce kısmı ise map işleminin çıktısına bağımlı olduğu için buradaki herhangi bir değişiklik reduce operasyonunun yeniden çalıştırılmasını gerektirir

Incremental Map-Reduce

- Reduce kısmı da belirli bir ölçüde paralel hale getirilebilirse bu bölüm de sadece incremental değişiklikler için çalıştırılabilir

İletişim

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

