Büyük Veri İçin İstatistiksel Öğrenme (Statistical Learning for Big Data)

M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2017." ve "Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman, Stanford University, 2011." kitapları kullanılarak hazırlanmıştır.

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

MapReduce ve Yazılım Kümesi

- Günümüz veri madenciliği uygulamaları, **çok büyük boyutlu verilerin hızlı şekilde yönetilmesini gerektirmektedir.**
- Çoğu uygulamada, **veri** son derece **düzgün yapıdadır** ve **eş zamanlı çalışmayı destekler.**
- Web sayfalarının ranking işleminde boyutları milyarları bulan matrislerin çarpımı yapılmaktadır.
- Sosyal ağlarda arkadaş arama işleminde milyonlarca düğüm (node) ve milyarlarca bağlantıdan (edge) oluşan graflar kullanılır.
- **Bu tür uygulamalar için eşzamanlı çalışmayı destekleyen programlama sistemleri geliştirilmiştir.**
- Bu sistemler **Ethernet** ağı üzerinden birbirine bağlı **hesaplama düğümlerinden oluşabilir** ve **dağıtık dosya sistemi** (distributed file system) kullanırlar.

3

MapReduce ve Yazılım Kümesi

- Dağıtık dosya sistemleri için yüksek seviyeli programlama sistemleri geliştirilmiştir.
- **Bu yeni programlama sistemlerinin temelinde MapReduce programları vardır.**
- **MapReduce** büyük ölçekli verilerde **etkin hesaplamayı sağlar** ve hesaplama sırasındaki donanımsal **hataları tolere edebilir.**
- MapReduce programları yüksek seviyeli programlama dilleriyle oluşturulabilir.
- MapReduce geliştirilirken SQL kullanılabilir.

4

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

5

Dağıtık Dosya Sistemleri

- Çoğu işlem, bir CPU, cache, main memory ve lokal disk kullanılarak gerçekleştirilebilir.
- Geçmişte, paralel işlem uygulamaları çok işlemcili ve özel donanımlı paralel bilgisayarlarla gerçekleştirilmekteydi.
- **Büyük ölçekli Web servislerinin yaygınlaşmasıyla birlikte, birbirine doğrudan bağlı olmayan çok sayıda node üzerinde işlemler yapılmaktadır.**
- Bu tür işlem düğümleri özel amaçlı bilgisayarlara göre çok düşük maliyete sahiptir.
- Bu sistemler, **dağıtık ve paralel çalışmanın avantajlarına sahiptir**, ancak çok sayıda bağımsız bileşenden kaynaklanan güvenilirlik (reliability) **problemleriyle karşı karşıyadır.**

6

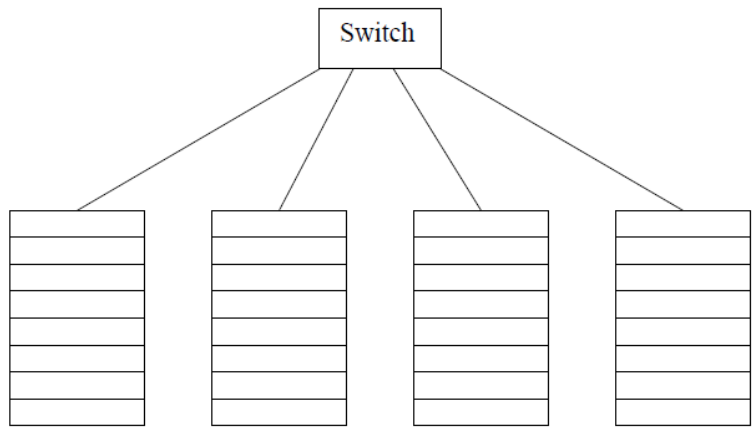
Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

7

Düğümlerin fiziksel organizasyonu

- Cluster computing yapısında işlem node'ları rack kabin üzerindedir.
- Rack kabin üzerindeki düğümler gigabit Ethernet ile birbirine bağlıdır.
- Birden fazla rack kabin birbirine anahtar (switch) ile bağlanır.



8

Düğümlerin fiziksel organizasyonu

- Çok sayıda düğüme sahip **sistemlerde** sıklıkla **bir veya birkaç düğümde arıza oluşabilir.**
- Bazen bir rack kabindeki düğümlerin tamamı çalışmayabilir.
- Kısa sürede tamamlanacak **işlemler** arızalar nedeniyle **uzayabilir** veya başarılı bir şekilde **tamamlanamayabilir.**
- **Dosyalar fazladan kopyalar halinde saklanabilir.** Böylelikle, bir dosyada sorun olursa diğer kopyası kullanılabilir.
- **İşlemlerin tamamı görevler halinde parçalanıp düğümlere dağıtılabilir.** Böylelikle, bir görev restart edilirken diğer görevler etkilenmez.

9

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

10

Büyük ölçekli dosya sistemi organizasyonu

- Cluster computing yapısında dosya sistemlerinin dağıtık (Distributed File System - DFS) olması gereklidir.
- Google File System (GFS) dağıtık dosya sistemidir (Google).
- Hadoop Distributed File System (HDFS) açık kaynak dağıtık dosya sistemidir (Apache).
- CloudStore, açık kaynak dağıtık dosya sistemidir (Kosmix).
- **Dosya boyutları çok büyüktür** (terabyte seviyesinde).
- **Dosyalar nadiren güncellenir.** Aralıklarla yeni veriler eklenir.
- Dosyalar parçalar (chunks) halinde ve **birden fazla kopya olarak farklı rack kabinlerdeki düğümlerde saklanır.**
- **Dosya parçalarının bulunduğu yeri saklamak** için küçük ayrı bir dosya **(master node) kullanılır.**

11

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

12

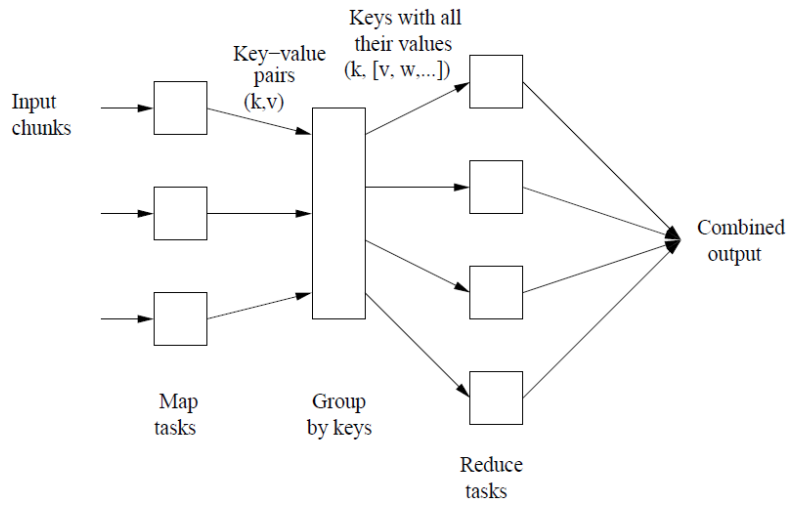
MapReduce

- **MapReduce, çok sayıdaki büyük ölçekli işlemi donanım hata toleransına sahip bir şekilde gerçekleştirir.**
- MapReduce için, Map ve Reduce isimli iki fonksiyonun yazılması gereklidir.
- MapReduce aşağıdaki işlemleri gerçekleştirir:
  - **Belirli sayıda Map görevi (task) tanımlanır.** Dağıtık dosya sistemindeki parçaları sıralanmış key-value ikilisi haline dönüştürür.
  - Giriş değerinden üretilecek key-value ikilisi Map fonksiyonu yazılırken belirlenir.
  - Her bir Map görevinin key-value ikilisi master controller tarafından toplanır ve key değerine göre sıralanır. **Key değerleri Reduce görevlerine dağıtılır.**
  - **Reduce görevleri** aynı anda **bir key üzerinde çalışır** ve **key ile ilişkili değerleri birleştirir.**
  - Birleştirme yönteminin nasıl çalışacağı Reduce fonksiyonu yazılırken belirlenir.

13

MapReduce

- MapReduce işlemleri için şematik gösterim.



14

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

15

Map işlemleri

- **Map görevi için giriş dosyaları** herhangi bir türdeki element'lerden oluşabilir.
- **Map görevlerinin çıkışları** ve **Reduce görevlerinin girişleri/çıkışları** anahtar-değer ikilisi şeklindedir.
- Map fonksiyonu parametre olarak bir element alır ve anahtar-değer ikilileri üretir.
- Buradaki anahtar, unique anahtar **olmak zorunda değildir.**

16

Map işlemleri

Örnek

- Doküman topluluğundaki her bir kelimenin tekrar sayısının bulunması klasik bir örnek uygulamadır.
- Map fonksiyonu için **giriş**, **doküman topluluğudur.**
- Her bir **doküman** ise **element**'i ifade eder.
- Map fonksiyonunda **anahtar olarak kelimeler**, **değer** olarak dokümandaki **tekrar sayıları** alınır.
- Her bir doküman için anahtar-değer ikilisi aşağıdaki gibi oluşur.

$$(w_1, 1),\ (w_2, 1), \ldots, (w_n, 1)$$

- Burada, w_x anahtar kelimeleri ifade eder ve aynı kelime birden fazla bulunabilir.
- **Aynı anahtara sahip** olan m tane **ikili (w, m) olarak birleştirilebilir.**

17

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

18

Anahtara göre gruplandırma

- Map görevleri başarılı bir şekilde sonuçlandığında **aynı anahtar değerine sahip ikililer gruplandırılır.**
- Belirlenen Reduce görev sayısına göre **master controller** anahtar-değer ikililerini **Reduce görev girişi olan lokal dosyalara aktarır.**
- Master controller, **anahtar - lokal dosya** eşleştirmesi için **hash fonksiyonu** kullanır.
- Hash fonksiyonu dışında başka eşleştirme yöntemleri kullanılabilir.
- **Aynı anahtar-değer ikilileri sadece bir Reduce görevine atanabilir.**
- **Master controller**, Map işleminin sonuçlarını **anahtara göre birleştirerek** ilgili **Reduce görevine atar.**

$$(k, v_1),\ (k, v_2), \ldots, (k, v_n) \quad \text{---------> } \quad (k, [v_1, v_2, \ldots, v_n])$$

19

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

20

Reduce işlemleri

- Reduce fonksiyonunun girişi gruplandırılmış **anahtar-değer listesidir.**
- Reduce fonksiyonunun çıkışı **anahtar-değer ikilileridir.**
- **Birden fazla reducer** aynı anda **dağıtık çalışabilir.**

Örnek:

- Kelime sayılarını bulma uygulamasında Reduce fonksiyonu **aynı değere sahip anahtarlara ait değerlerin toplamını hesaplayabilir.**
- Reduce çıktısı ise anahtar-tekrar toplamı (w, n) olur. **Burada *n* tüm dokümanlardaki tekrar sayısını ifade eder.**

21

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

22

Birleştiriciler

- Bazı uygulamalarda Reduce fonksiyonundaki işlemlerde **sıralama** (değer yer değişimi 3+4 = 4+3) ve **önceliklendirme önemli değildir.**
  (2+3)+4 = 2+(3+4)
  (2/3)+4 ≠ 2/(3+4)
- Bu durumda, **bazı reducer bileşenleri Map görevlerine aktarılabilir.**
- Örneğin, kelimelerin frekanslarının bulunmasında aynı anahtar kelimelere ait değerlerin toplanması Map fonksiyonunda yapılabilir.
- Map fonksiyonunda $(w_1, 1), (w_2, 1), \ldots, (w_n, 1)$ yerine (w, m) hesaplanabilir.
- Maksimum paralel çalışma için **her anahtara ayrı Reduce görevi atanabilir.**
- **Her bir Reduce görevi** de **ayrı bir düğümde çalışabilir.**

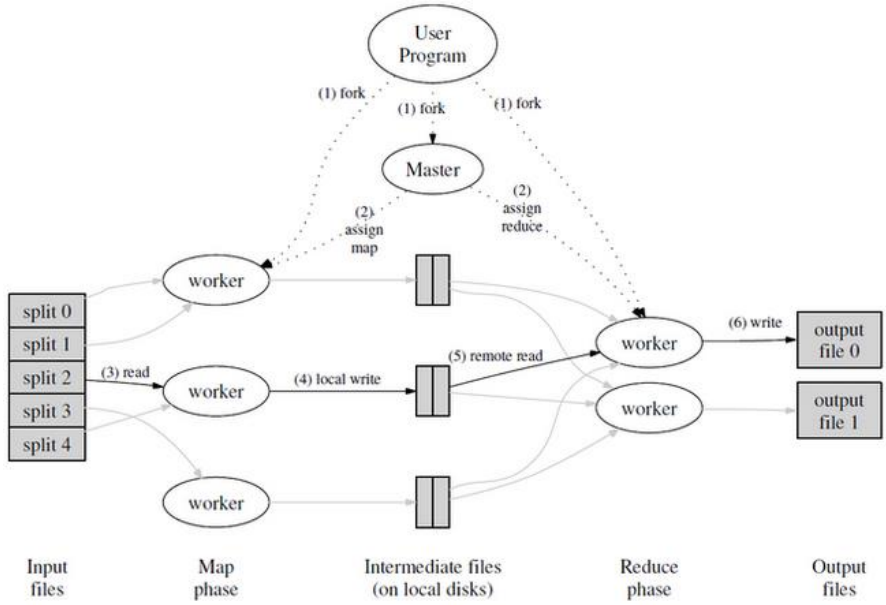
23

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

24

MapReduce işlemleri



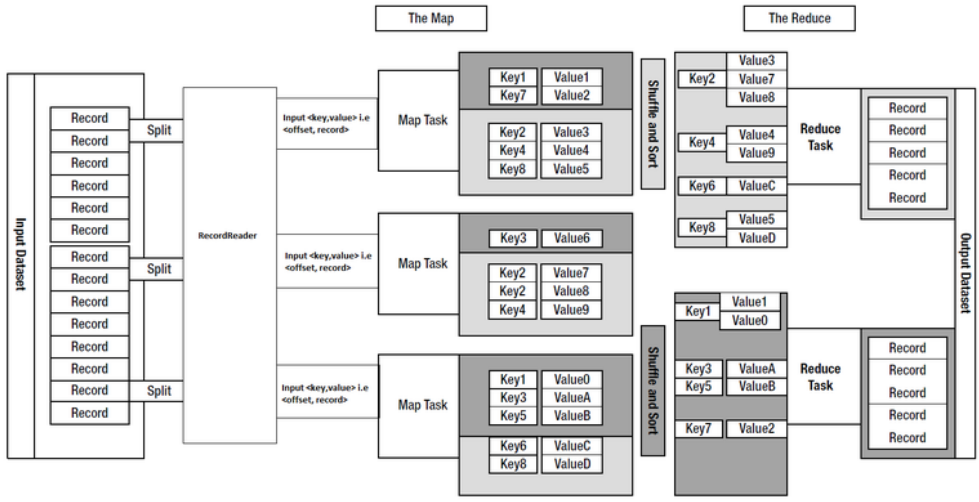
25

MapReduce işlemleri

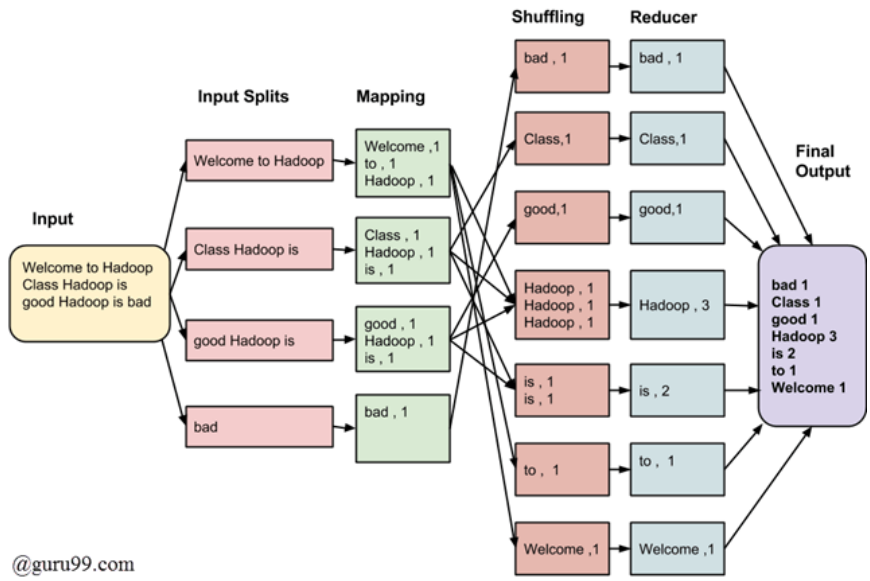
- Uygulama programı, bir **master controller process** ile belli sayıda **worker process'i** farklı düğümlerde başlatır.
- Bir **worker** process, **map görevi** veya **reduce görevi** yapabilir.
- **Uygulama programı giriş dosyasını** belirli sayıda **parçaya böler.**
- **Map görevi** process'i giriş dosyasından **kendisine atanmış kısmı okur.**
- Map görevi tarafından **elde edilen anahtar-değer ikilileri lokal diske kaydedilir.**
- **Reduce görevi** atanmış process, **diskten anahtar-değer ikililerini okur ve gerekli işlemi** (sıralama, gruplandırma) **yapar.**
- **Reduce görevi kendi çıkışını** kullanıcının belirlediği **dosyaya kaydeder.**
- Tüm **atama** ve **yönetme** işlemleri **master controller process tarafından yapılır.**

26

MapReduce işlemleri

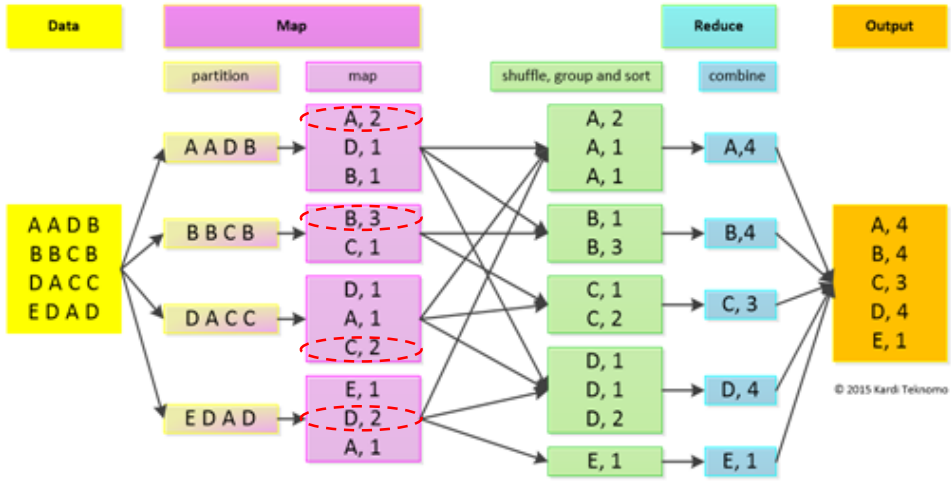


MapReduce işlemleri



MapReduce işlemleri

Map görevinde Reduce işlemi **yapılıyor.**



29

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

30

Node hataları

- MapReduce için en kötü durum **master controller process'in çalıştığı düğümün arızalanmasıdır.**
- Map ve Reduce görevlerini çalıştıran **diğer düğümlerdeki arızalar master controller process tarafından denetlenir** ve MapReduce başarılı bir şekilde tamamlanır.
- Arızalanan düğümdeki **Map görevleri başka bir worker'a atanır.**
- Sonuçların bulunduğu **dosyanın yeni lokasyonu ilgili Reduce görevlerine aktarılır.**
- Arızalanan düğümdeki **Reduce görevleri başka bir worker'a atanır.**

31

Konular

- MapReduce ve Yazılım Kümesi
- Dağıtık Dosya Sistemleri
  - Düğümlerin fiziksel organizasyonu
  - Büyük ölçekli dosya sistemi organizasyonu
- MapReduce
  - Map işlemleri
  - Anahtara göre gruplandırma
  - Reduce işlemleri
  - Birleştiriciler
  - MapReduce işlemleri
  - Node hataları
- MapReduce - Matris Çarpımı

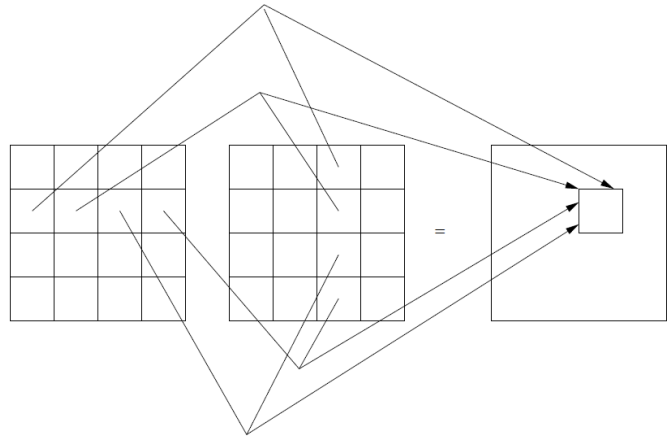
32

MapReduce - Matris Çarpımı

- P matrisi, M ve N matrislerinin çarpımına eşit olsun ($P = MN$).

$$p_{ik} = \sum_j m_{ij} n_{jk}$$

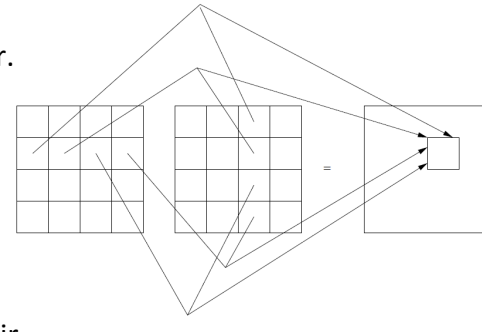
- m_{ij} i.satır j.sütun elemanı, n_{jk} j.satır k.sütun elemanı ve p_{ik} i.satır k.sütun elemanıdır.



33

MapReduce - Matris Çarpımı

- Bir matris üç özelliğin ilişkisi olarak düşünülebilir: **satır numarası**, **sütun numarası** ve **değer**.
- ***M*(*I*, *J*, *V*)** üçlü olarak (i, j, m_{ij}), ***N*(*J*, *K*, *W*)** üçlü olarak (j, k, n_{jk}) gösterilir. (***J* ortak özelliktir.**)
- ***MN* birleşiminden (*i*, *j*, *k*, *v*, *w*) elde edilir.** M 'den (i, j, v), N 'den (j, k, w) gelir.
- Bu beşli ile (m_{ij}, n_{jk}) ikilisi ifade edilir.
- **Çarpım sonucu $m_{ij}n_{jk}$ şeklinde (*i*, *j*, *k*, *v* x *w*) gösterilir.**



- **Çarpma işlemi için bir MapReduce, toplama ve birleştirme için ayrı bir MapReduce işlemi** yapılabilir.

34

MapReduce - Matris Çarpımı

1. Map işlemi

- M matrisinin her m_{ij} elemanı için **(*j, (M, i, m_{ij})*)** ikilisi elde edilir. N matrisinin her n_{jk} elemanı için **(*j, (N, k, n_{jk})*)** ikilisi elde edilir.

1. Reduce işlemi

- Her j elemanı için, M'den gelen **(*M, i, m_{ij}*)** elemanı ile N'den gelen **(*N, k, n_{jk}*)** elemanından, **anahtarı (*i, k*)** ve **değeri $m_{ij}n_{jk}$** olan anahtar-değer ikilisi oluşturulur **((*i, k*), $m_{ij}n_{jk}$)**.

2. Map işlemi

- Her **(*i, k*)** anahtarı için **((*i, k*), ($m_{i1}n_{1k}$, $m_{i2}n_{2k}$, …, $m_{ij}n_{jk}$))** ikilisi elde edilir.

2. Reduce işlemi

- Her **(*i, k*)** anahtarı için **((*i, k*), *toplam*)** ikilisi elde edilir.

35