

# HBase

Mimari ve Kullanım



# Kim?

- Erdem Ağaoğlu\*\*
- Nokta\* Altyapı-Teknoloji ekibi yazılımcısı
- Operasyon, sistem, veri analiz
- Önceden; web programcısı, yazılım kalitesi, dağıtık sistemler

\*: izlesene, alkışlarla yaşıyorum, sinemalar, blogcu, doviz, fotokritik, ...

\*\* : erdem@nokta.com, @agaoglu

A P A C H E  
HBASE

# Kapsam

- HBase nedir?
- Temel HBase mimarisi
- Kullanım örnekleri
- Sorular

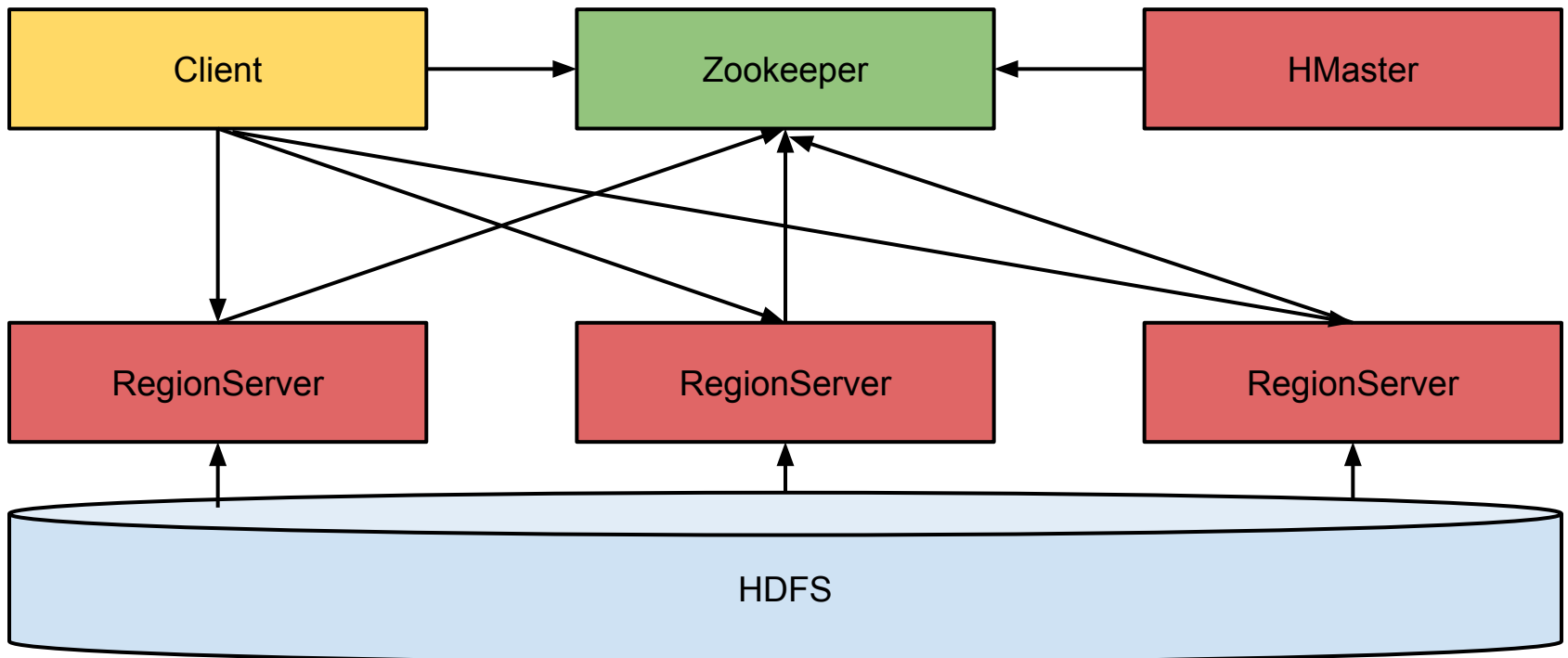
# HBase

- Dağıtık veritabanı
  - Birden çok makinede çalışacak şekilde tasarlanmış
- Sütun odaklı
  - Satır ya da obje değil
- Rastgele erişilebilen
  - HDFS değil
- Fikir Google BigTable, Uygulama Apache
- Facebook, StumbleUpon, Cloudera, Hortonworks, Trend Micro, ...

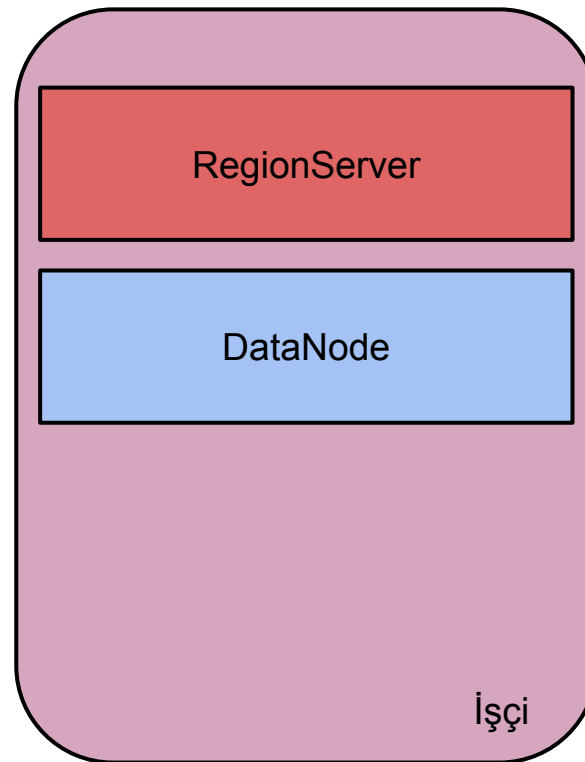
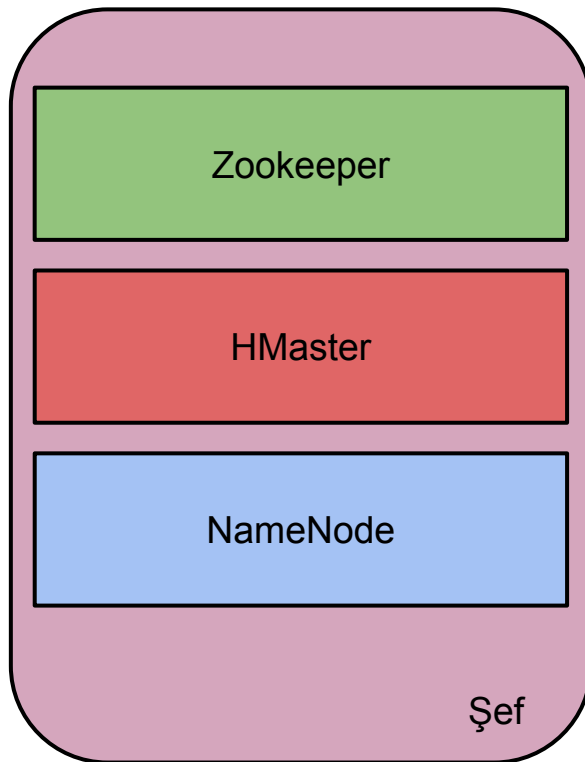
# Dağıtık

- Standart sunucularda çalışır
  - Özel donanım yatırımı gerektirmez
- Tek sunucuda çalışmaz
  - Test/Geliştirme hariç
- Sunucu eklenerek kapasite doğrusal olarak artırılabilir
- Belli bir kayba kadar hatalara dayanıklıdır

# Küme mimarisi

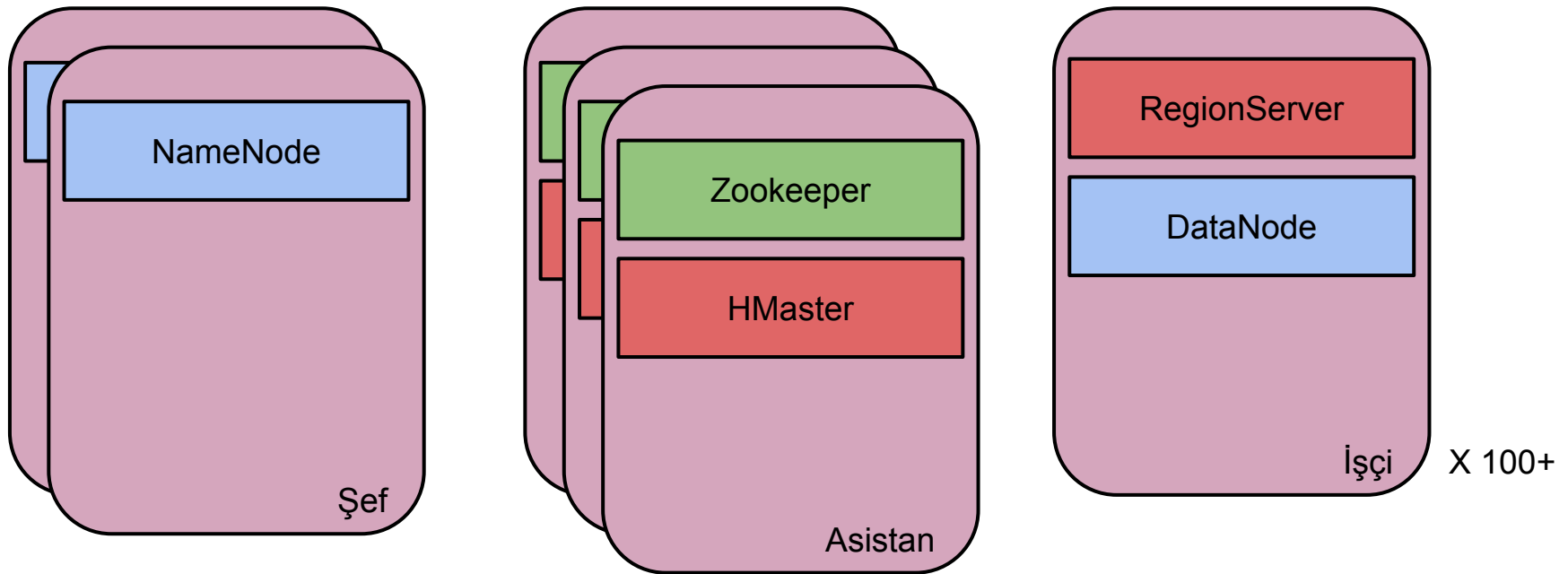


# Küme Kurulum (Küçük)



X 5-20

# Küme Kurulum (Büyük)





# Sütun Odaklı

- Statik şema tanımı yok
- Sadece temel veri kayıt detayları önden tanımlanır
- Veriler sütunlara yazılır
- Zaman ve sayı bazlı versiyonlama yapılır

# Tablo

```
CREATE TABLE t1 {  
  id int,  
  name string,  
  updated timestamp  
}
```

3 alanı bulunan basit  
RDB tablosu

```
create 't1', 'f1'
```

1 sütun ailesi bulunan  
basit HBase tablosu

A P A C H E  
**HBASE**

# Veri katmanları

1. Tablo
2. Satır
3. Alan

1. Tablo
2. Satır
3. Sütun ailesi
4. Sütun tanımlayıcı
5. Zaman damgası

# Örnek veri

## İlişkisel veritabanı

id	name	age	updated
1	Ahmet	25	2013-04-05
2	Mehmet	NULL	2013-04-06

## Sütun odaklı veritabanı

1	f1:age	25
	f1:name	Ahmet
	f1:updated	2013-04-05
2	f1:name	Mehmet
	f1:updated	2013-04-06

# Veri Kayıt

- Veriler her zaman sıralı
  - satır ID'leri ve sütun tanımlayıcılarına göre
- Veriler bu sıraya göre region'lara bölünür
- Region'lar farklı RegionServer'larda işlenir
- Yeterince büyük bir tablo binlerce Region'a bölünüp yüzlerce RegionServer'da paralel çalışabilir

# Erişim

- Java API
- Thrift
- REST
- MapReduce
- No SQL (?!)

# Java API

```
HTable table = new HTable(t1);  
table.put(new Put(1).add(f1, name, Ahmet)  
           .add(f1, age, 25)  
           .add(f1, updated, ...));  
...  
Result r = table.get(new Get(1));  
r.getValue(f1, name); // Ahmet  
r.getValue(f1, age);  // 25
```

# Java API

- Delete
- Scan
- Filter (Get & Scan)
- Increment (Atomic)
- checkAndPut (Atomic)
- ...



# Kullanım

- Çok yoğun yazma/okuma gerekiyor
- Erişim mantığı nispeten basit
- Veri çok fazla
  - Veri çok büyük değil ama diğer çözümler çok pahalı

# Örnek - Reklam

- Milyonlarca kullanıcıya hedefli reklam
- Sürekli;
  - Yeni kullanıcı, eksik veriler
  - Büyüyen veritabanı
  - Aktif yayın
  - Analiz
- Kullanıcı tablosu sürekli güncellenir
- MapReduce ile analizler yapıp raporlanır

# Örnek - YGZ Analiz

- Herhangi bir işlemin anlık sayılması/takibi
  - Şu an sitemde kaç kullanıcı var?
  - Son saatte en çok hangi hatalar oluştu?
  - ...
- 
- Satır ID: metrik + zaman, değerler incrementColumnValue ile toplanır
  - Saniyede yüzbinlerce kayıt, anlık takip edilir
  - OpenTSDB

# Sirada...

- BigTable => HBase
- Colossus ~> Coprocessors
- Dremel ~> Apache Drill

# Sorular?

[erdem@nokta.com](mailto:erdem@nokta.com)

[@agaoglu](#)