

HBase



Treinamento Hadoop – Big Data Open Source

Instrutor: Marcio Junior Vieira.
marcio@ambientelivre.com.br



O que é HBase

- Table storage for semi-structured data
- É um banco de dados orientada por colunas distribuída. HBase usa o HDFS por sua subjacente armazenamento e suporta os cálculos de estilo lote usando MapReduce e ponto consultas (leituras aleatórias).
- Conhecido como Hadoop Database
- criado pela Powerset



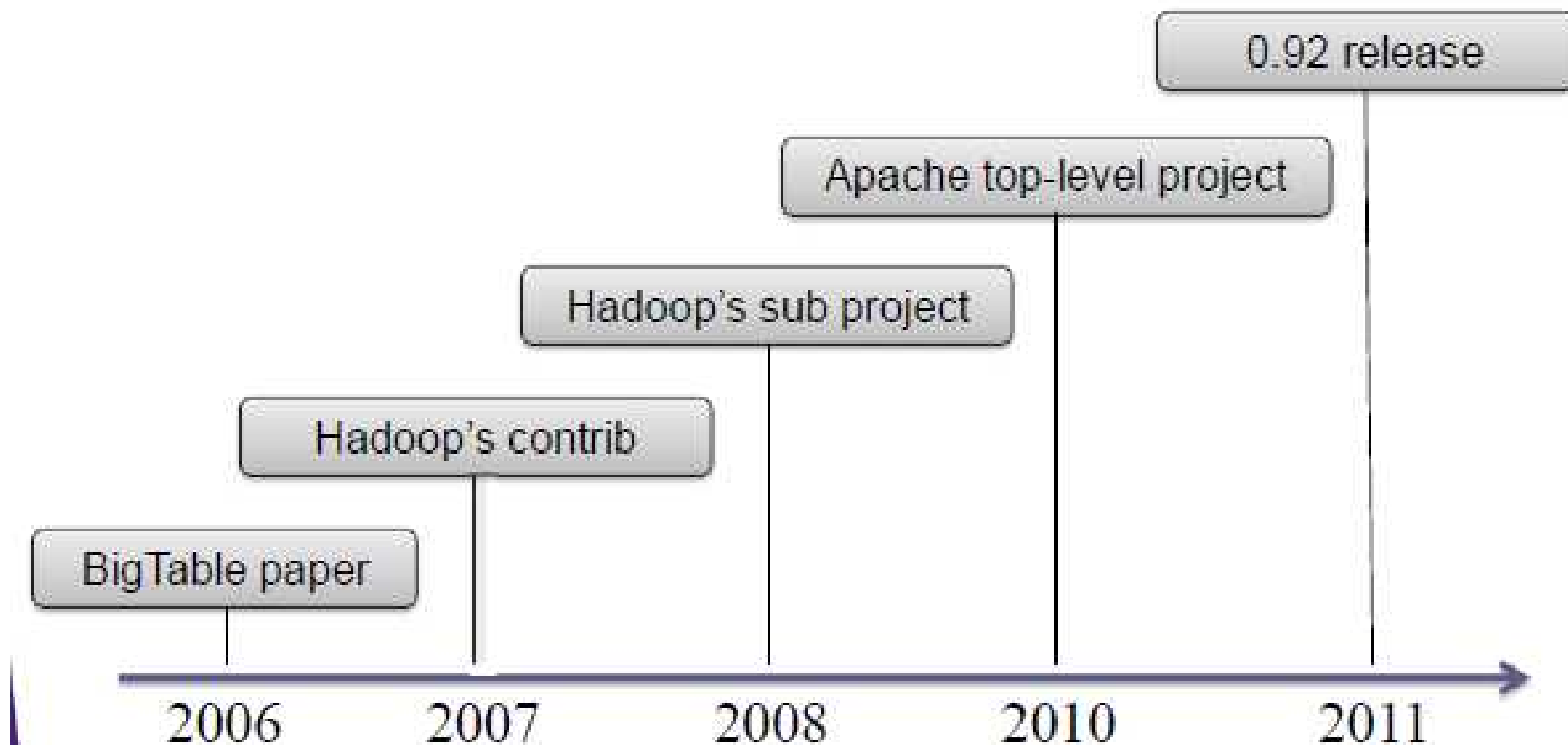
Características do HBase

- Suporta aleatória CRUD em tempo real operações (ao contrário do HDFS)
- Distribuída - projetado para atender grandes tabelas
 - Bilhões de linhas e milhões de colunas
- Funciona em um conjunto de hardware commodity
- Open-source, escrito em Java
- Tipo de "NoSQL" DB
 - Não fornece um acesso baseado em SQL
 - O que não aderem ao Modelo Relacional para armazenamento

Características do HBase

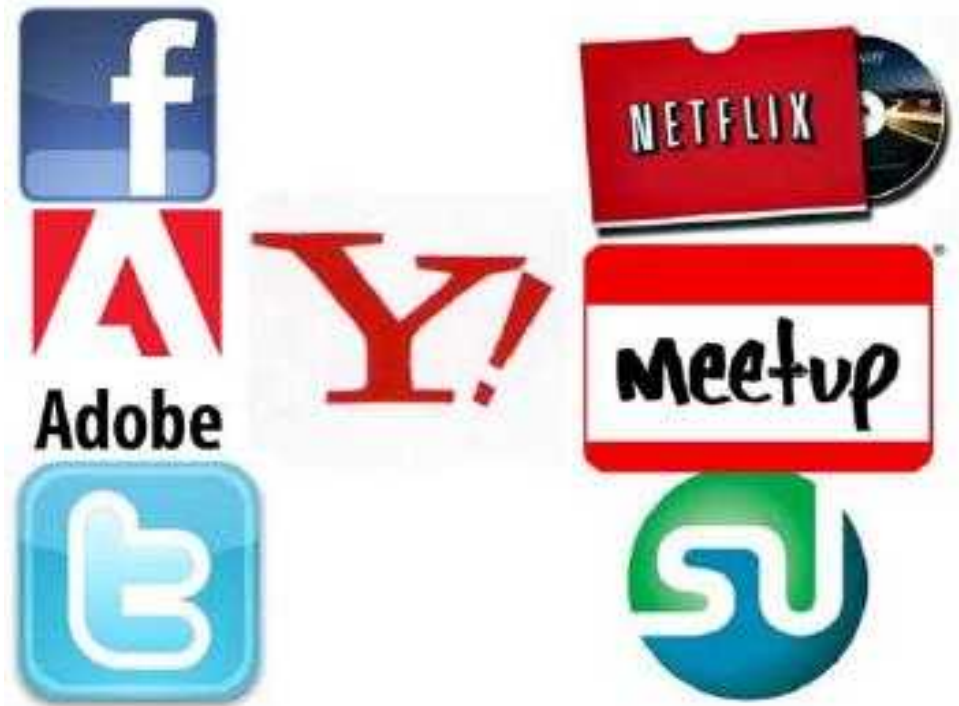
- horizontalmente escalável
- Fortemente consistente lê e escreve
- Simples Java API
- Integração com o Map / Reduce
- Thrift, Avro e Web-service REST-full
- Baseado no Bigtable do Google
 - Assim como BigTable é construído em cima do Google File System (GFS), HBase é implementado em cima do HDFS

Histórico



Quem Usa

- Facebook
- Adobe
- Twitter
- Yahoo!
- Netflix
- Meetup
- Stumbleupon



Quando Usar

- **Não é adequado para todos os problemas**
 - Comparado com RDBMs tem API muito simples e limitado
- **Bom para grandes quantidades de dados**
 - milhões ou bilhões de linhas
 - Se os dados é muito pequeno todos os registros vai acabarem um único nó, deixando o resto do grupo inativo
- **Hardware**
 - No mínimo 5 nós
 - Existem vários processos daemon gestão (Namenode, HBaseMaster, Zookeeper, etc)
 - HBase faz uso intensivo de memória e CPU

Quando Usar

- Grande quantidade de clientes / solicitações
- Ótimo para selects aleatórios individuais e grande range busca por chave
- Ótimo para esquema variável
 - Linhas podem diferir drasticamente
 - Se o seu esquema tem muitas colunas e a maioria deles são nulos

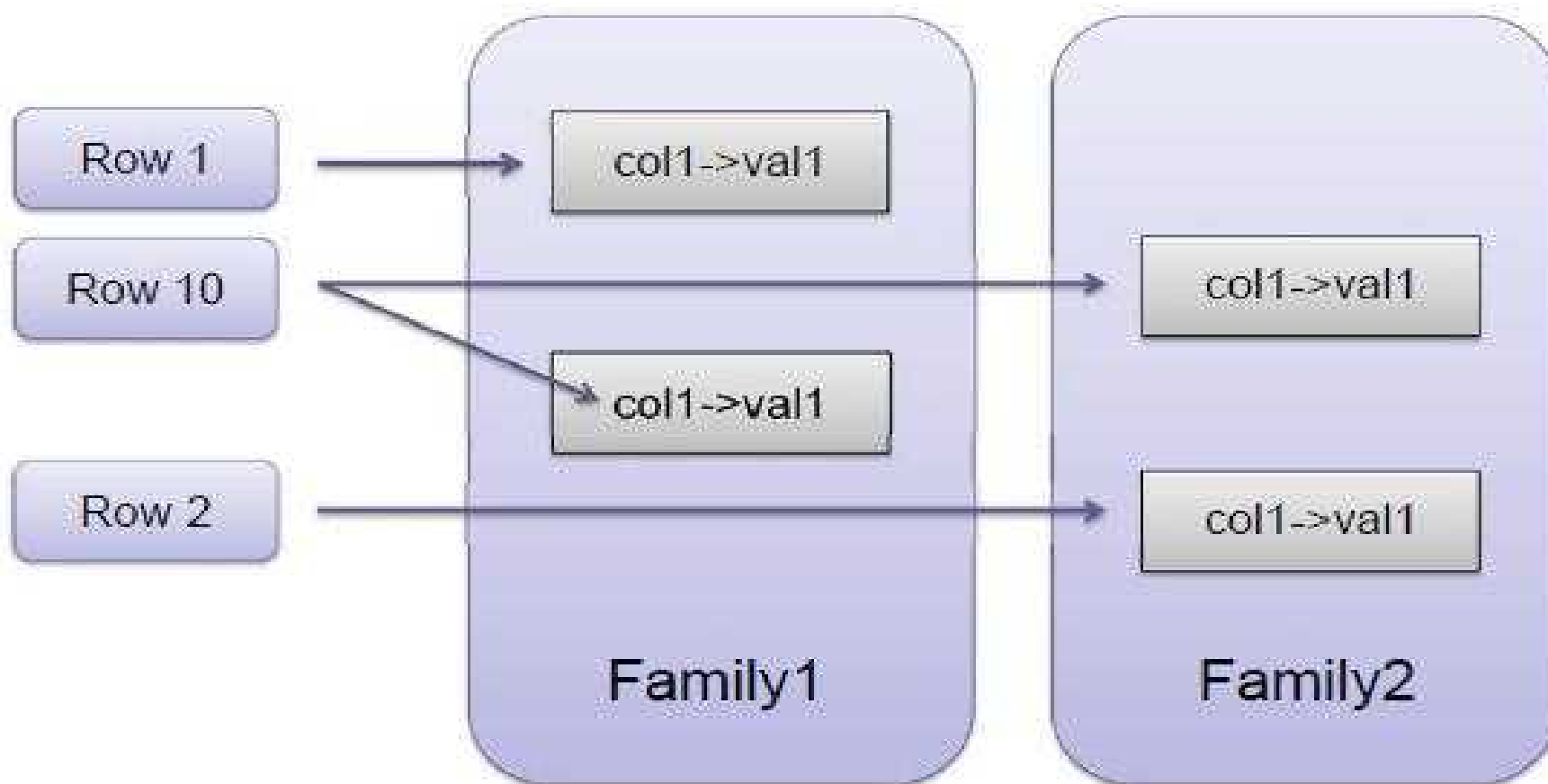
Quando não usar

- **Ruim para a recuperação tradicional RDBMs**
 - Aplicações transacionais
 - Relacionais Analytics
 - 'group by', 'join', e 'where column like ', etc
- **Ruim para o acesso de busca baseada em texto**
 - Alguns projetos de fornecer a solução que usar Hbase Lily = HBase + Solr
<http://www.lilyproject.org>

HBase Data Model

- Os dados são armazenados em **Tabelas**
- As tabelas contêm **linhas**
 - As linhas são referenciados por uma chave única
 - Key é uma matriz de bytes
 - Qualquer coisa pode ser uma chave: string, long e sua própria estruturas de dados serializados
- Linhas são feitas de colunas agrupadas em **famílias** de colunas
- Os dados são armazenados em **células**
 - Identificados por linha x coluna-família x coluna
 - O conteúdo da célula também é um array de bytes

Composição

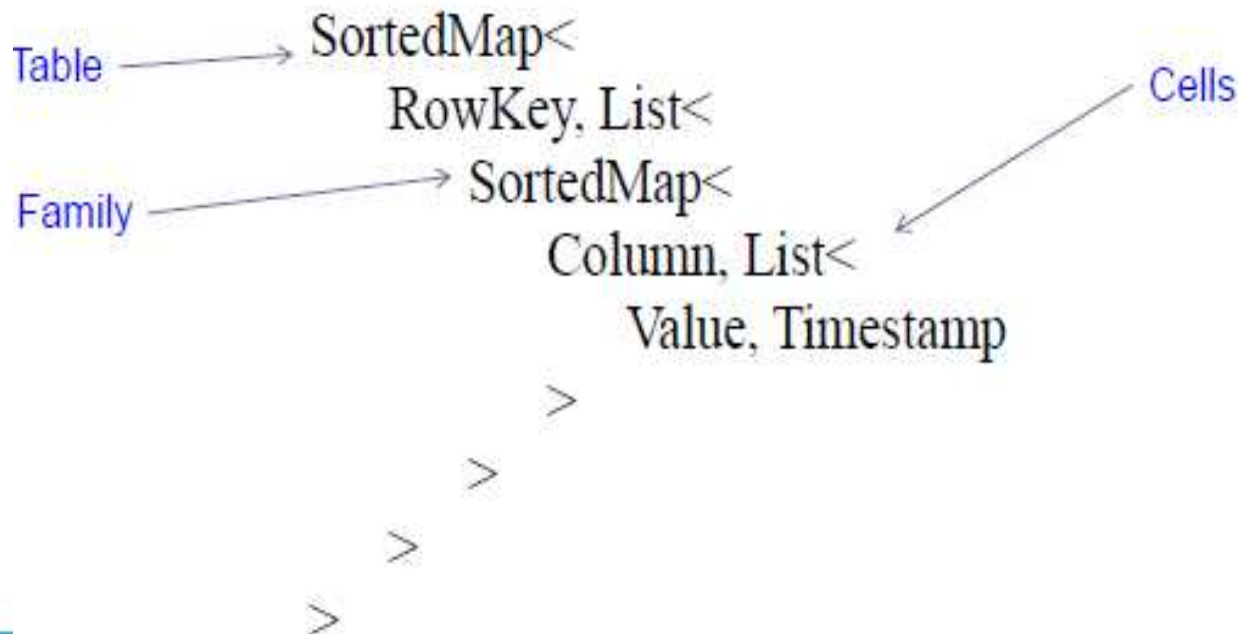


HBase TimeStamp

- Os valores das células são versionadas
 - Para cada célula várias versões são mantidas (3 por padrão)
- As versões são armazenadas em ordem decrescente timestamp
- Leia as últimas em primeiro lugar - otimização para ler o valor corrente

HBase Cell

- Value =
Table+RowKey+Family+Column+Timestamp
- Programming language style:

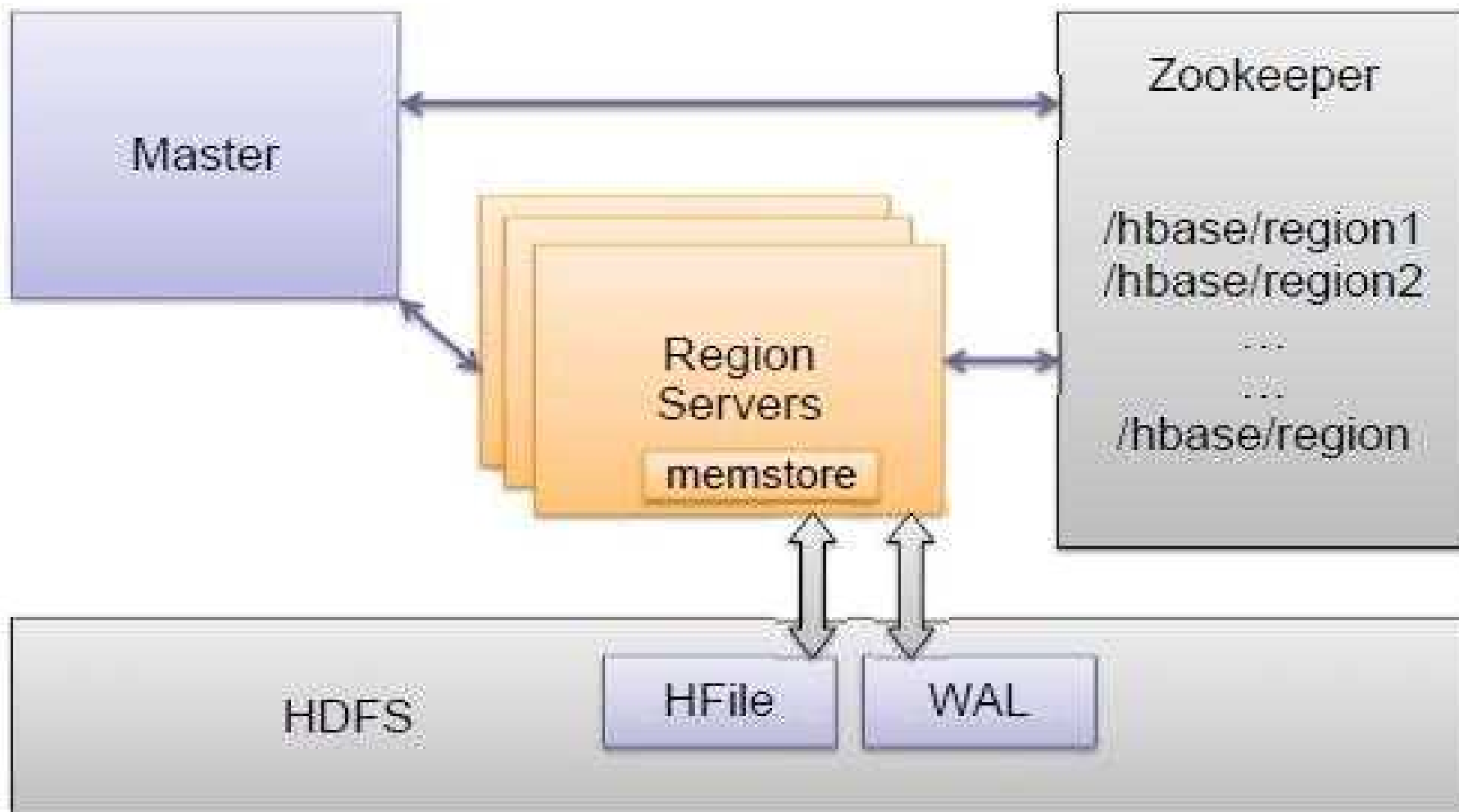


Hbase Cell

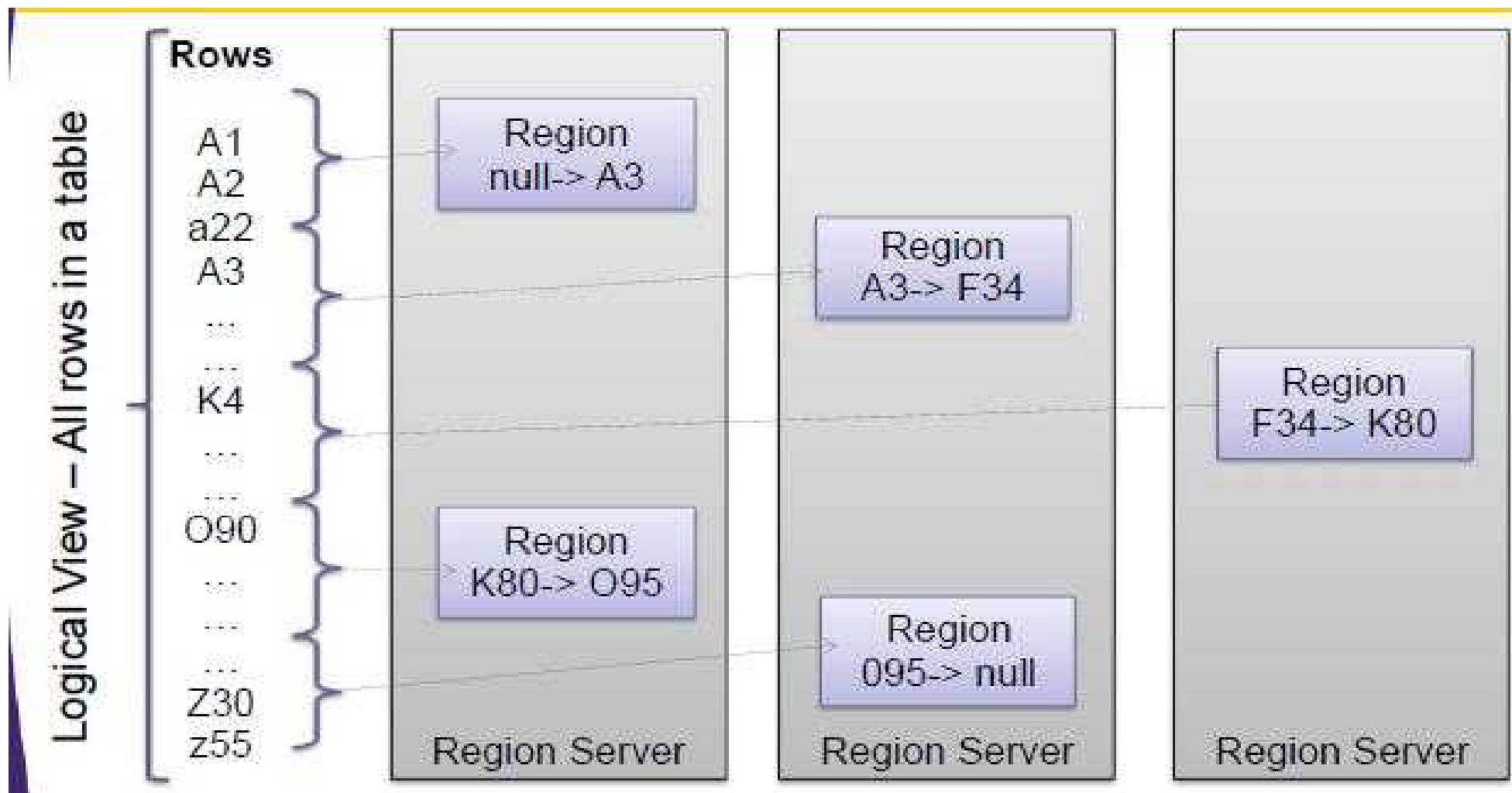
- Um exemplo - Lógico representação de como os valores são armazenados

Row Key	Time stamp	Name Family		Address Family	
		first_name	last_name	number	address
row1	t1	<u>Bob</u>	<u>Smith</u>		
	t5			10	First Lane
	t10			30	Other Lane
	t15			<u>7</u>	<u>Last Street</u>
row2	t20	<u>Mary</u>	Tompson		
	t22			77	One Street
	t30		<u>Thompson</u>		

Componentes



Distribuição



Acesso ao HBase

- HBase Shell
- Native Java API
- Avro Server
- **Hbql**
 - SQL syntax para Hbase
- PyHBase
- REST Server (texto, XML, JSON, buffers de
- protocolo, binário)
- Thrift (é um compilador esquema entre linguagens)

Referências

- <https://hbase.apache.org/>