Redis

@julioviegas

Engenheiro de Software

Palestrante

- ~15 anos trabalhando com tecnologia
- Instrutor, arquiteto, mentor e engenheiro de software e eletronica
- Participa de comunidades de usuarios: RSJUG, XP-RS, CEJUG, Open4Education, NoSQLBr...
- Sun Certified Trainer, SCEA 1 e 5, SCDJWS 1.4 e 5, SCWCD 1.4, SCBCD 5, SCJP 5, SCJA 1
- Engenheiro de Software da Summa Technologies e Instrutor Globalcode

Agenda

- NoSQL?
- Redis
- MongoDB
- CouchDB
- Memcached
- Cassandra

NoSQL?

- !SQL
- !ACID
- Distribuido
- Cloud
- Performance
- Persistencia variada
- Consistencia
- Sem schema/tipo

NoSQL?

- Persistencia ou RAM
- Objeto
- Tabular
- Tupla
- Documento
- Imagem
- Chave/Valor
- Hierarquico/Estruturado

NoSQL?

- Geracao web
- RDBMS nao eh arquitetura padrao
- Centralizar?
- Consistencia?
- Fail-over?
- Backup/recovery?
- Uso como cache: mais disco, mais memoria

NoSQL?

- CAP
  - Consistencia: nem sempre o dado visualizado eh o mais atual
  - Disponibilidade: sempre no ar, rapido e otima resposta
  - Particionamento: o sistema opera mesmo perdendo dados
- Disponibilidade versus consistencia
- Latencia versus consistencia
- Centralizacao versus particionamento

Redis

- Em memoria e persistencia
- Tipos de dados: Strings, Lists, Sets e SortedSets
- Atomico
- Clientes para varias linguagens
- Master/Slave
- Sharding manual
- Configuracao, gerenciamento e instalacao simples

Redis

- Instalacao: download, unpack, make
- Servidor: ./redis-server
- Cliente(teste): ./redis-cli set key value; ./redis-cli get key

MongoDB

- Muitas funcionalidades, grandes cases
- Orientado a documentos estilo JSON
- Indexacao de atributos
- Auto Sharding, Resharding, Replicacao, Alta Disponibilidade
- Muitas facilidades para consulta
- Atualizacoes de documentos ou dados pontuais internos
- Map Reduce
- Suporte a documentos grandes usando GridFS

MongoDB

- Online shell e tutorial em http://www.mongodb.org/#

MongoDB

- Instalacao: binarios ou via compilacao de fontes
- Servidor: ./mongod
- Cliente(teste): ./mongo
- Salvando dados: db.foo.save({ a : 1 })
- Lendo dados: db.foo.find()
- Consultas: db.foo.find({ $or : [{ a : 1 } , { b : 2 }] })

CouchDB

- JSON
- Javascript
- RESTful API
- Escrito em Erlang
- Indexavel
- Suporte a consultas
- Distribuido e replicavel

CouchDB

- Views
- Schema free
- Javascript Map/Reduce
- Versionamento
- BLOB

CouchDB

- Instalacao: unpack; ./configure; make; sudo make install
- Servidor: sudo -u couchdb ./couchdb
- Teste: http://localhost:5984/_utils
- Cliente: curl http://localhost:5984 # {"couchdb":"Welcome"}

CouchDB

- Requisicao HTTP RESTful PUT

```
PUT /example/some_doc_id HTTP/1.0
Content-Length: 29
Content-Type: application/json

{"greetings":"Hello, World!"}
```

- Resposta HTTP

```
HTTP/1.0 201 Created
Server: CouchDB/0.9.0 (Erlang OTP/R12B)
Etag: "1-518824332"
Date: Wed, 24 Jun 2009 13:33:11 GMT
Content-Type: text/plain;charset=utf-8
Content-Length: 51
Cache-Control: must-revalidate

{"ok":true,"id":"some_doc_id","rev":"1-518824332"}
```

CouchDB

- Requisicao HTTP RESTful GET(via browser)

http://localhost:5984/example/some_doc_id

- Resposta HTTP(via browser)

{"_id":"some_doc_id","_rev":"1-518824332","greetings":"Hello, World!"}

Memcached

- Em memoria apenas
- Chave/valor
- Suporte a timeout
- Reciclavel usando LRU
- Toda a logica de hashing, sharding e particionamento no cliente
- Protocolo binario ou texto. Em texto vc pode usar comandos via telnet.

Memcached

- Valores de ateh 1Mb
- stats
- atomico
- sem: autenticacao, failover, replicacao

Memcached

- Instalacao: unpack; ./configure; make; sudo make install
- Necessita de libevent
- Servidor: ./memcached -d -m 4096 -p 11212
- Cliente:

```
function get_foo(foo_id)
    foo = memcached_get("foo:" . foo_id)
    return foo if defined foo

    foo = fetch_foo_from_database(foo_id)
    memcached_set("foo:" . foo_id, foo)
    return foo
end
```

Cassandra

- Criado pelo Facebook, e usado em parte dele
- Baseado no Amazon Dynamo
- Usuarios: Digg, Reddit, Cisco Webex, Rackspace, Twitter
- Chave+valor
- Tipos dinamicos versionados com timestamp
- Atomicidade por chave
- Configuracao simples

Cassandra

- MySQL: muito IO imprevisivel!
- Solucoes baseadas em arquivos precisam de lock!
- Armazenamento descentralizado
- Tolerante a falhas
- Elastico: adicione mais servidores a quente
- Escala horizontalmente

Cassandra

- Disponibilidade e tolerancia a perda de particoes sobre consistencia(CAP)
- Consistencia configuravel
- Tradeoff: Consistencia versus Latencia
- Particionamento e replicacao baseado no Dynamo (Amazon) e modelo de dados estruturado(ColumnFamily) baseado no Bigtable(Google)

Cassandra

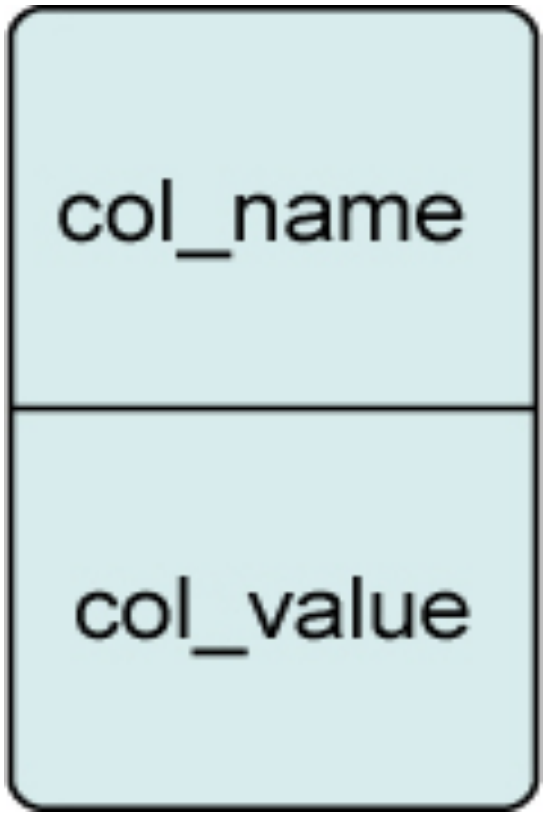
- Administracao minima
- Sem ponto unico de falha
- Particionador configuravel
- Replicacao configuravel: quantidade, rack e site “aware”
- Gossip, messaging, bootstrap, compaction
- Memtable
- Monitoracao e ferramentas de administracao
- Commit log

Cassandra

- Escritas: quorum(maioria) ou assincronas(1), sem leituras, posicionamento, rapida(0.12ms vs 300ms do MySQL), atomica, sempre garantida
- Leitura: memtable e +RAM, mais lenta que a escrita(15ms vs 350ms do MySQL), escala a bilhoes de linhas
- Consistencia: 1, quorum, all. Cuidado com a latencia!

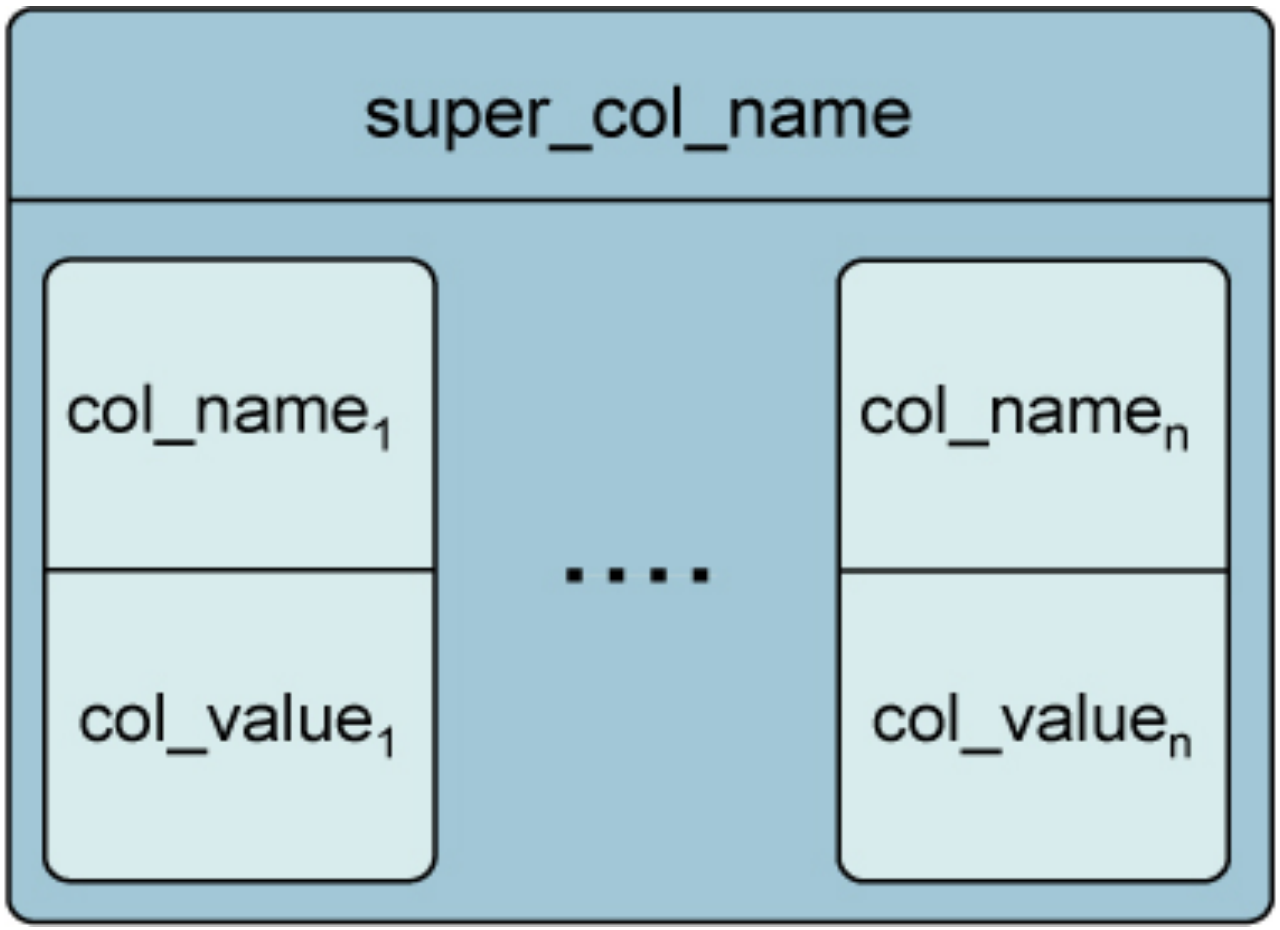
Cassandra

- Keyspace: grupo de chaves ou banco/schema
- Column: chave e valor ou tabela de uma coluna e linha



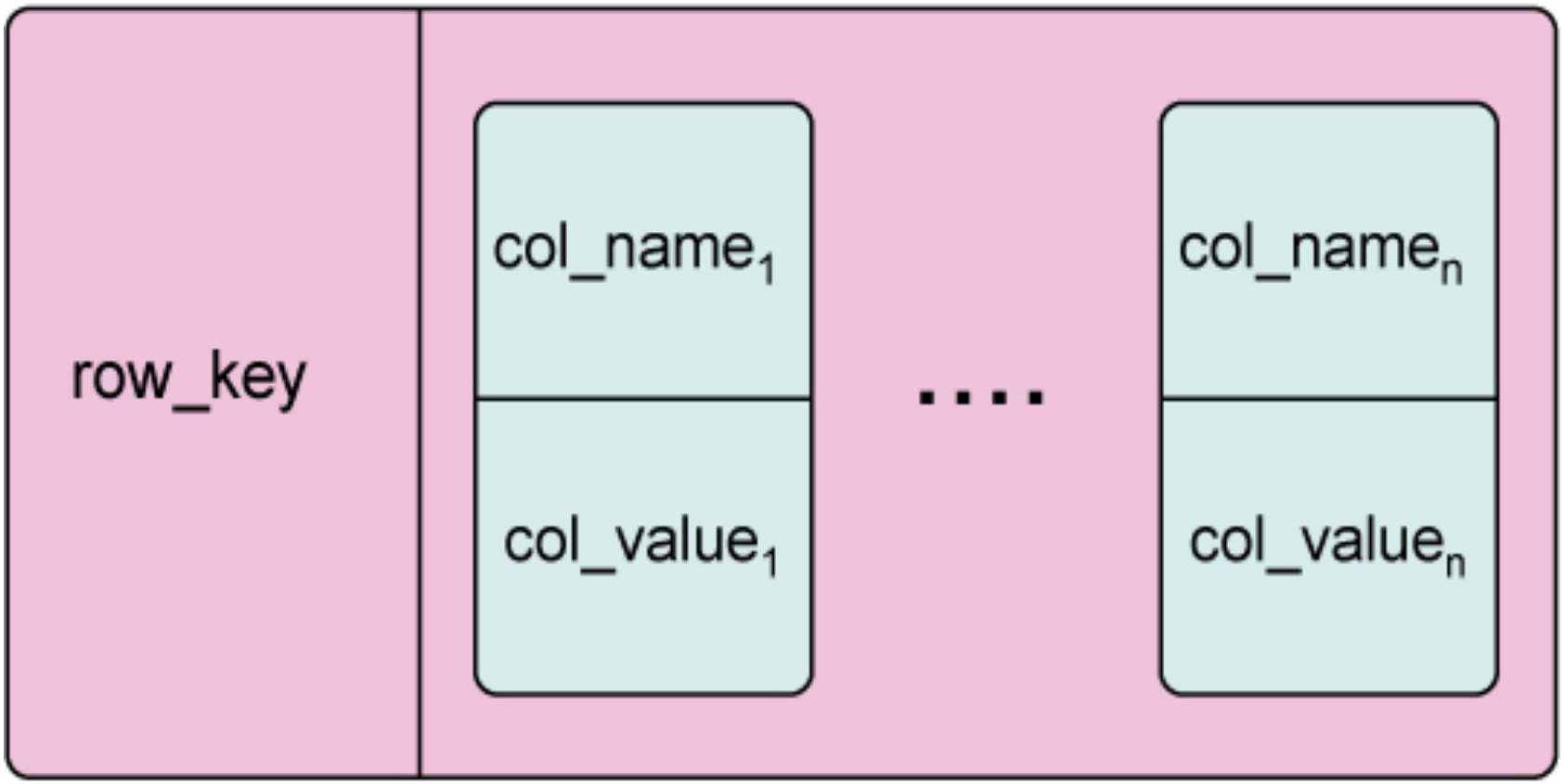
Cassandra

- SuperColumn: mapa ou tabela de n colunas e uma linha apenas



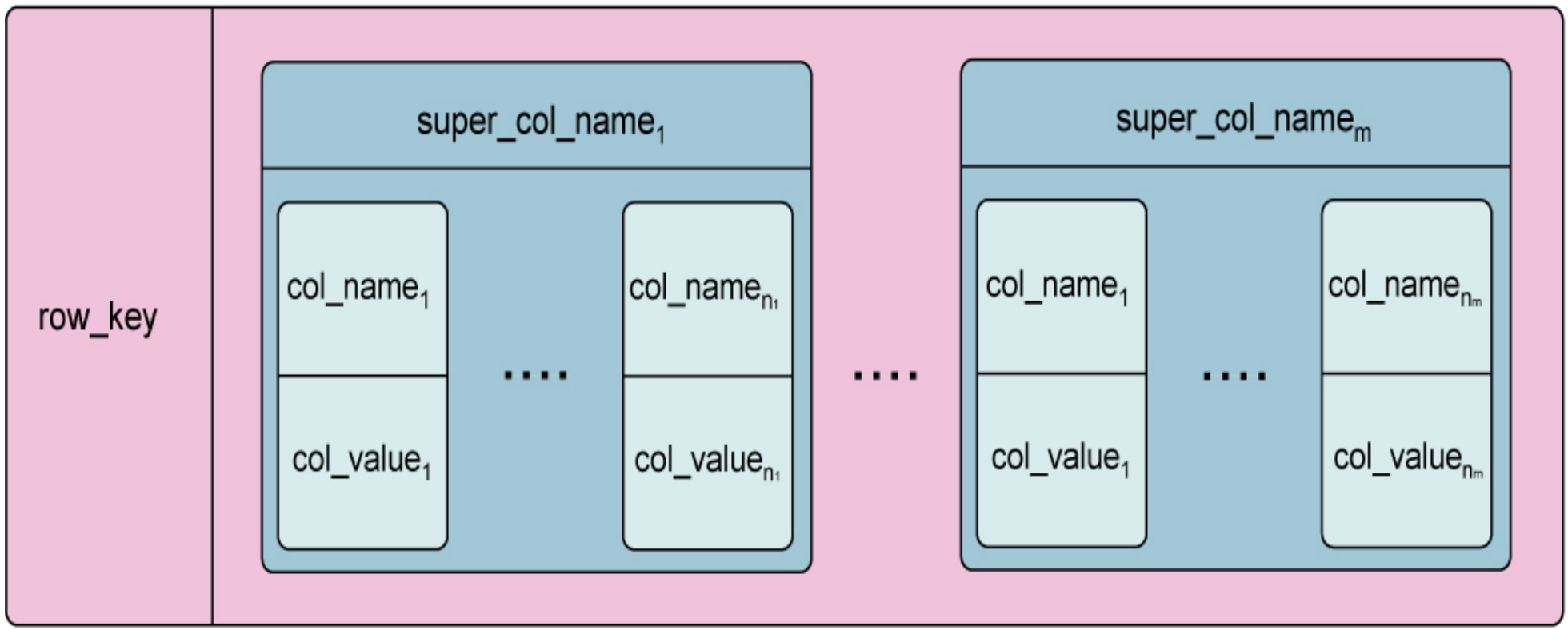
Cassandra

- ColumnFamily: lista de mapas ou tabela



Cassandra

- SuperColumnFamily: grupos de SuperColumn usando chave



Cassandra

- Configuravel em storage-conf.xml
- Ordenacao eh feita na insercao, usando o nome da coluna
- ColumnFamily CompareWith
- SuperColumn CompareSubcolumnsWith
- Tipos: BytesType, UTF8Type, LexicalUUIDType, TimeUUIDType, AsciiType, LongType

Cassandra

- $CASSANDRA_HOME/conf/storage-conf.xml
- ClusterName
- AutoBootstrap
- Keyspace
  - ColumnFamily
  - ReplicaPlacementStrategy & EndPointSnitch: Unaware, Rack ou Datacenter
  - ReplicationFactor

Cassandra

- Autenticador
- Particionador: Random por padrao ou ordenado
- InitialToken: armazena chaves com token similar
- CommitLogDirectory
- DataFileDirectory
- Seed
- ListenAddress/StoragePort
- ThriftAddress/ThriftPort

Cassandra

- Apache Thrift
- ConsistencyLevel: ZERO, ANY, ONE, QUORUM, ALL
- ColumnOrSuperColumn
- Column
- SuperColumn
- ColumnPath
- ColumnParent

Cassandra

- SlicePredicate
- SliceRange
- KeyRange
- KeySlice

- Mutation
- Deletion

Cassandra

- DEMO

???

Obrigado! :)

@julioviegas

Engenheiro de Software