Oficina de BD: Bigtable - Um Sistema de Armazenamento Distribuído para Dados Estruturados

Bruno Velasco [UFPR]

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber [Google]

Curitiba, 5 de Novembro de 2013

Introdução

Alternativa a SGBD

- Difícil de escalar
- Escalonamento vertical
- Alto custo
- Dificuldade em dados semi-estruturados

O que é Bigtable?

- É um sistema de armazenamento distribuído para dados estruturados
- Não dá suporte a operações 100% relacionais
- Escalável
- Autônomo

Bigtable

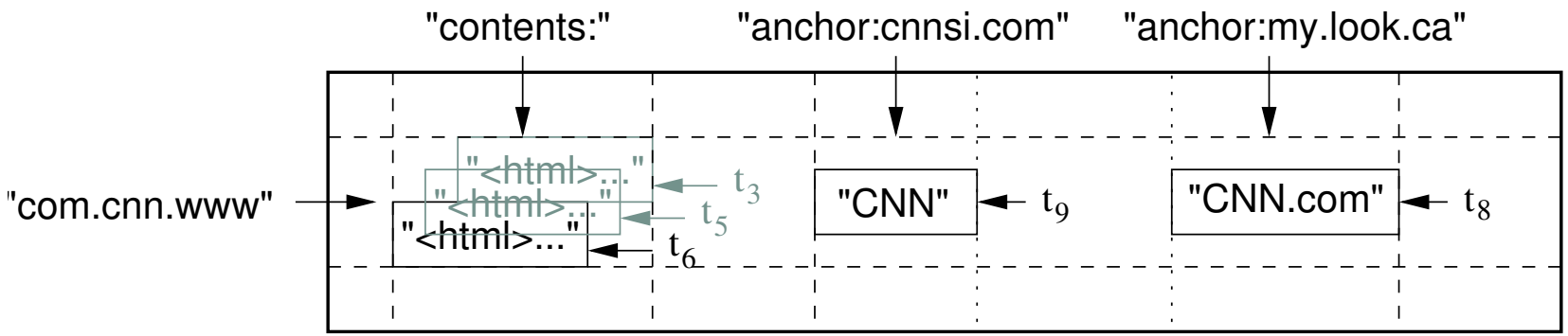
Utilizado em mais de 60 produtos

- Google Analytics
- Google Finance
- Google Earth
- Google ...

Objetivos

- Aplicabilidade diversa
- Escalabilidade
- Alto desempenho
- Alta disponibilidade

Modelo de dados



Características

- Bigtable é um mapeamento esparso, distribuído, persistente, multidimensional e ordenado
- (row:string, column:string, timestamp:int64) $\longrightarrow$ string

Modelo de dados

Linha

- Atomicidade
- Ordem lexográfica
- Intervalo conhecido por *tablet*

Coluna

- Unidade básica de controle
- *Column Families* - Grupamento
- Poucas CF, muitas colunas

Timestamp

- Cada célula possui várias versões
- *Garbage collector*

Exemplos

Escrevendo

```
// Open the table
Table *T = OpenOrDie("/bigtable/web/webtable");

// Write a new anchor and delete an old anchor
RowMutation r1(T, "com.cnn.www");
r1.Set("anchor:www.c-span.org", "CNN");
r1.Delete("anchor:www.abc.com");
Operation op;
Apply(&op, &r1);
```

Scan

```
Scanner scanner(T);
ScanStream *stream;
stream = scanner.FetchColumnFamily("anchor");
stream->SetReturnAllVersions();
scanner.Lookup("com.cnn.www");
for (; !stream->Done(); stream->Next()) {
  printf("%s %s %lld %s\n",
         scanner.RowName(),
         stream->ColumnName(),
         stream->MicroTimestamp(),
         stream->Value());
}
```

Blocos

Bigtable utiliza alguns serviços Google

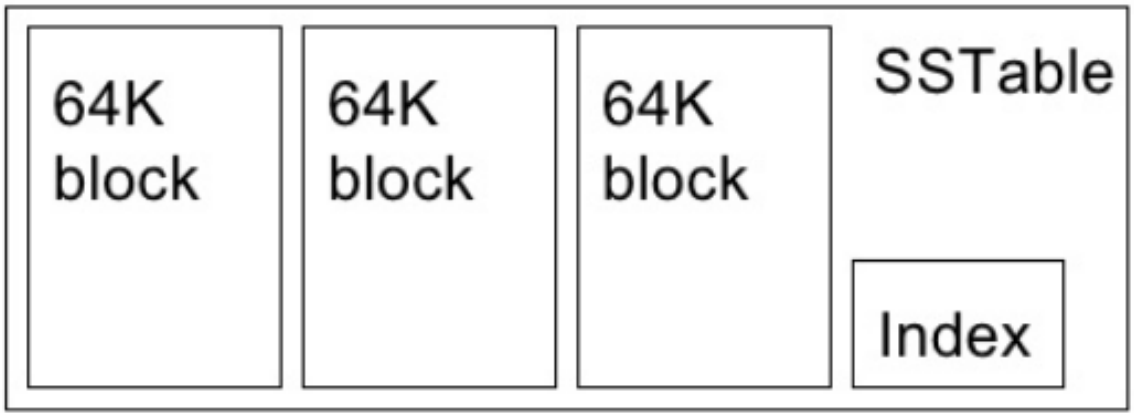
- GFS - logs e dados
- Chubby - controle de réplicas, e lock distribuído
- SSTable - formato de arquivo

Chubby

Características

- Serviço de lock distribuído
- Controle de réplicas (usa Paxos)
- Utiliza diretórios e arquivos
- Responsável por bootstrap do Bigtable
- Completamente responsável pelo Bigtable

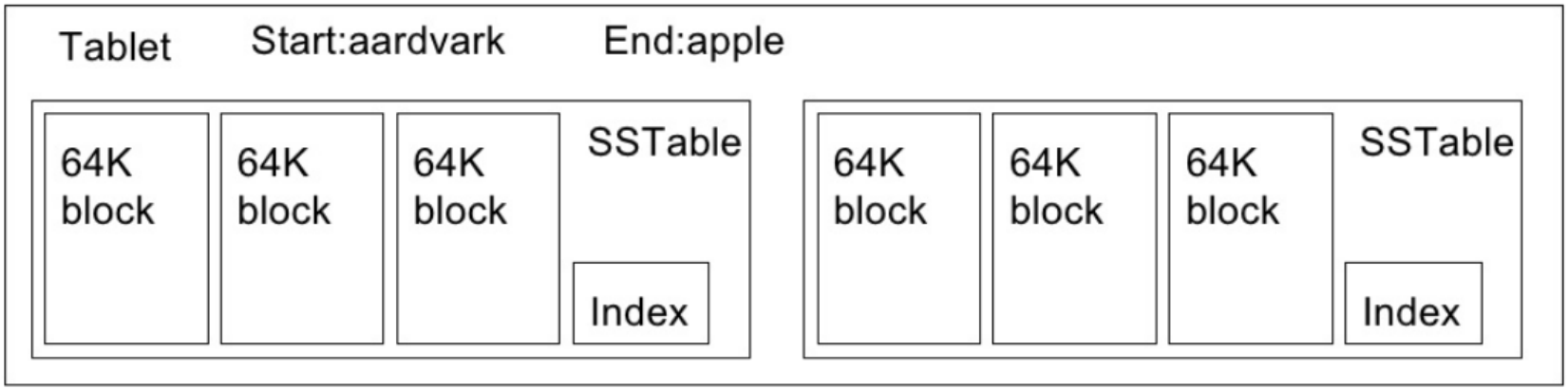
SStable



Características

- Formato de arquivo
- Chave - valor
- Imutável

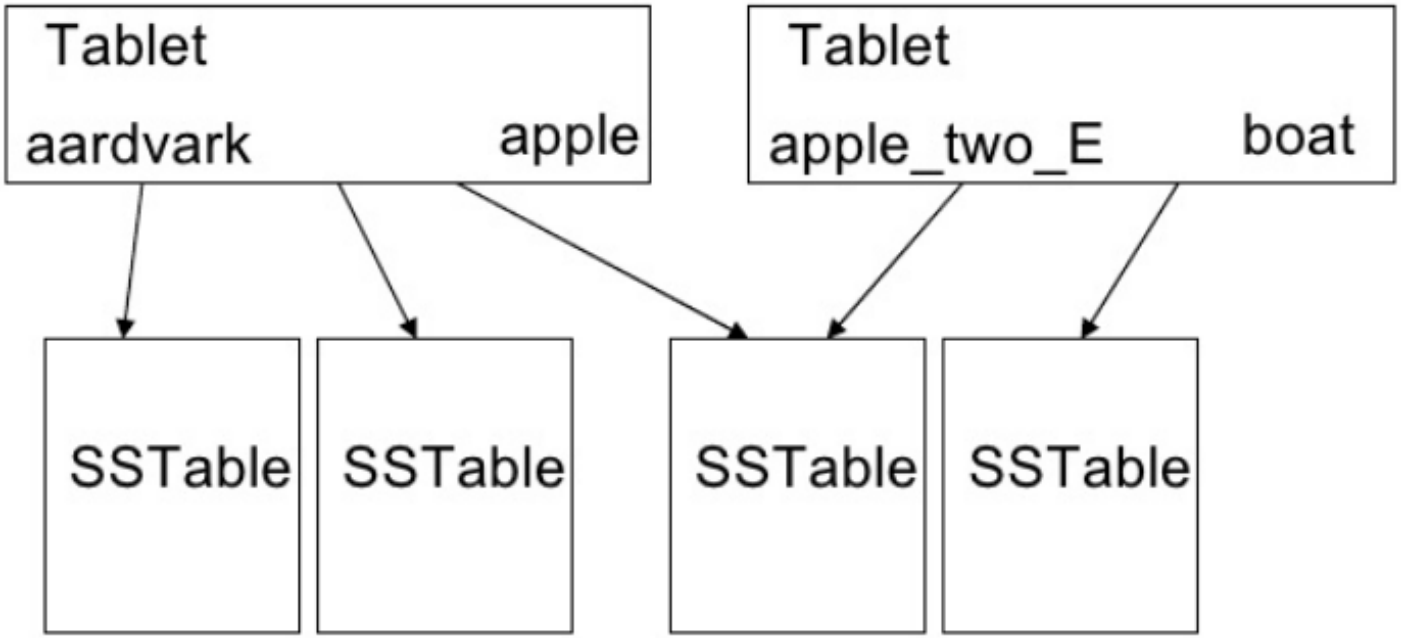
Tablet



Características

- Unidade básica de armazenamento
- Composto por intervalos bem definidos
- Construído sobre SSTables
- Gerenciado por *Tablet server*

Tabela

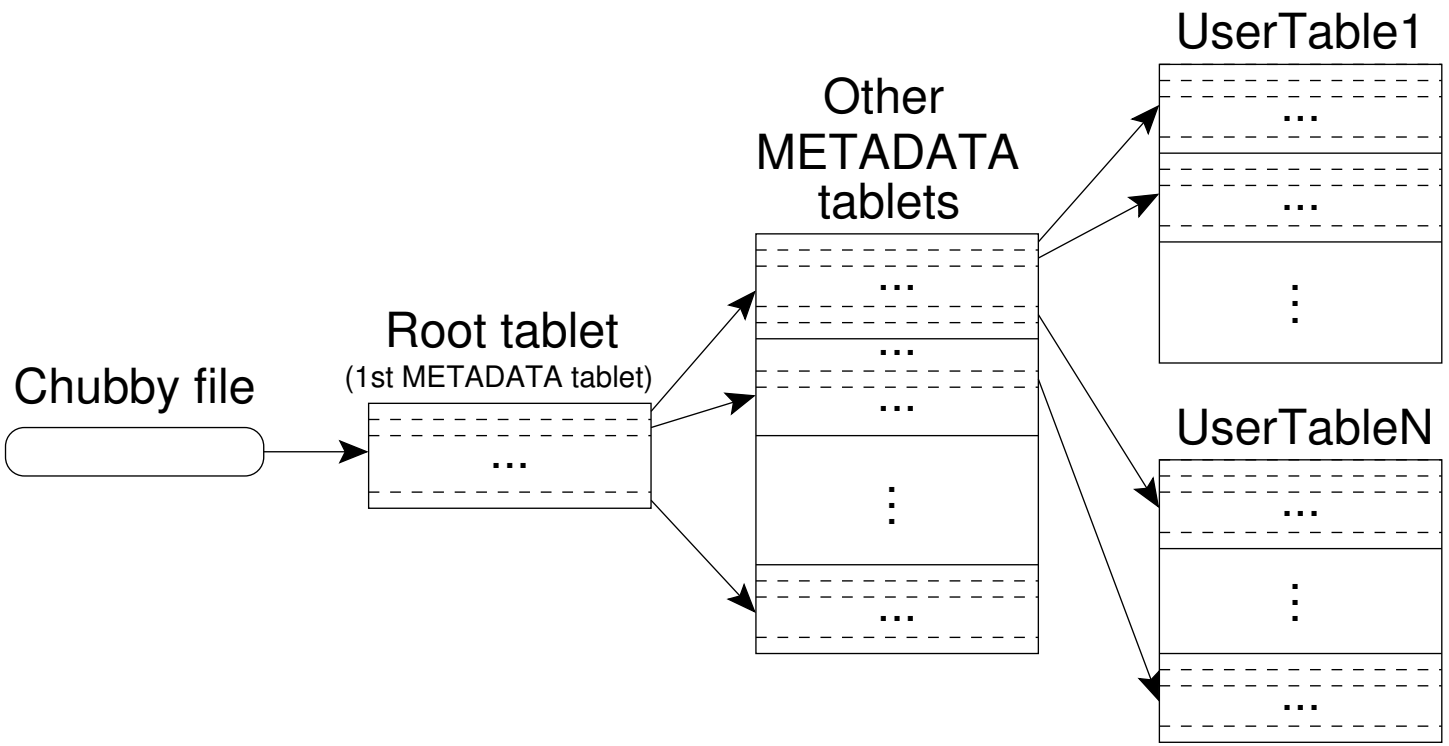

Tablet
aardvark
apple
Tablet
apple_two_E
boat
SSTable
SSTable
SSTable
SSTable

Funcionamento

Requisitos

- Cliente carrega lib
- 1 master: responsável por associar tablets a tablet servers. De 10 a 1000. Monitora mudança de esquemas. Balanceamento de carga
- N tablet servers (gerenciamento dinâmico). Clientes se comunicam diretamente com eles.

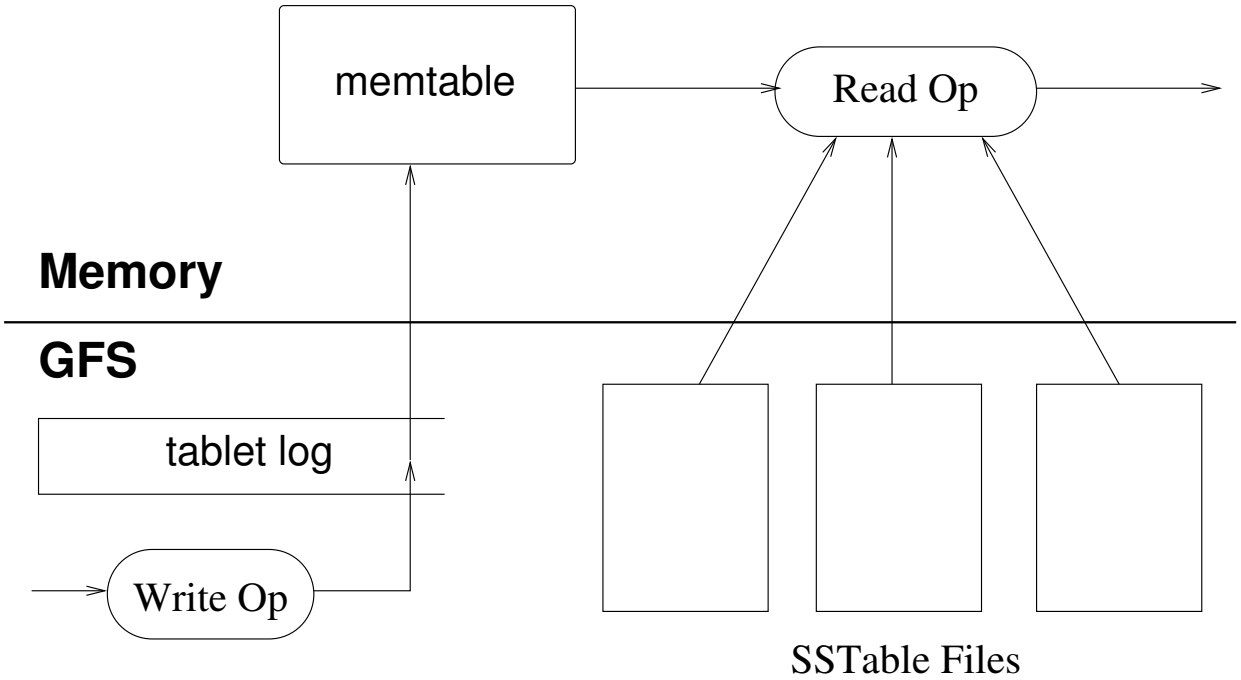
Encontrando tablet



Requisitos

- Semelhante à arvore B+

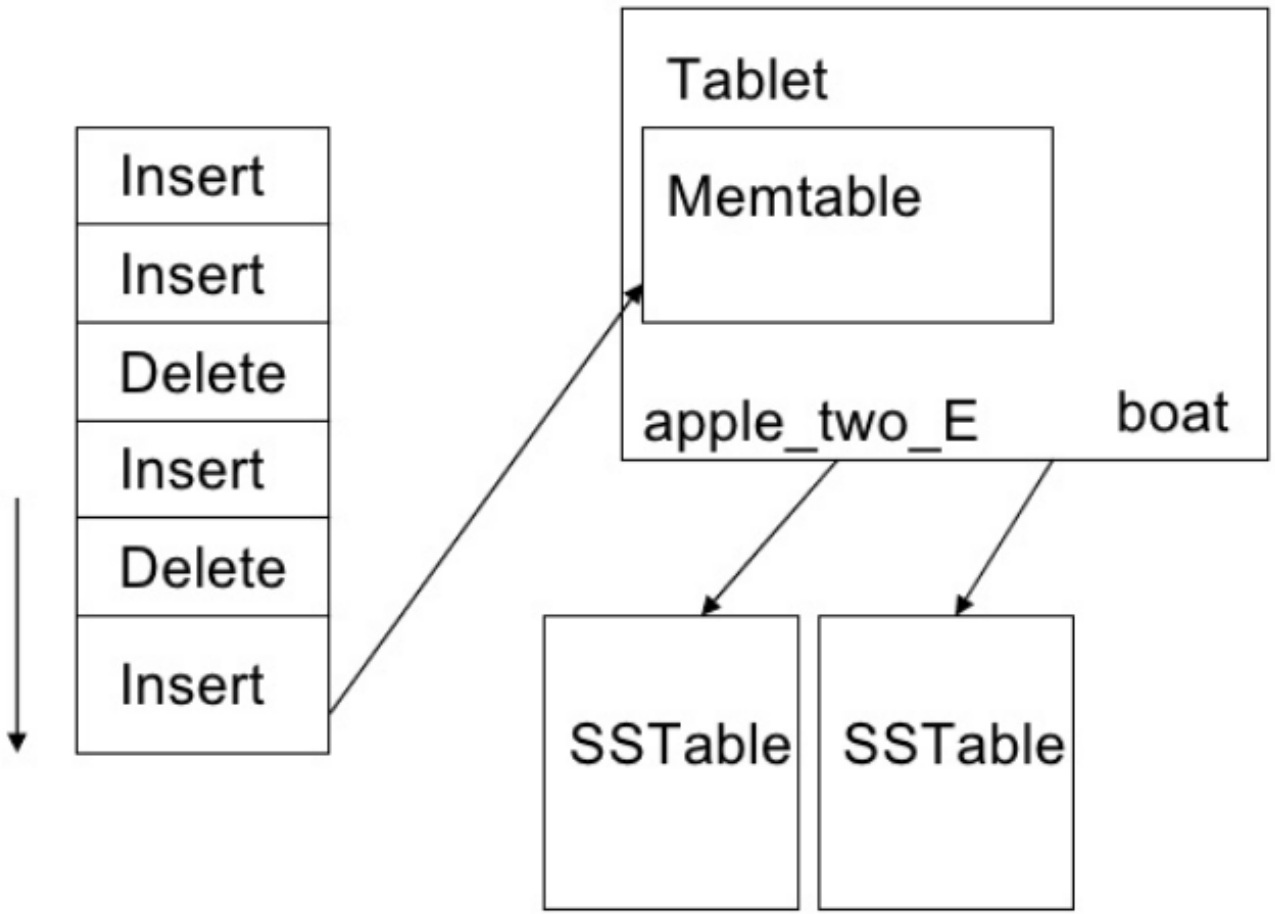
Servindo tablet



Características

- *memtable* é uma cache ordenada
- Recém commits vão para *memtable*
- Compactações: pequenas, junções, grandes

Exemplo


Tablet
Memtable
Insert
Insert
Delete
Insert
Delete
Insert
apple_two_E
boat
SSTable
SSTable

Refinamentos

Grupos de localidade

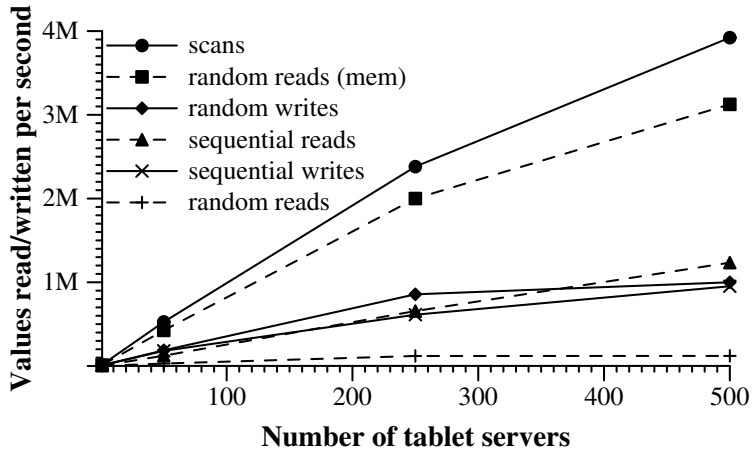
Agrega column families em um mesmo servidor, permitindo estar na memória também

Compressão

Cada SSTable pode ser comprimido

Experimentos

Experiment	# of Tablet Servers			
	1	50	250	500
random reads	1212	593	479	241
random reads (mem)	10811	8511	8000	6250
random writes	8850	3745	3425	2000
sequential reads	4425	2463	2625	2469
sequential writes	8547	3623	2451	1905
scans	15385	10526	9524	7843



Análises

- Escritas melhor devido ao único commit no log
- Leitura sequencial melhor pois se beneficia da localidade espacial
- Scan melhor pois não há RCP
- Escala em um fator de 100

Into the Wild

Project name	Table size (TB)	Compression ratio	# Cells (billions)	# Column Families	# Locality Groups	% in memory	Latency-sensitive?
Crawl	800	11%	1000	16	8	0%	No
Crawl	50	33%	200	2	2	0%	No
Google Analytics	20	29%	10	1	1	0%	Yes
Google Analytics	200	14%	80	1	1	0%	Yes
Google Base	2	31%	10	29	3	15%	Yes
Google Earth	0.5	64%	8	7	2	33%	Yes
Google Earth	70	–	9	8	3	0%	No
Orkut	9	–	0.9	8	5	1%	Yes
Personalized Search	4	47%	6	93	11	5%	Yes

Considerações

- Contempla requisitos de alta disponibilidade, desempenho, escalabilidade e armazenamento
- Satisfatoriamente empregado em diversos produtos Google (mais de 60)
- Justifica a importância de um *design* simples, acima de tudo. Novas funcionalidade apenas quando bem definidas e anteriores funcionando corretamente.