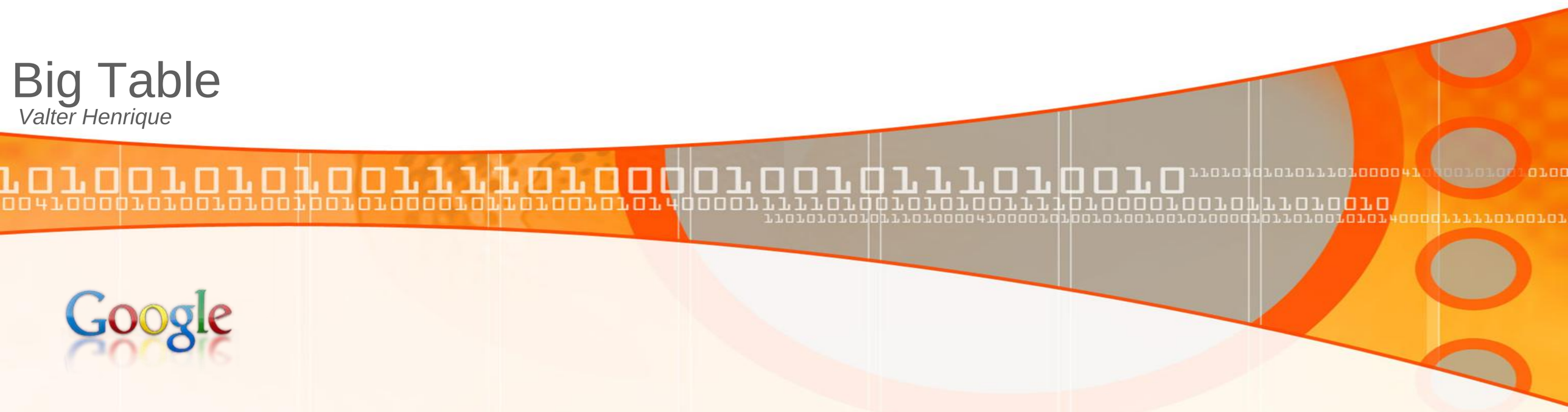
Big Table

Valter Henrique

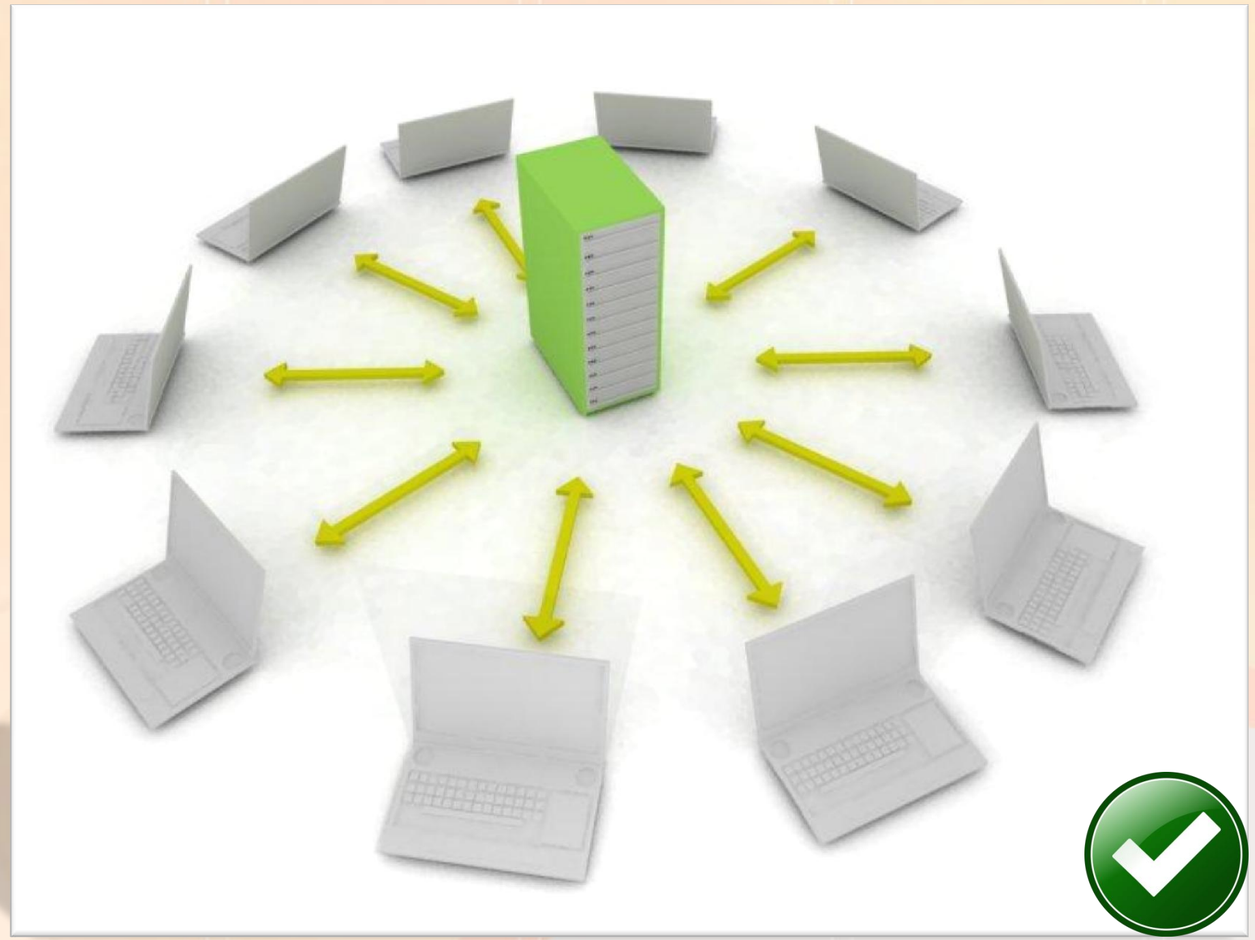


Google

O que é BigTable ?

BigTable

Sistema
de armazenamento
de dados
estruturados



Motivação

- Muitas (semi-)estruturas de dados no mundo e no Google
  - URL
    - Conteúdo, rastreamento de metadados, links, âncoras, pagerank
  - Dados por usuário
    - Preferências dos usuário, queries recentes, resultados de busca
  - Localizações Geográficas
    - Entidades físicas (lojas, restaurantes, etc)
    - Estradas, satélites, imagens, anotações dos usuários, ..
  - Larga escala
    - Bilhões de URL's, muitas versões/páginas (+- 20k / versão)
    - Centenas de milhões de usuários
    - Milhares de queries por segundo
    - Mais de 100TB de imagens de satélite

Por que não usar um BD comercial ?

- A escala é muito grande para os BD’s comerciais

- Mesmo se não fosse, o custo seria muito grande
  - Construir internamente significa que o sistema possa ser aplicado em muitos projetos por baixo custo incremental

- Otimizações de baixo-nível ajudam na perfomance significativamente
  - Muito mais difícil fazer quando se esta no topo da camada do banco de dados, já que não se sabe muito bem o que esta acontecendo por baixo

Objetivos

- *Processos devem ser assíncronos para atualizarem diferentes diferentes partes dos dados*
  - *Querer acessar o dado mais atual a qualquer momento*

- *Precisa suportar*
  - *Altas taxas de leitura/escrita (milhões de operações por segundo)*
  - *Examinar todos ou subconjuntos de dados interessantes*
  - *Joins eficientes, one-to-one, one-to-many*

- *Frequentemente analisar mudanças dos dados o tempo todo*
  - *Conteúdo de uma página web que possui muito rastreadores*

Big Table

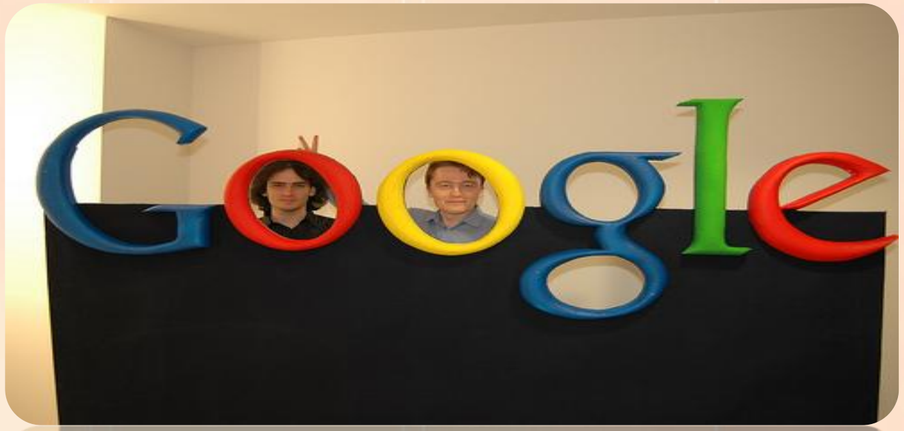
- Distribuída em multi-camadas
  - Interesse em modelagem dos dados

- Tolerância a falha
- Persistente
- Escalável
  - Milhares de servidores
  - Terabytes de dados em memória
  - Petabytes de dados em disco
  - Milhões de leitura/escritas por segundo
  - Buscas eficientes

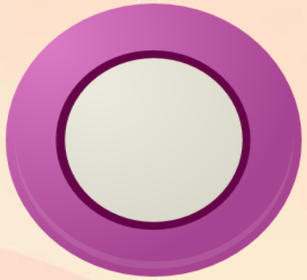
- Auto-gerenciável
  - Servidores podem ser adicionados/removidos dinamicamente
  - Servidores ajustam o balanceamento de carga

Status

- *Implementação projeto começou em 2004*
- *Atualmente por volta de 100 células da Big Table*
- *Uso em produção ou desenvolvimento ativo em vários projetos:*
  - *Rastreamento*
  - *Pipeline de indexação*



Status - Projetos



Construindo de bloco em bloco

- *GFS: Google File System*
  - *Armazenamento bruto*

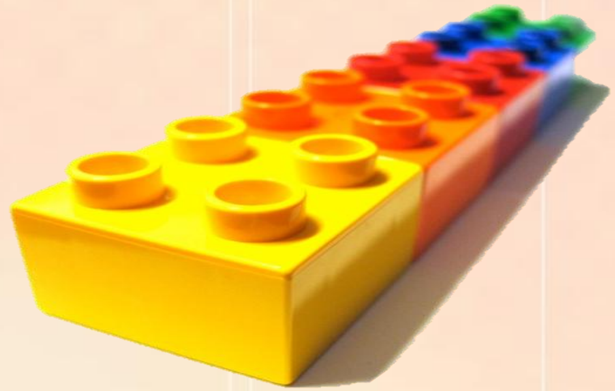
- *Scheduler*
  - *Permitir rodar diferentes tarefas em diferentes máquinas*
  - *Responsável por gerenciar as máquinas e realocá-las se necessário*

- *Serviço de bloqueio*
  - *Gerenciador de bloqueio distribuído*
  - *Pode manter arquivos pequenos de forma confiável com alta disponibilidade*
  - *Servidores podem se registrar por armazenar arquivos no sistema assim outros servidores podem ler quais se registraram Sabendo quem trabalha e quem esta dísponivel*

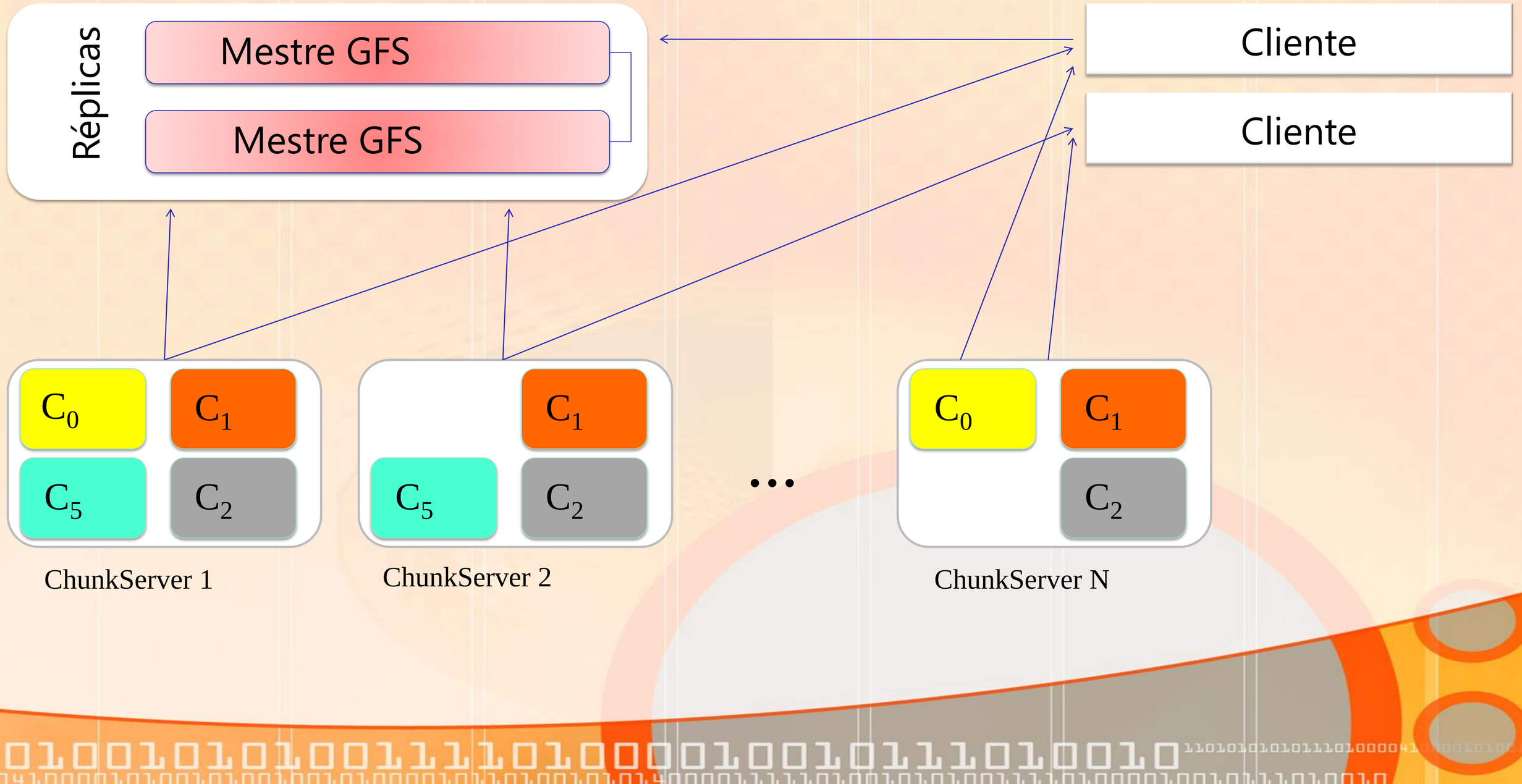
Construindo de bloco em bloco

- *MapReduce*
  - simplifica o processamento em larga-escala

- Big Table utiliza:
  - GFS: armazena o estado persistente
  - Scheduler : rodar tarefas em várias máquinas e realocar
  - Serviço de bloqueio: eleição do mestre, boostraping
  - MapReduce: frequentemente usado para ler/escrever dados na Big Table



Google File System



Google File System

- *Mestre gerencia os metadados*
- *Transferência dos dados acontece diretamente entre clientes / chunkservers*
- *Arquivos são quebrados em partes (chunk), geralmente 64MB*
- *Partes são triplicadas em três máquinas por segurança*

MapReduce : Ciclos fáceis de usar

- *Muitos dos problemas do Google*
  - *“Processar dados que irão gerar outros dados”*

- *Muitos tipos de entradas*
  - *Registros de documentos*
  - *Arquivos de log*
  - *Organizar estrutura de dados no disco*
  - *…*

  - *Usar facilmente centenas, milhares de CPU’s*

MapReduce : Ciclos fáceis de usar

- *MapReduce é um framework que fornece:*
  - *Distribuição / Paralelização automática e eficiente*
  - *Tolerância a falha*
  - *Agendamento de I/O*
  - *Estado / Monitoramento*
  - *Usuário usam funções Map e Reduce*
  - *Altamente usado : +- 3000 tarefas, milhares de máquinas que utilizam MapReduce*

- *MapReduce : “Processamento de dados simplificado nos grandes clusters”*

- *BigTable pode ser entrada e/ou saída para o processamento do MapReduce*

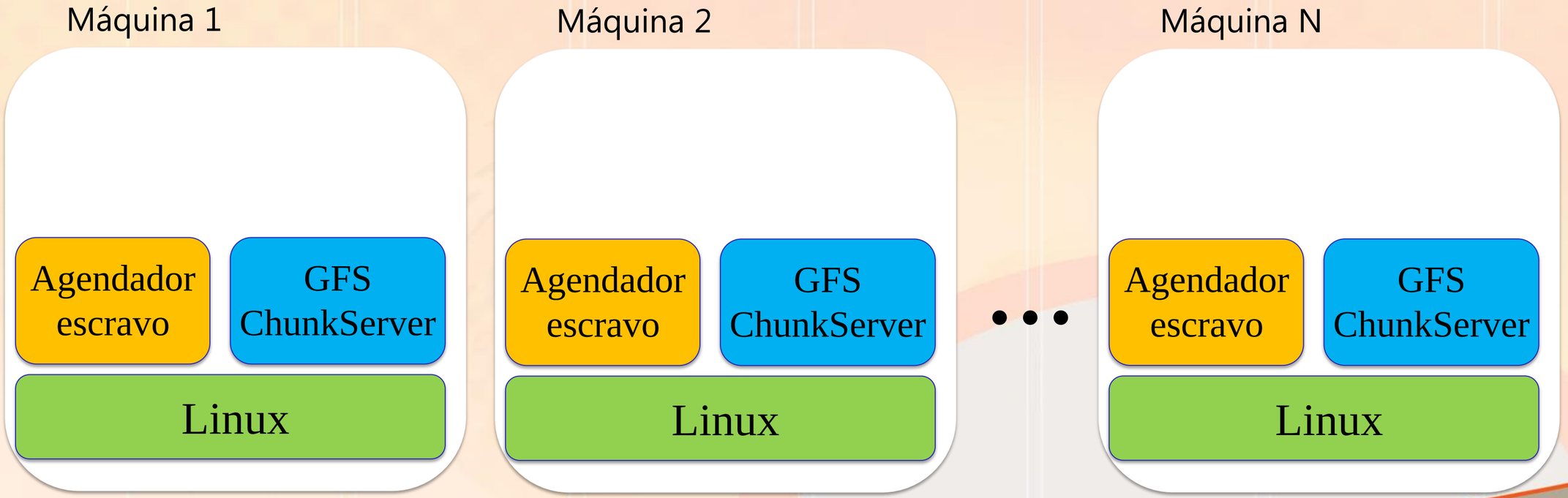
Cluster típico

Google

Mestre Agendamento Cluster

Serviço de bloqueio

Mestre GFS



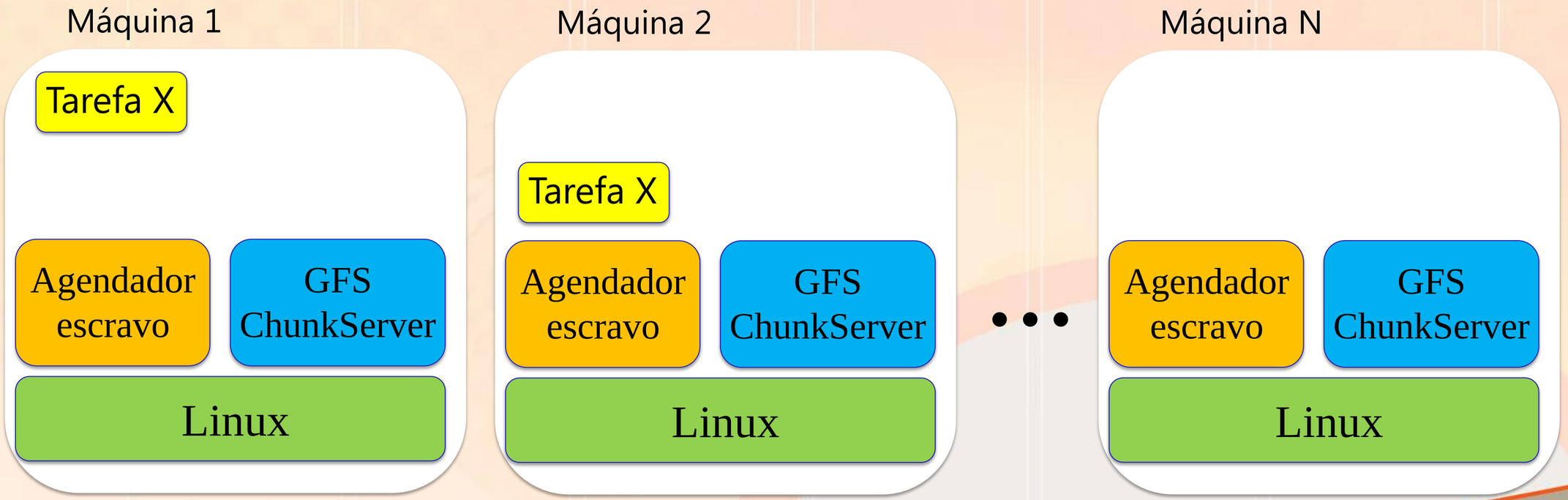
Cluster típico

Google

Mestre Agendamento Cluster

Serviço de bloqueio

Mestre GFS



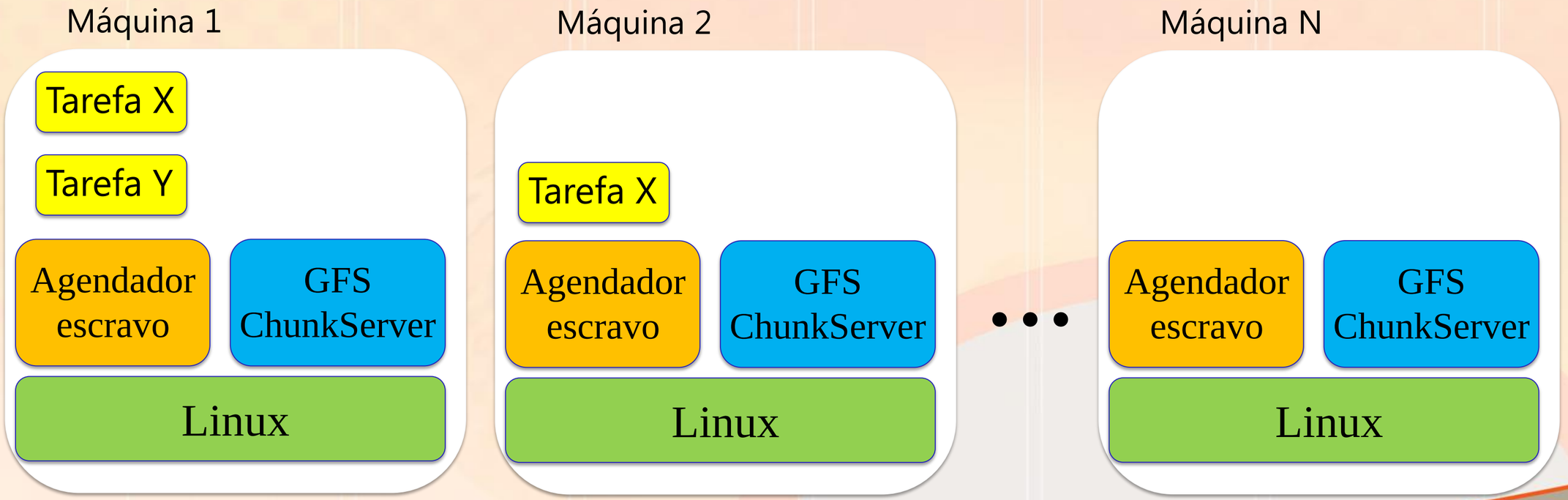
Cluster típico

Google

Mestre Agendamento Cluster

Serviço de bloqueio

Mestre GFS



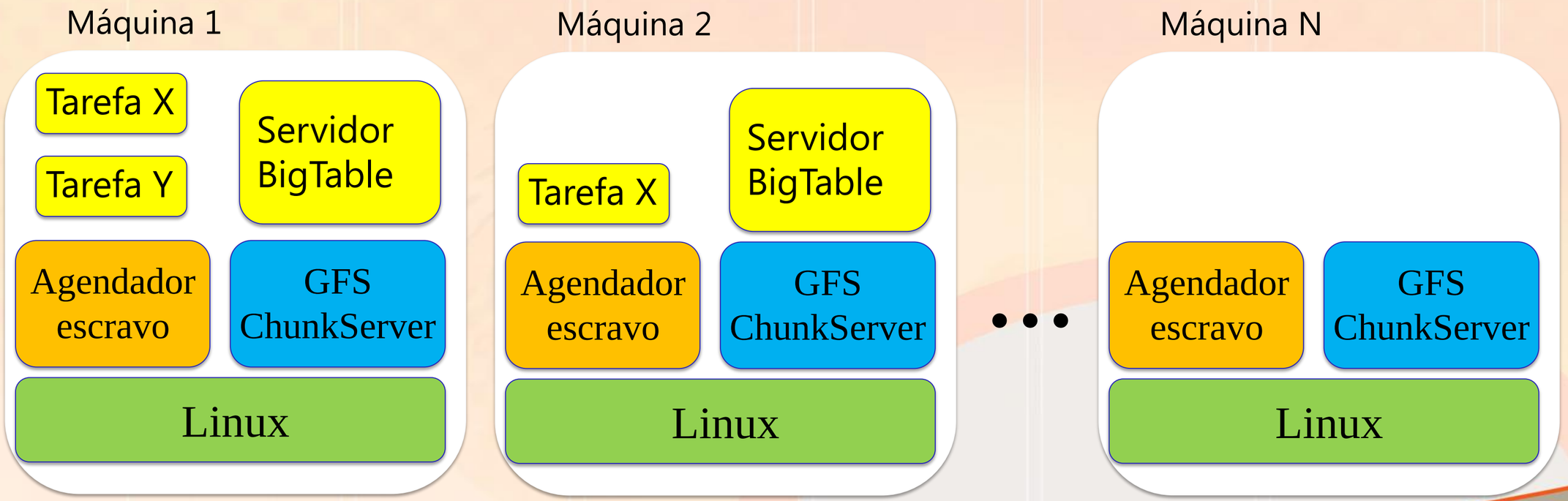
Cluster típico

Google

Mestre Agendamento Cluster

Serviço de bloqueio

Mestre GFS



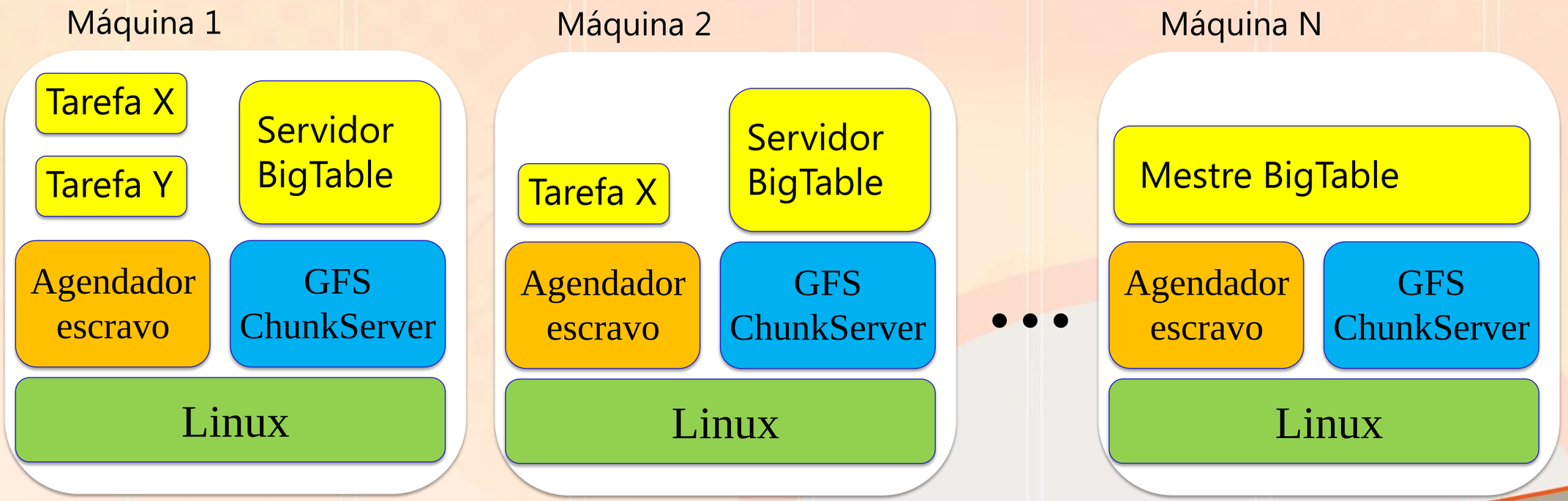
Cluster típico

Google

Mestre Agendamento Cluster

Serviço de bloqueio

Mestre GFS

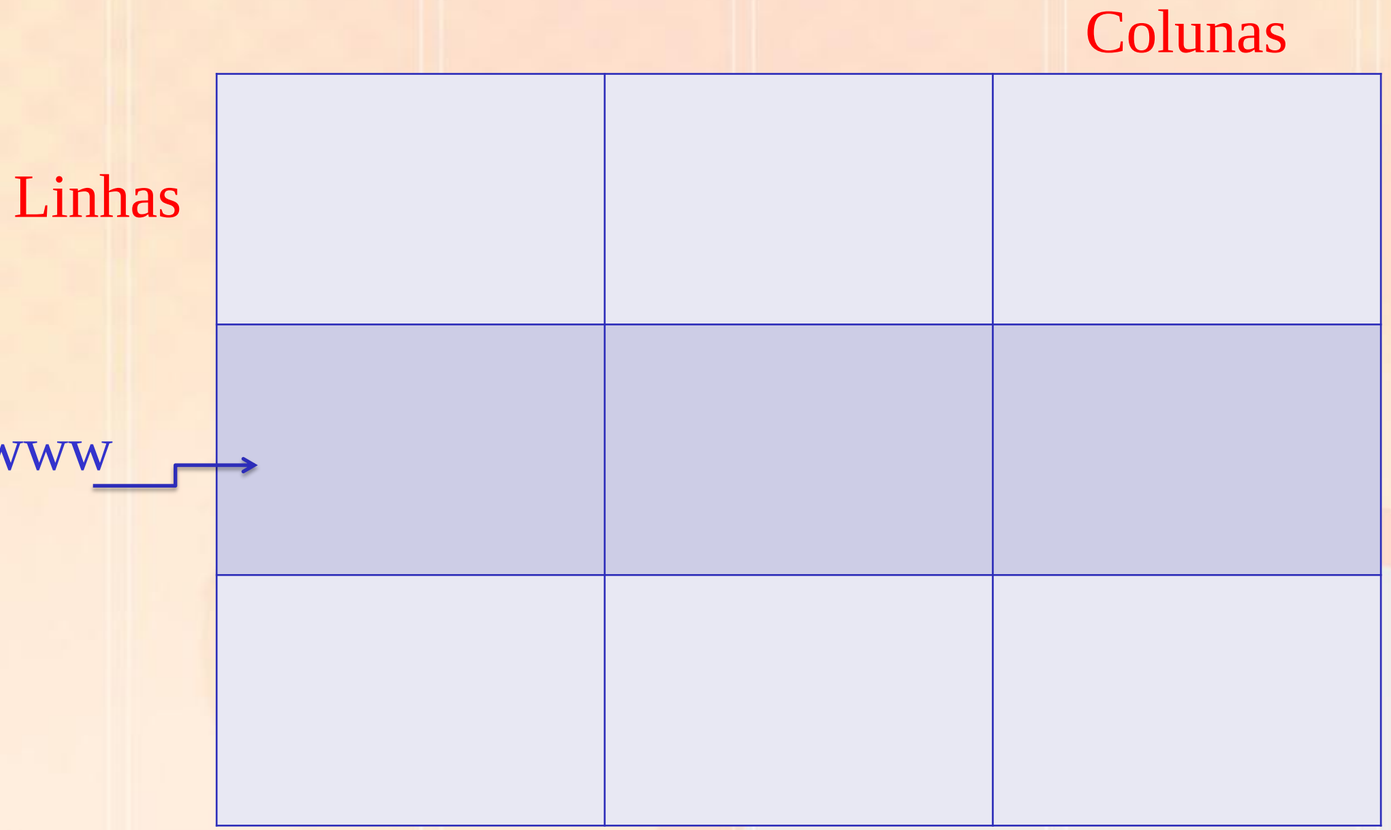


Visão Geral BigTable

- *Modelos dos dados*
- *Tablets*
- *Compactadores*
- *Grupos locais*
- *API*
- *Detalhes*
  - *Logs compartilhados, compressão, replicação, …*
- *Atualmente e futuro*

Modelo Básico de Dados

- *Mapeamento multi-dimensional esparso*
  - *(linhas, coluna, timestamp)* → *Conteúdo das células*

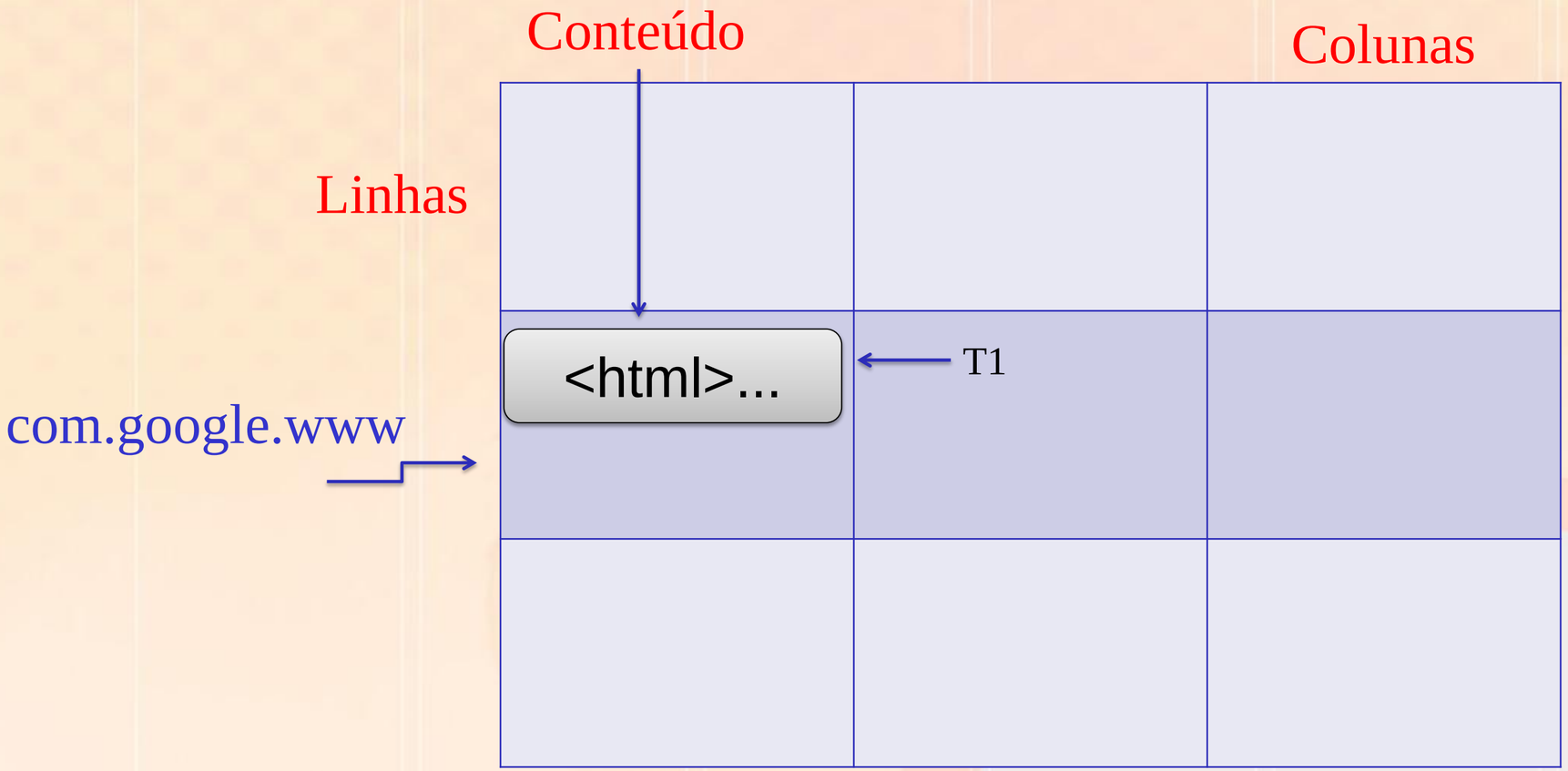


Modelo Básico de Dados

- *Mapeamento multi-dimensional esparso*
  - *(linhas, coluna, timestamp)* → *Conteúdo das células*

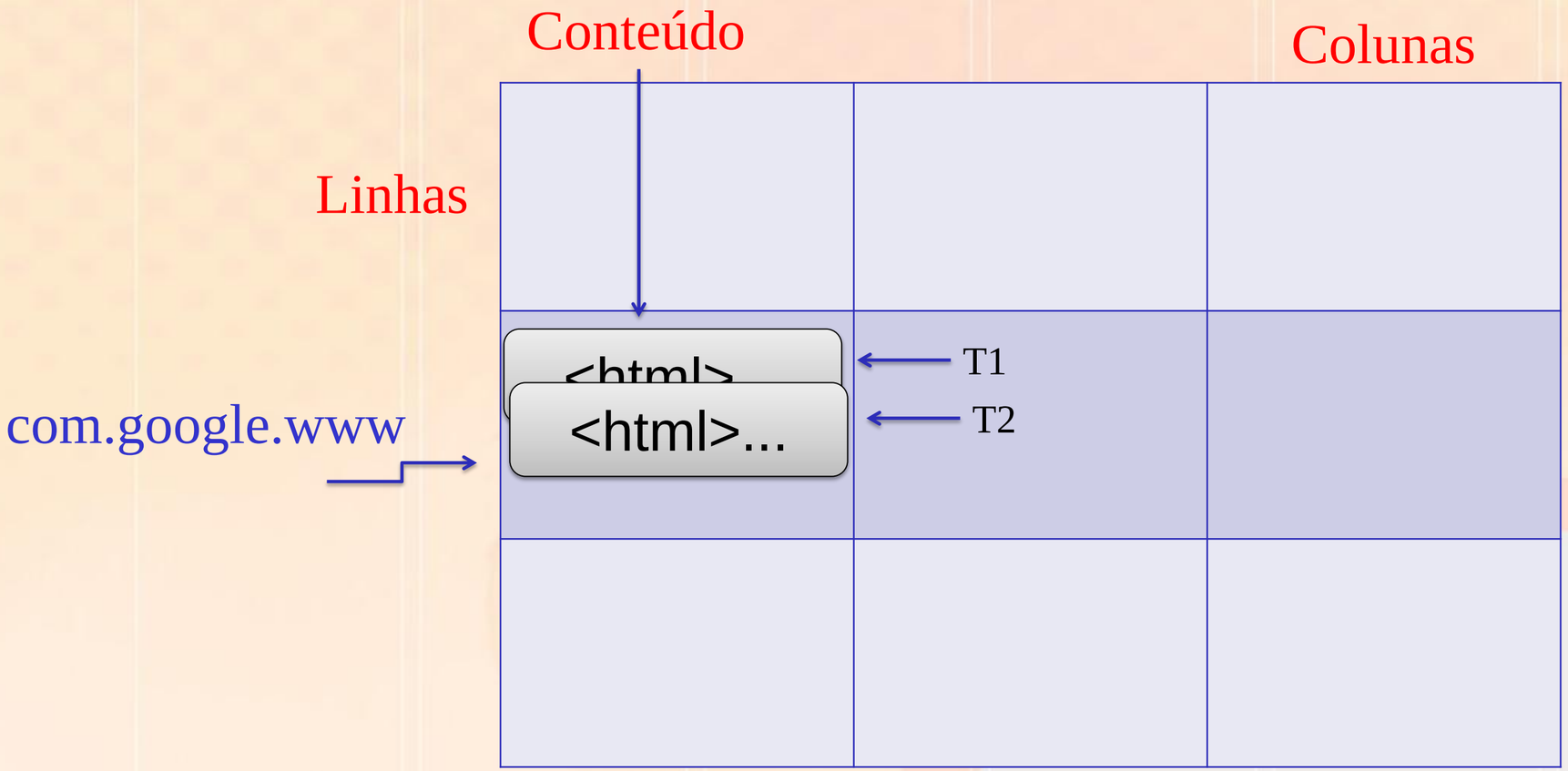
Modelo Básico de Dados

- *Mapeamento multi-dimensional esparso*
  - *(linhas, coluna, timestamp)* ⟶ *Conteúdo das células*



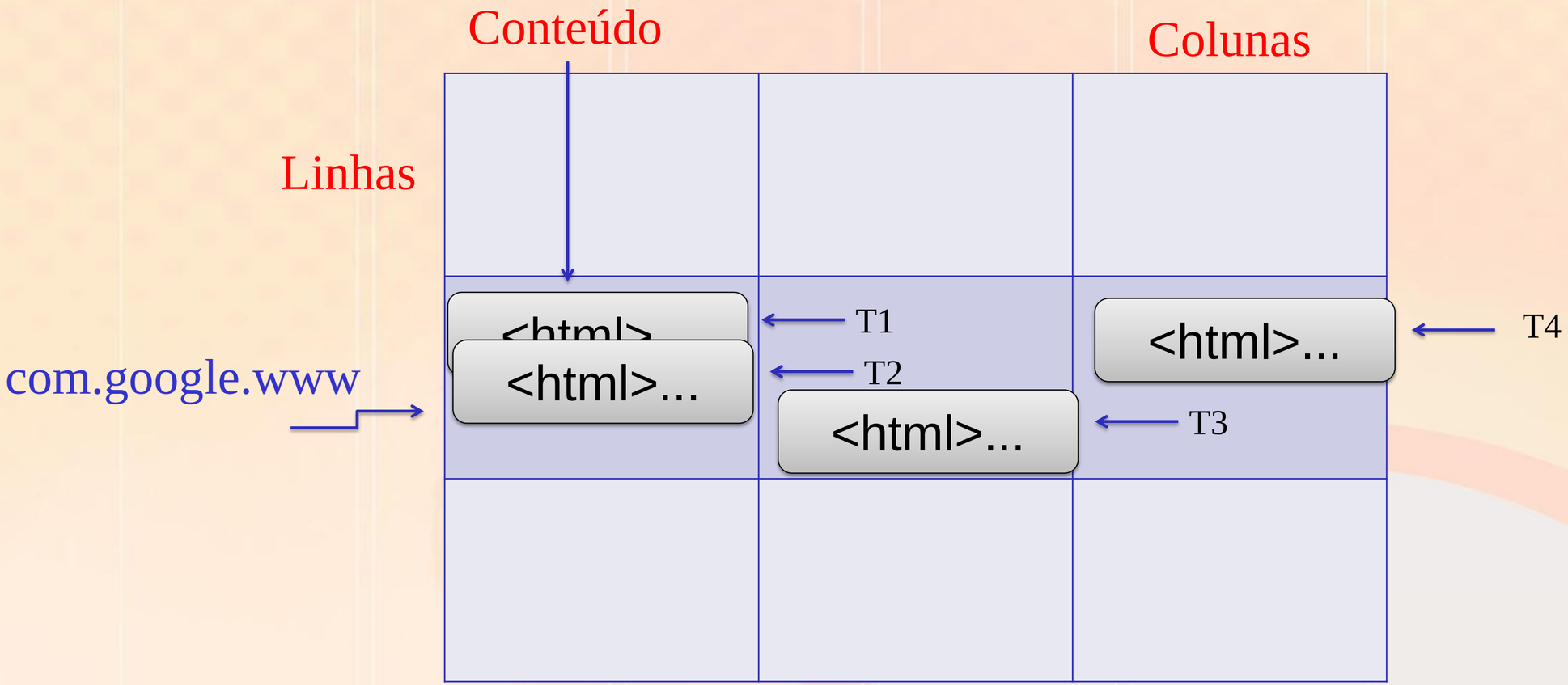
Modelo Básico de Dados

- *Mapeamento multi-dimensional esparso*
  - *(linhas, coluna, timestamp)* → *Conteúdo das células*



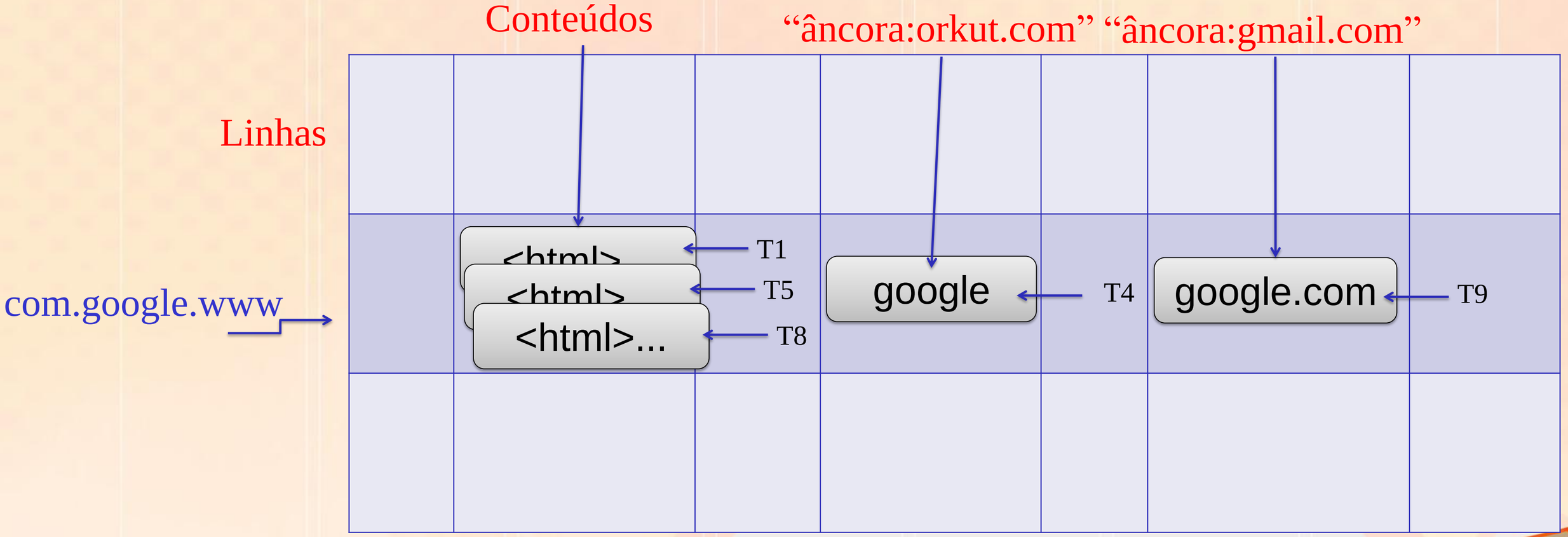
Modelo Básico de Dados

- *Mapeamento multi-dimensional esparso*
  - *(linhas, coluna, timestamp)* ⟶ *Conteúdo das células*



Modelo Básico de Dados

- *Mapeamento multi-dimensional esparso*
  - *(linhas, coluna, timestamp)* *Conteúdo das células*

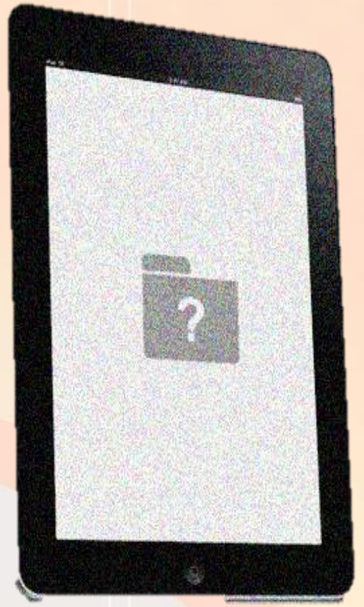


Linhas

- Nome da linha é uma string arbitrária
- Acesso a um dado em uma linha é atômico
- A criação de uma linha é implicita aos dados armazenados acima
- Linhas são ordenadas lexicograficamente
  - Linhas que são lexicograficamente próximas usualmente ficam em uma ou em um número menor de máquina, para desempenho maior

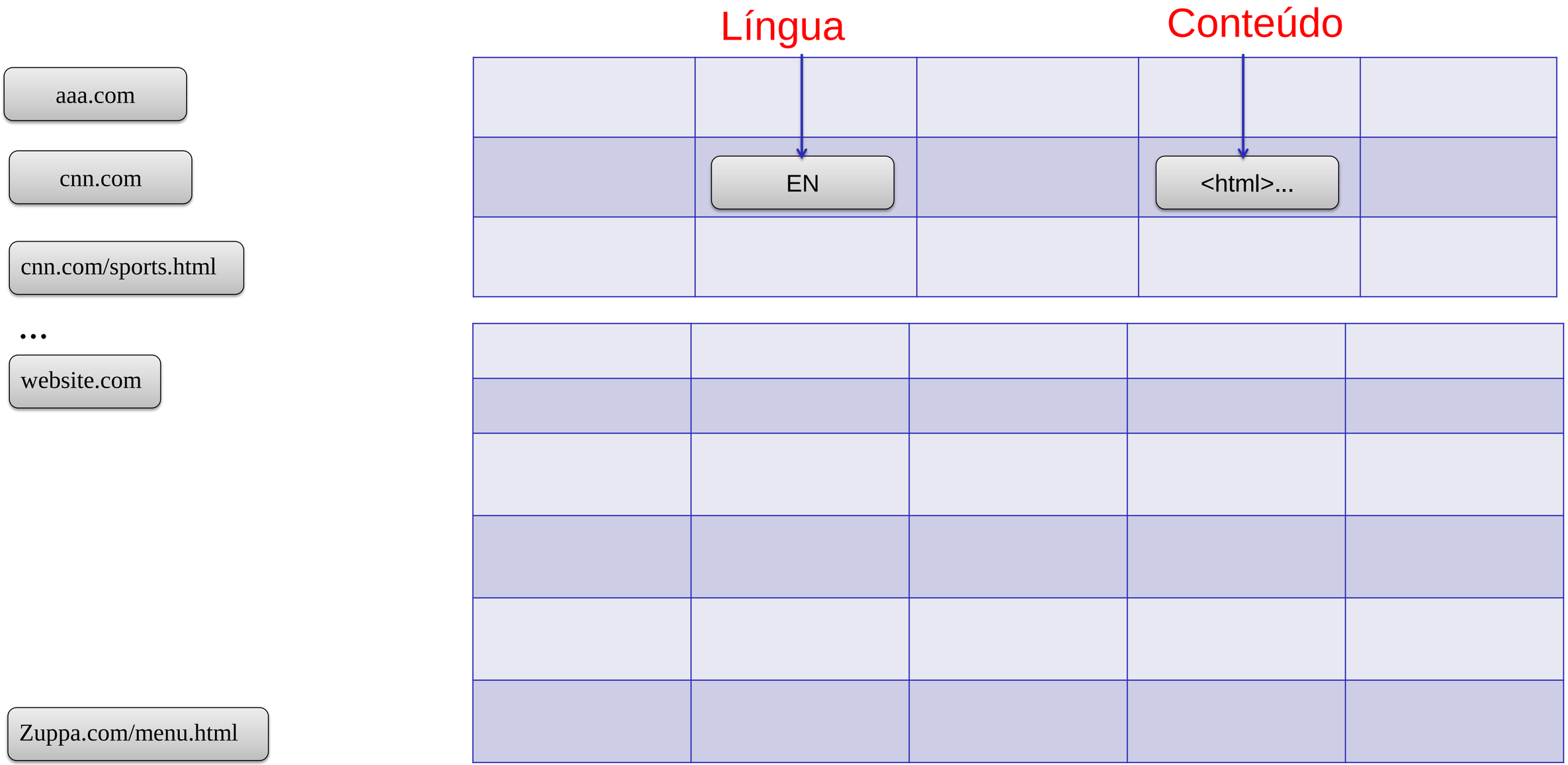
Tablets

- Tabelas grandes são quebradas em *tablets*
  - *Tablets* mantém um alcance contínuo de linhas
    - *Clientes podem frequentemente escolher a chave da linha para obter a localização*
- *As tablets podem possuir de 100 MB a 200 MB*
- *Máquinas servidoras são responsáveis por +- 100 tablets*
  - *Recuperação rápida dos dados*
  - *100 máquinas, cada uma pega 1 tablet de uma máquina que falhou*
- *Balanceamento de carga refinado*
  - *Migração de tablets para longe de máquina que esta sobrecarregada*
  - *Mestre toma as decisões de balanceamento de carga*



Tablets e Divisões

Google



Tablets e Divisões

Google

aaa.com

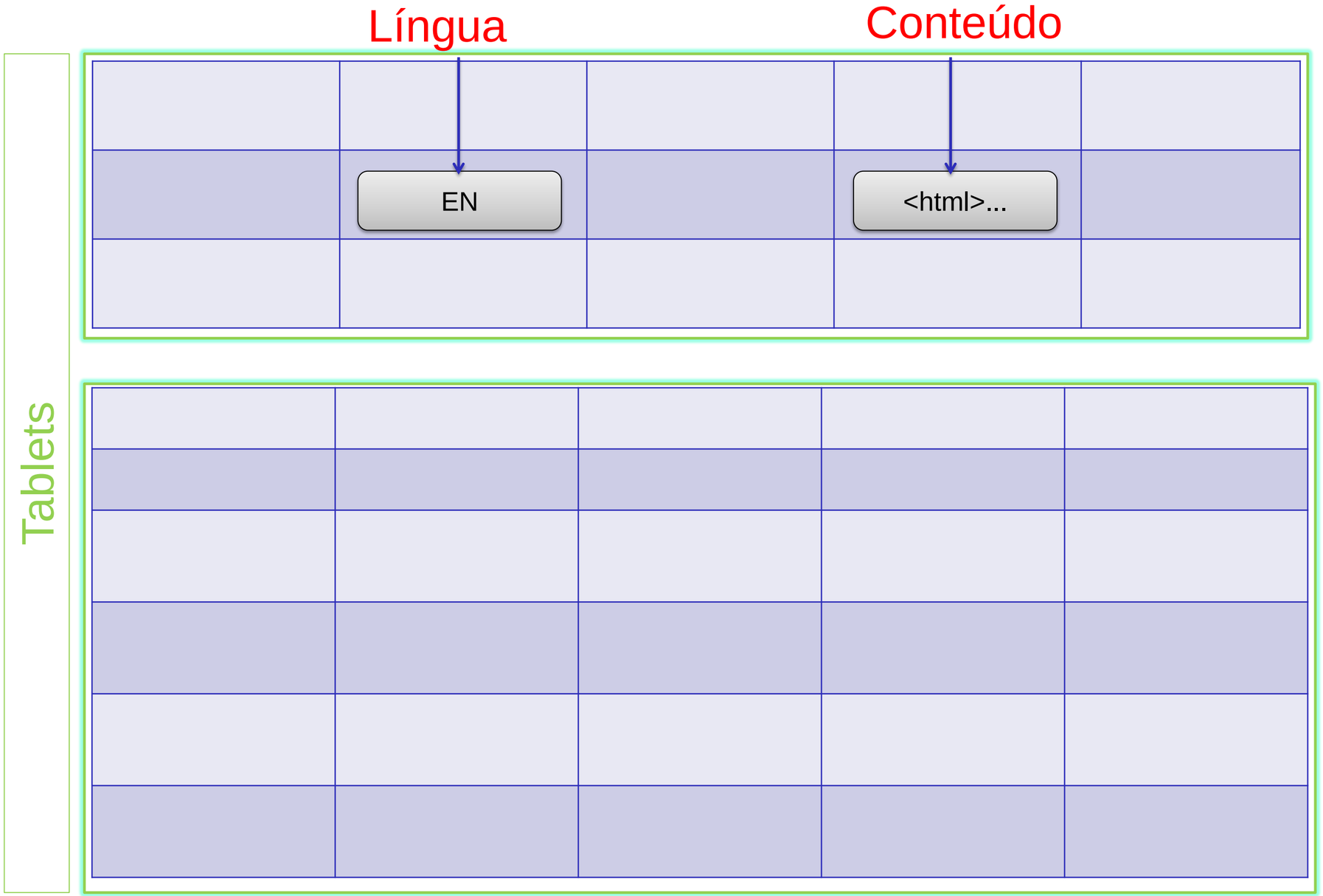
cnn.com

cnn.com/sports.html

...

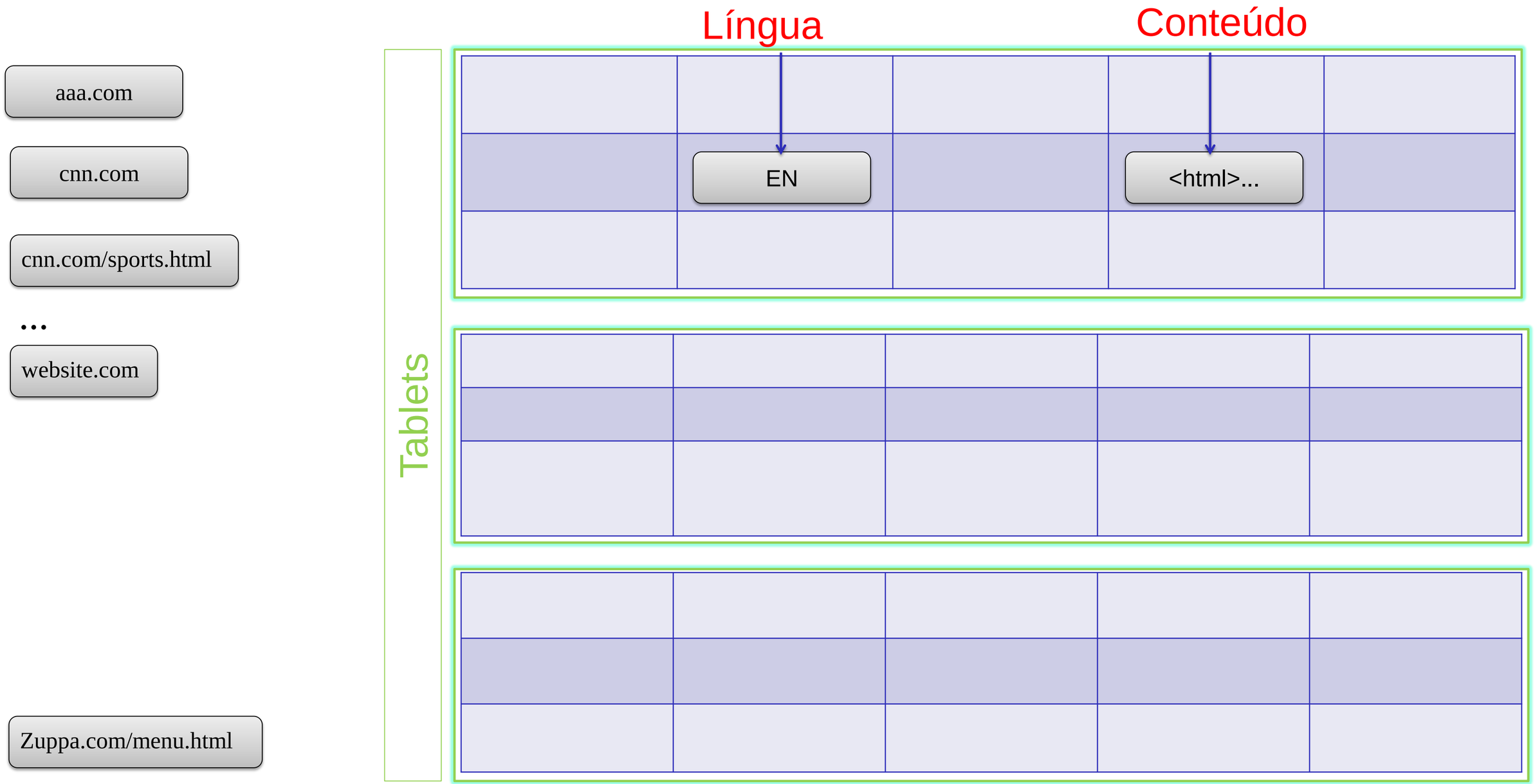
website.com

Zuppa.com/menu.html



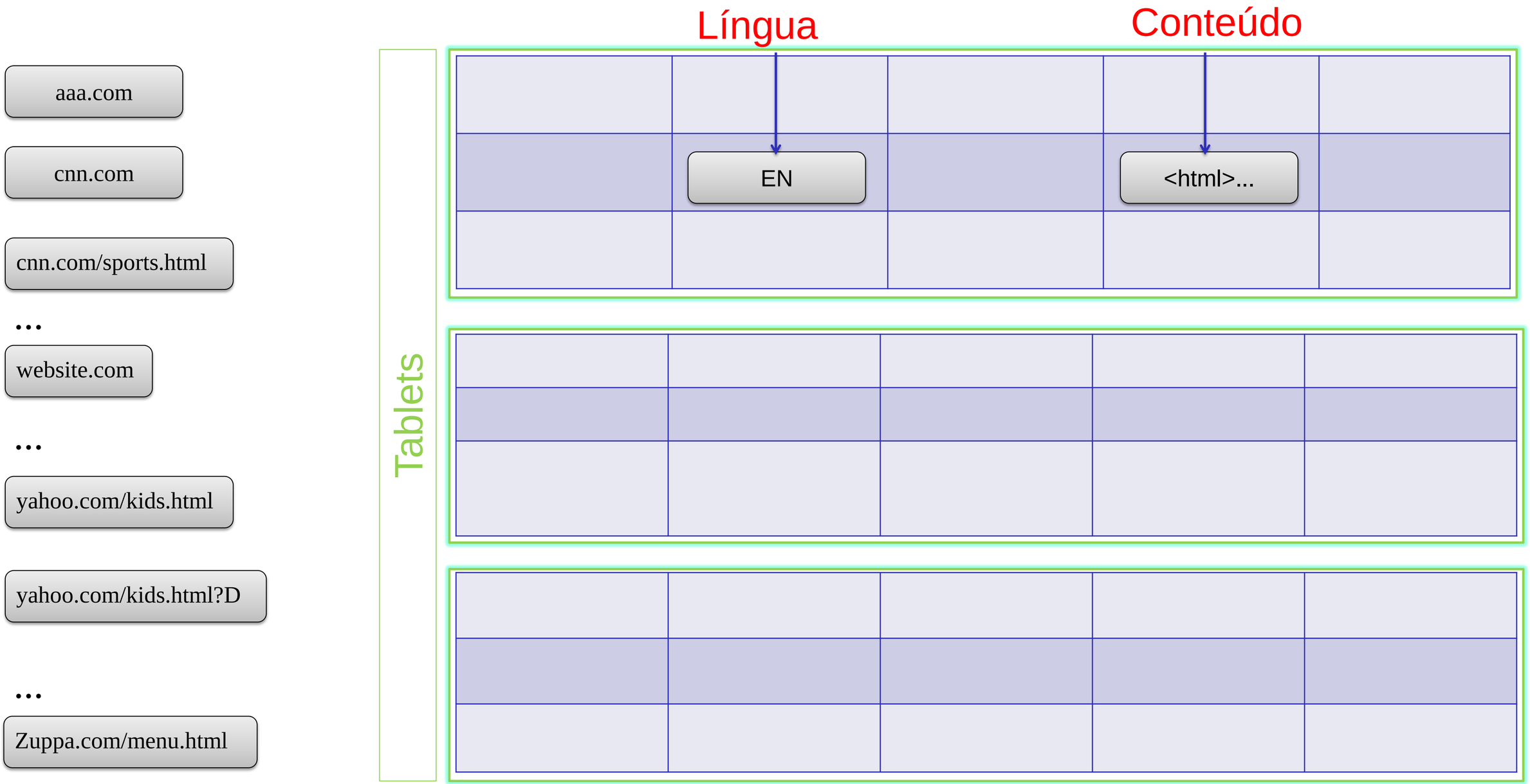
Tablets e Divisões

Google



Tablets e Divisões

Google



Tablet Serving Structure

Google

Célula BigTable

Mestre BigTable

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidor BigTable Tablet

Tablet Serving Structure

Célula BigTable

Mestre BigTable

Operações de metadados
Balanceamento de carga

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidor BigTable Tablet

Tablet Serving Structure

Célula BigTable

Mestre BigTable

Operações de metadados
Balanceamento de carga

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Tablet Serving Structure

Tablet Serving Structure

Célula BigTable

Mestre BigTable

Operações de metadados
Balanceamento de carga

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Mestre de agendamento de cluster

Google File System

Serviço de bloqueio

Gerenciar falhas
Monitoramento

Tablet Serving Structure

Célula BigTable

Mestre BigTable

Operações de metadados
Balanceamento de carga

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Mestre de agendamento de cluster

Google File System

Serviço de bloqueio

Gerenciar falhas
Monitoramento

Detém dados do tablet
Logs

Tablet Serving Structure

Célula BigTable

Mestre BigTable

Operações de metadados
Balanceamento de carga

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Mestre de agendamento de cluster

Google File System

Serviço de bloqueio

Gerenciar falhas
Monitoramento

Detém dados do tablet
Logs

Detém metadados
Gerencia eleição do mestre

Tablet Serving Structure

Google

Vários mestres - Só o mestre eleito é ativado em qualquer momento e os demais aguardam serem eleitos

Célula BigTable

Mestre BigTable

Operações de metadados
Balanceamento de carga

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Mestre de agendamento de cluster

Google File System

Serviço de bloqueio

Gerenciar falhas
Monitoramento

Detém dados do tablet
Logs

Detém metadados
Gerencia eleição do mestre

Tablet Serving Structure

Google

Vários mestres - Só o mestre eleito é ativado em qualquer momento e os demais aguardam serem eleitos

Célula BigTable

Mestre BigTable

Operações de metadados
Balanceamento de carga

Cliente BigTable

Biblioteca de cliente BigTable

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Mestre de agendamento de cluster

Google File System

Serviço de bloqueio

Gerenciar falhas
Monitoramento

Detém dados do tablet
Logs

Detém metadados
Gerencia eleição do mestre

Tablet Serving Structure

Google

Vários mestres - Só o mestre eleito é ativado em qualquer momento e os demais aguardam serem eleitos

Célula BigTable

Mestre BigTable

Operações de metadados
Balanceamento de carga

Cliente BigTable

Biblioteca de cliente BigTable

Abrir()

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Mestre de agendamento de cluster

Google File System

Serviço de bloqueio

Gerenciar falhas
Monitoramento

Detém dados do tablet
Logs

Detém metadados
Gerencia eleição do mestre

Tablet Serving Structure

Vários mestres - Só o mestre eleito é ativado em qualquer momento e os demais aguardam serem eleitos

Célula BigTable

Mestre BigTable

Operações de metadados
Balanceamento de carga

Cliente BigTable

Biblioteca de cliente BigTable

Abrir()

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidor BigTable Tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Servidores de dados
Ler/Escrever, dividir tablet

Mestre de agendamento de cluster

Google File System

Serviço de bloqueio

Gerenciar falhas
Monitoramento

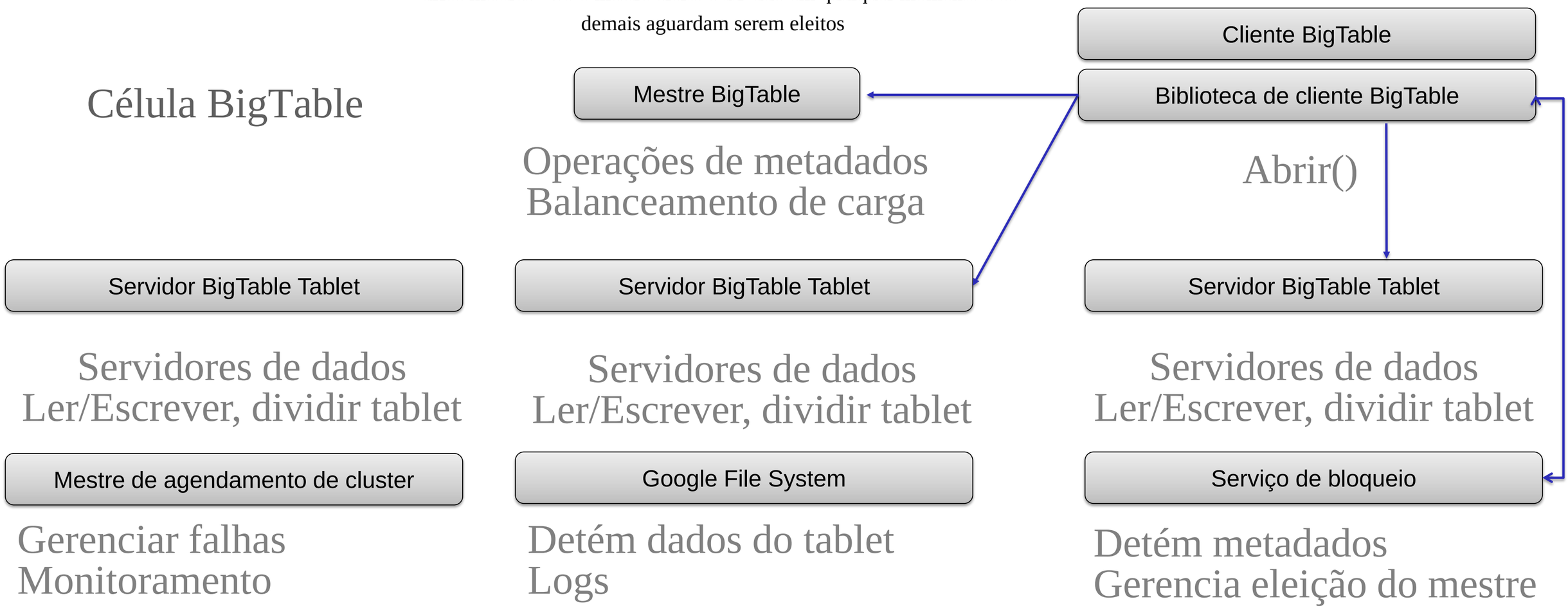
Detém dados do tablet
Logs

Detém metadados
Gerencia eleição do mestre

Tablet Serving Structure

Vários mestres - Só o mestre eleito é ativado em qualquer momento e os demais aguardam serem eleitos

Célula BigTable



Localização dos Tables

- *Desde que tablets se movimentam pelos servidores, dado uma linha,*
- *como encontrar a máquina correta que possui o dado ?*
  - *É necessário encontrar o tablet que possui o intervalo solicitado*

- *Uma alternativa : utilizar a BigTable mestre*

  Servidor central que quase certamente seria um gargalo em um grande sistema

  Outra alternativa: armazenar tabela especiais que contenham a localização

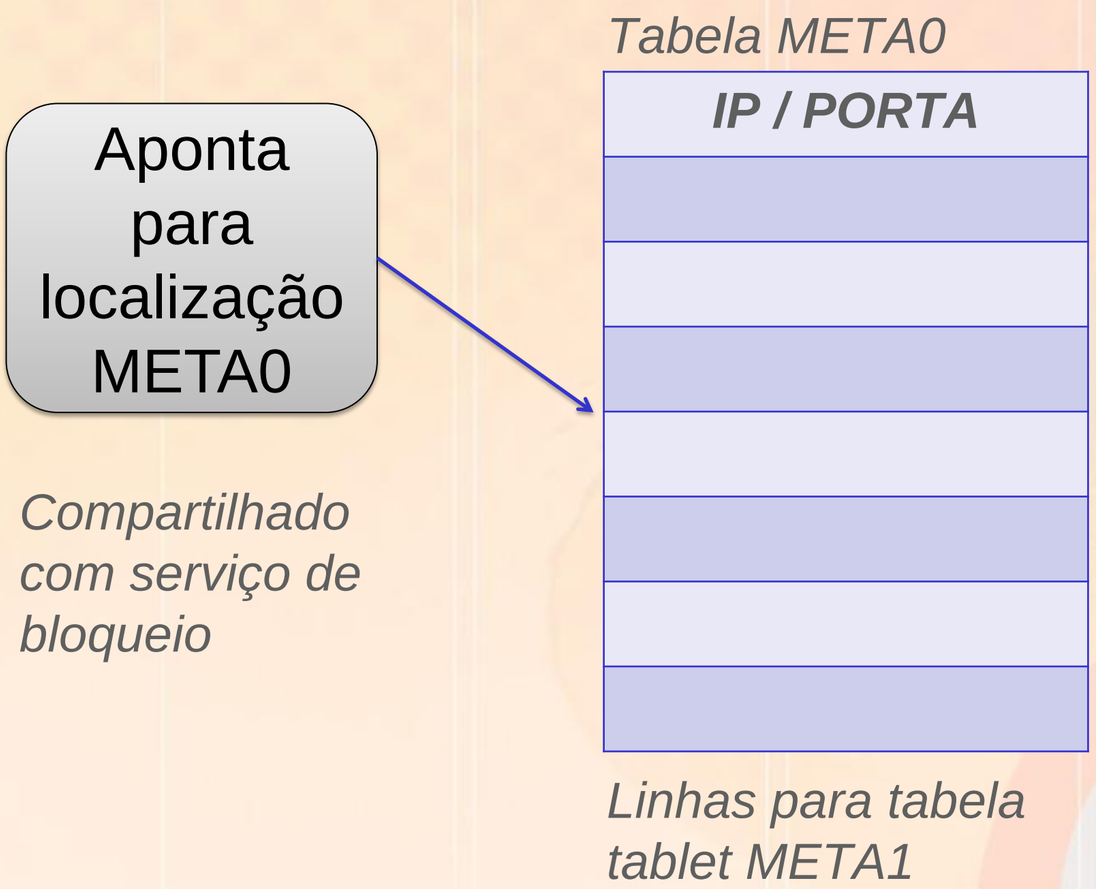
  Própria célula da BigTable

Localização Tablets

- *A abordagem possui 3 níveis de hierárquia para pesquisar por tabtlets*
- *A localização é ip, porta de um servidor*

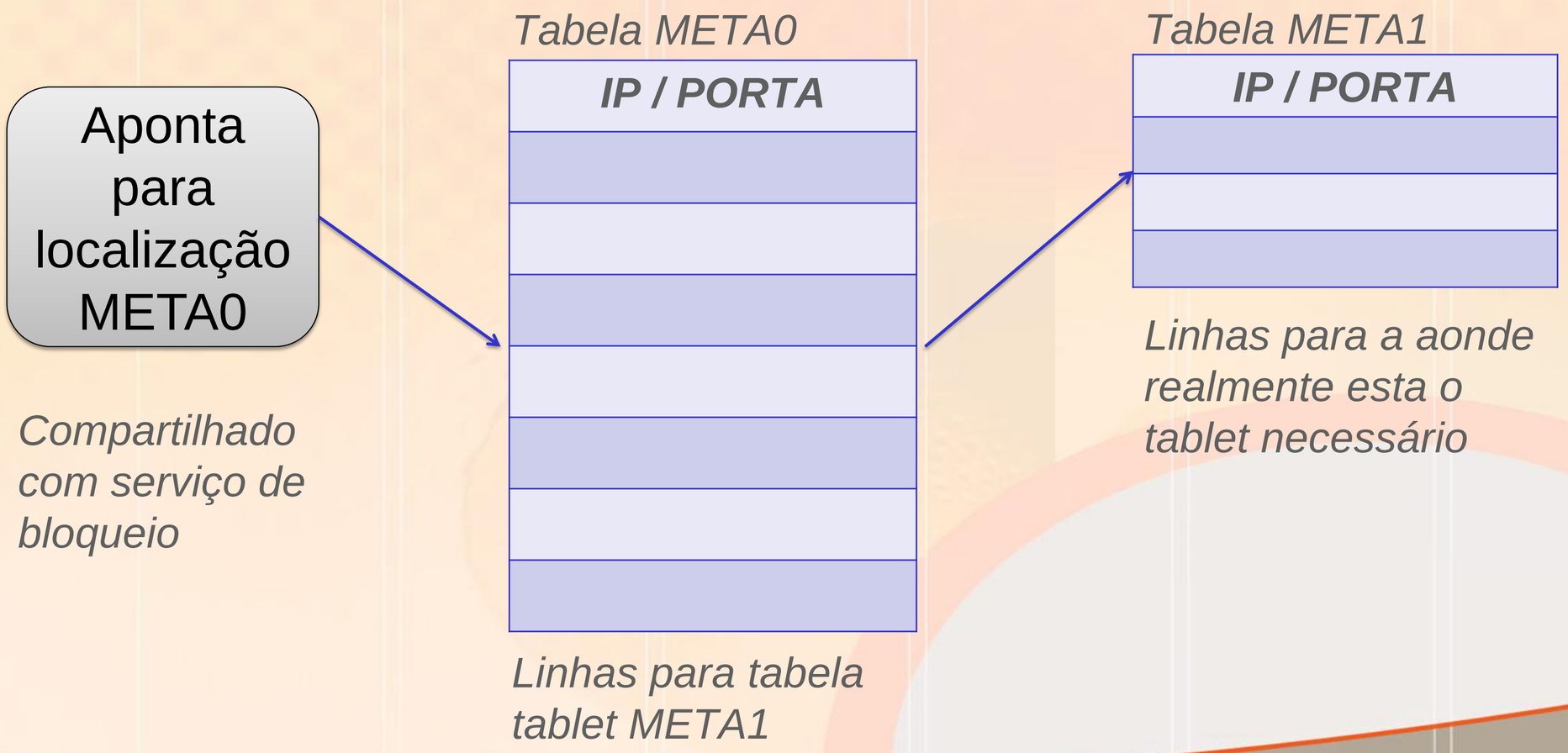
Localização Tablets

- *1° nível : serviço de bloqueio concede acesso, aponta para quem possui o META0*
  Tabela META0 não pode ser dividida, é indivisivel



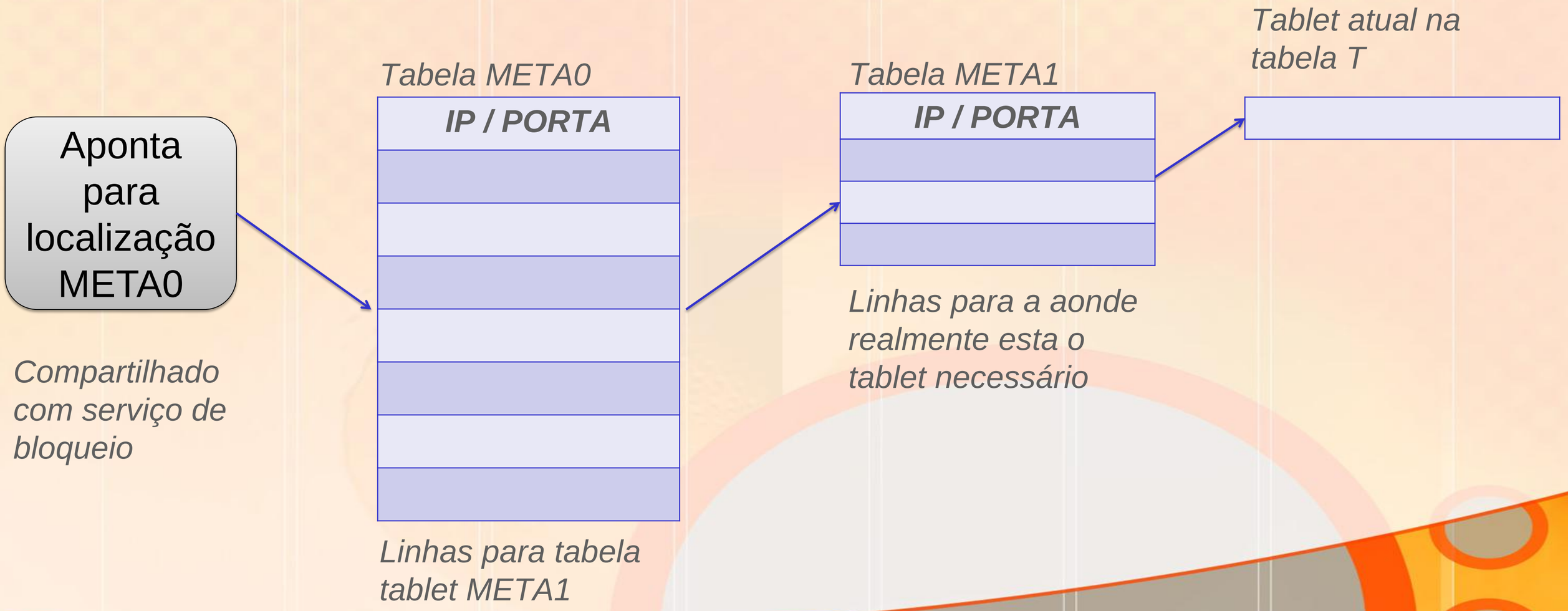
Localização Tablets

- *2° nível :Utiliza o dado META0 para encontrar quem possui o tablet META1*



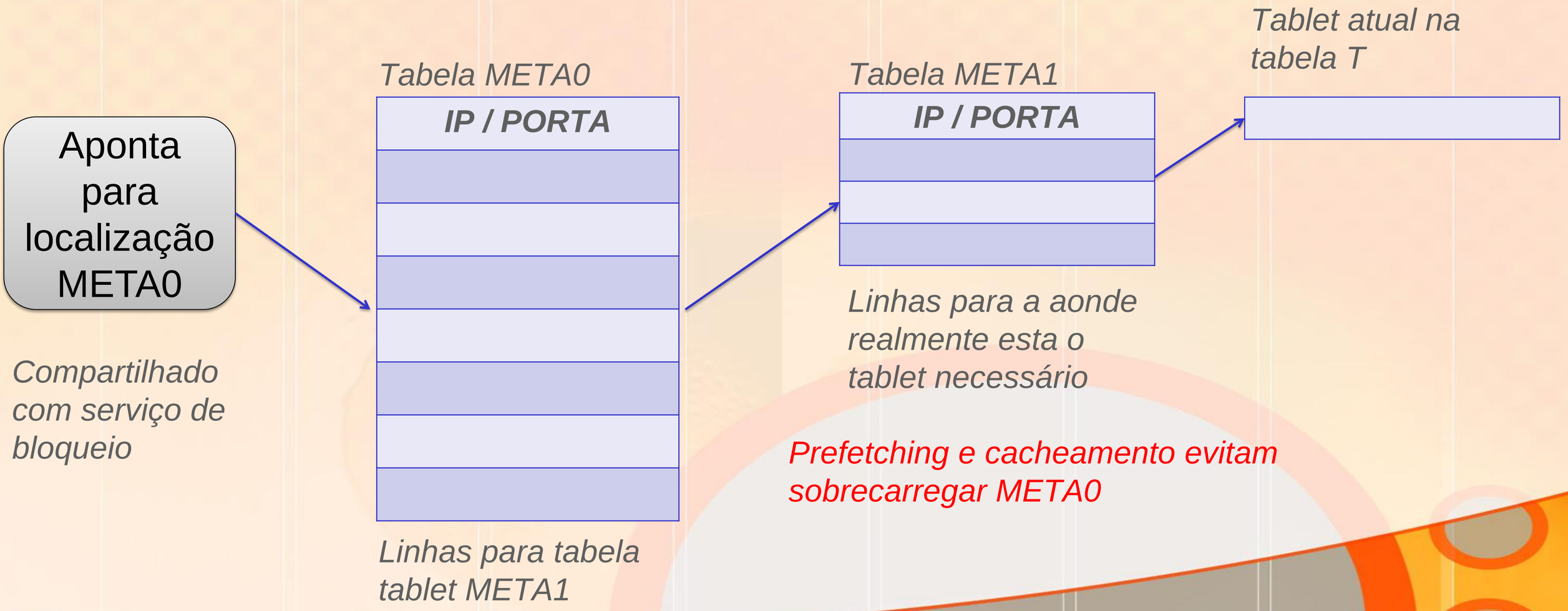
Localização Tablets

- *3° nível : Tabela META1 possui a localização do tablet de todos as outras tabelas*
  Tabela META1 pode ser divida em diversos tablets



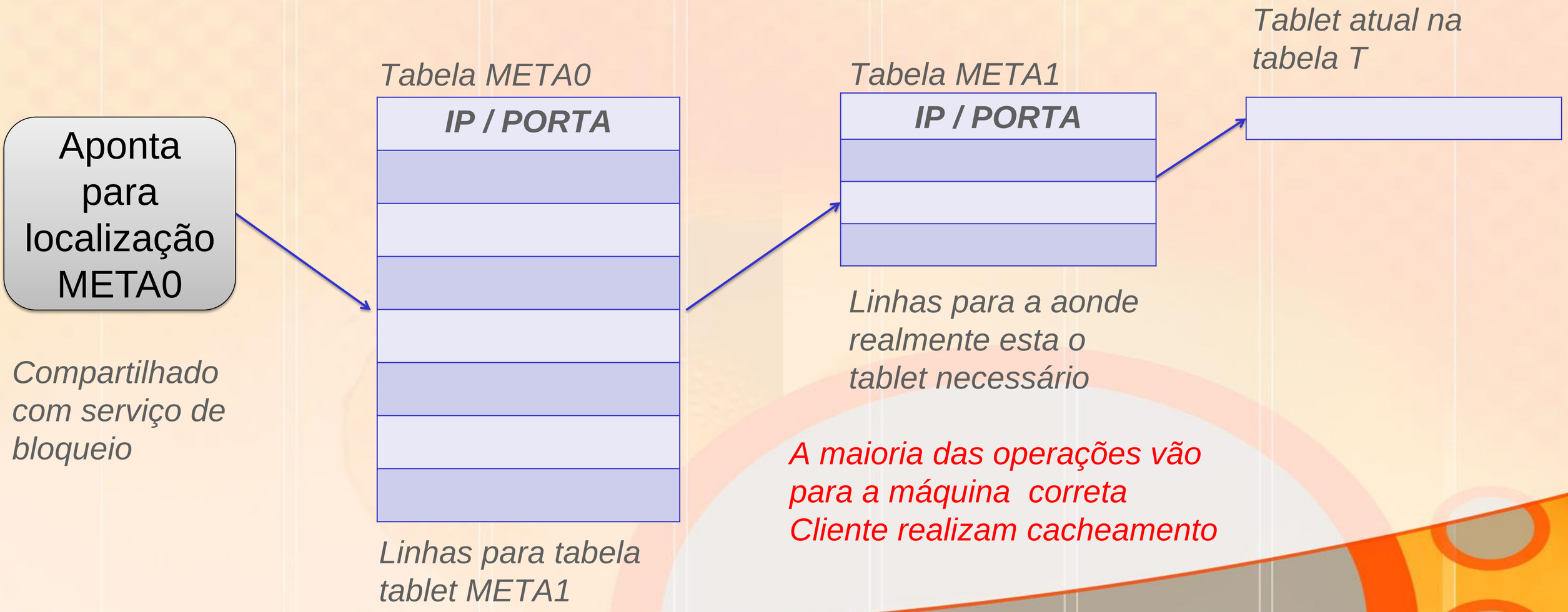
Localização Tablets

- *3° nível : Tabela META1 possui a localização do tablet de todos as outras tabelas*
  Tabela META1 pode ser divida em diversos tablets

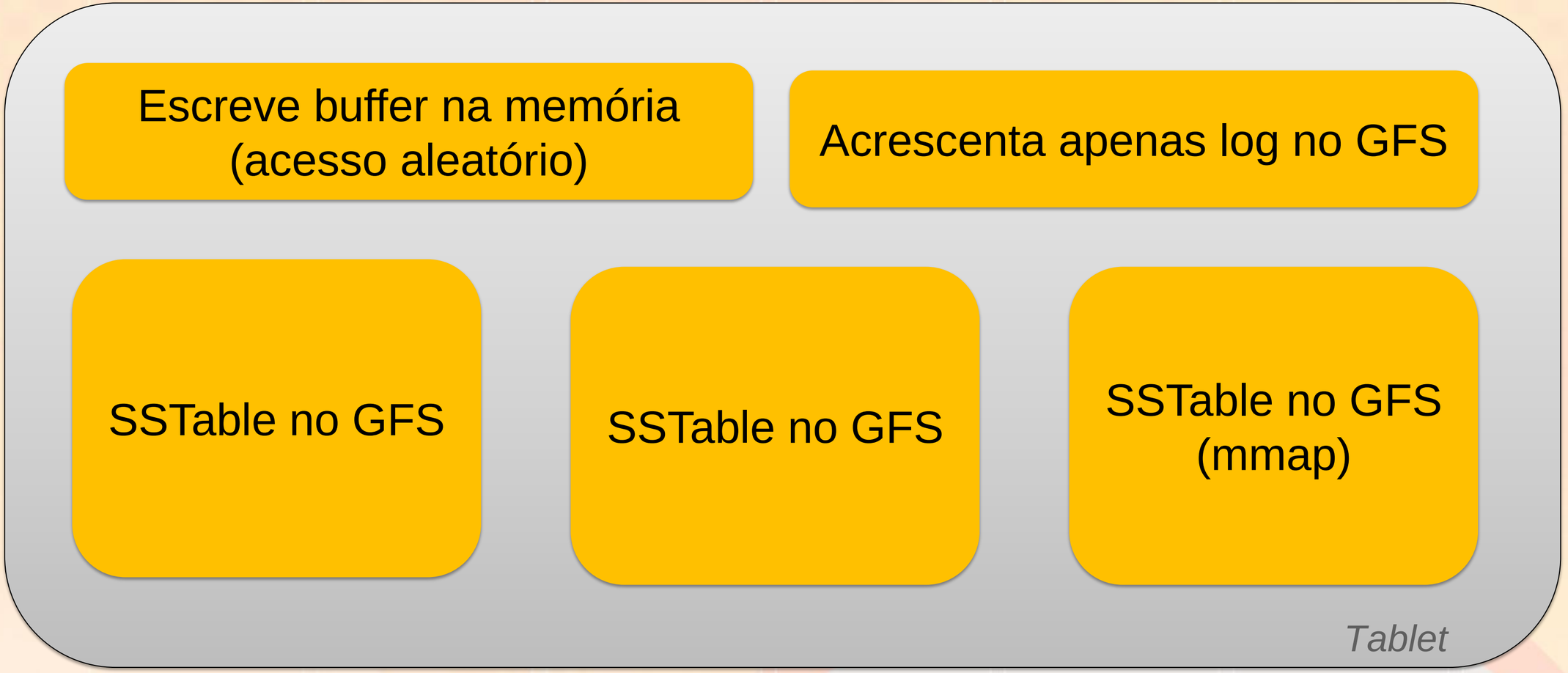


Localização Tablets

- *3° nível : Tabela META1 possui a localização do tablet de todos as outras tabelas*
  Tabela META1 pode ser divida em diversos tablets

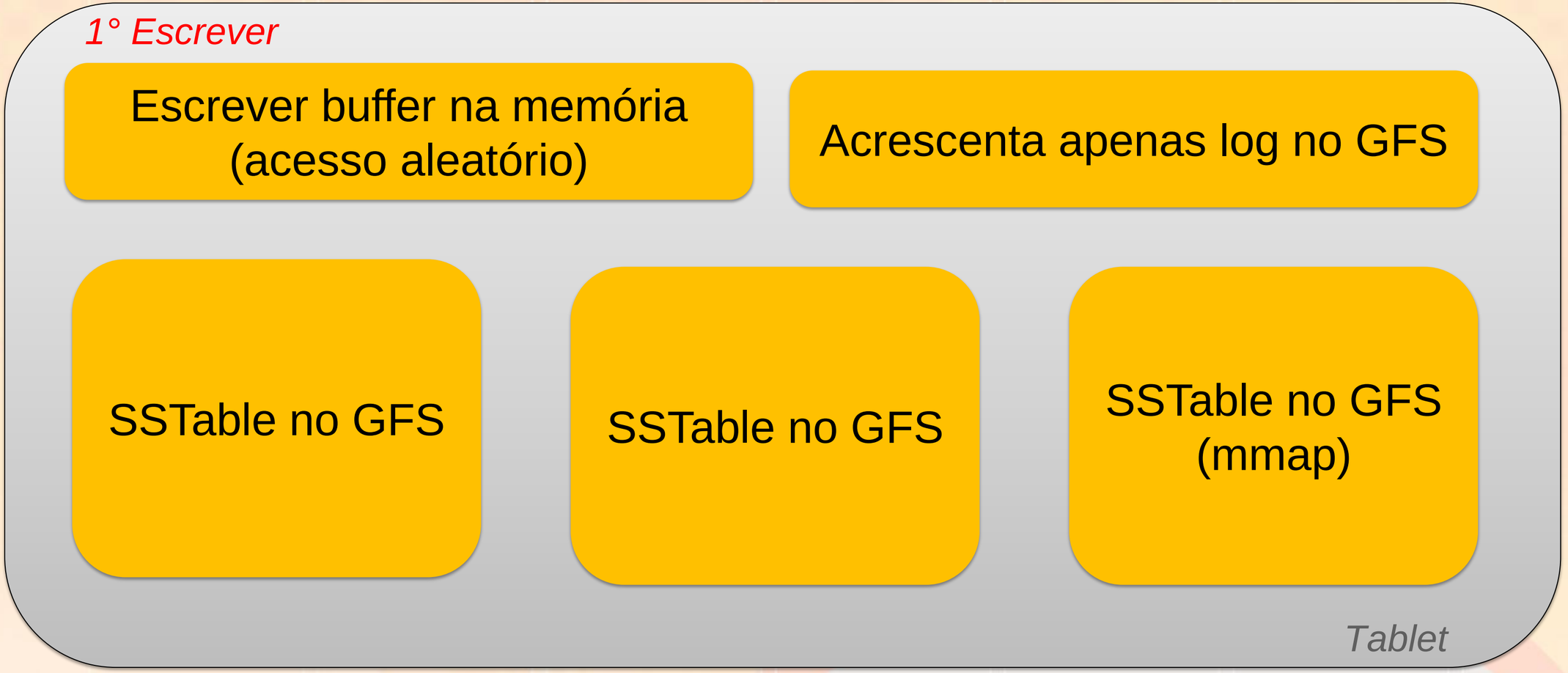


Representação de um tablet



SSTable: Imutável, ordenada em disco em um mapeamento string > string
Chaves string é uma tripla chave: linha, coluna, timestamp

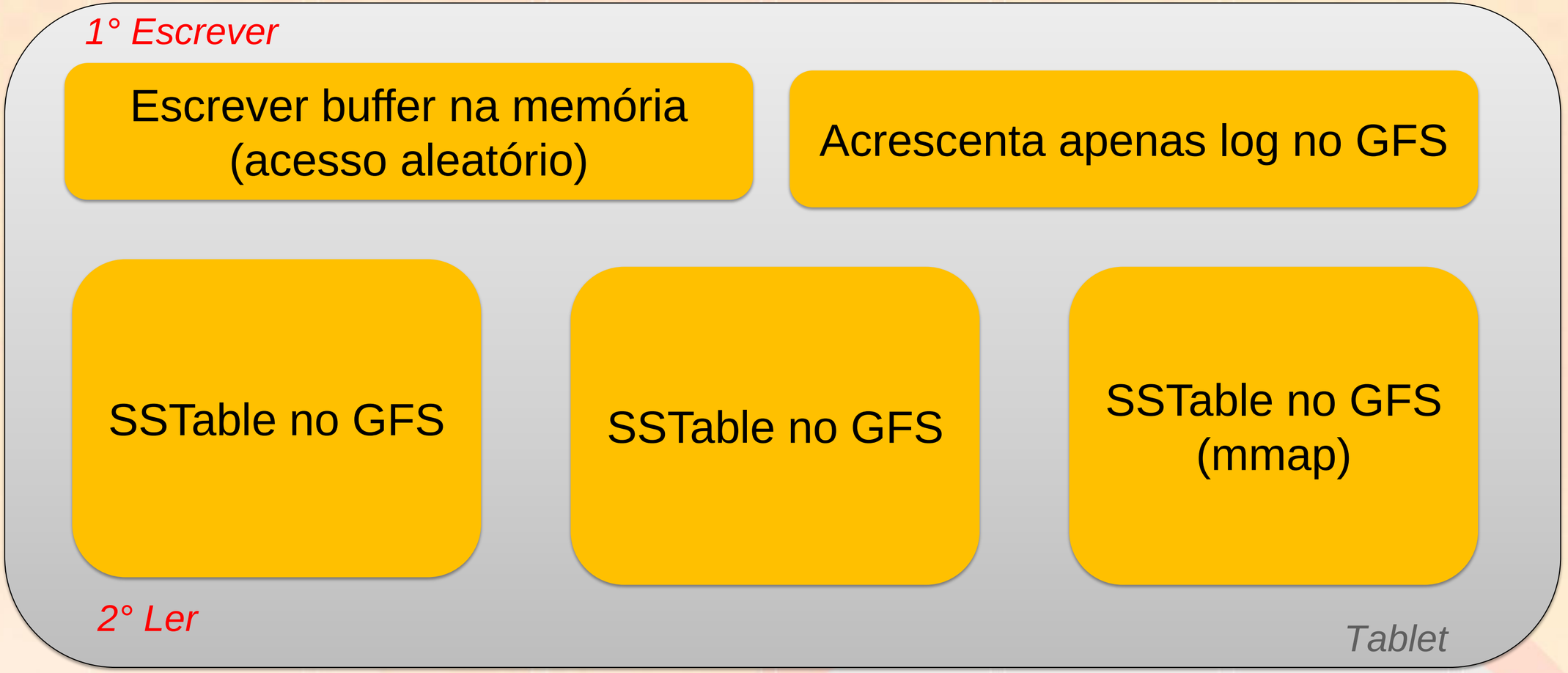
Representação de um tablet



SSTable: Imutável, ordenada em disco em um mapeamento string > string

Chaves string é uma tripla chave: linha, coluna, timestamp

Representação de um tablet



SSTable: Imutável, ordenada em disco em um mapeamento string > string

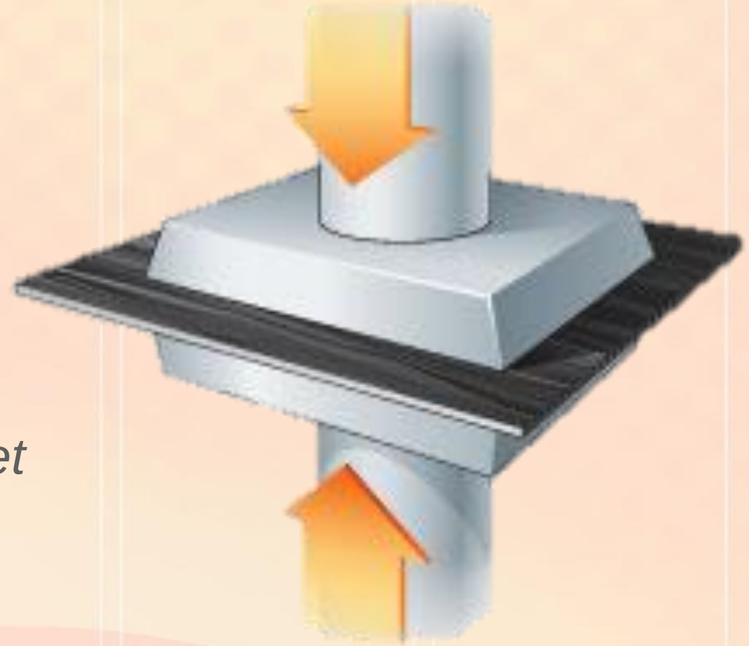
Chaves string é uma tripla chave: linha, coluna, timestamp

Compactações

- *Estado do tablet é representado como conjunto de arquivos SSTable imutáveis mais o log (bufferizado em memória)*

- *Compactação menor*
  - *Ocorre quando a memória esta cheia*
  - *Pega a tablet com a maioria dos dados e escreve seu conteúdo em SSTable armazenadas no GFS*

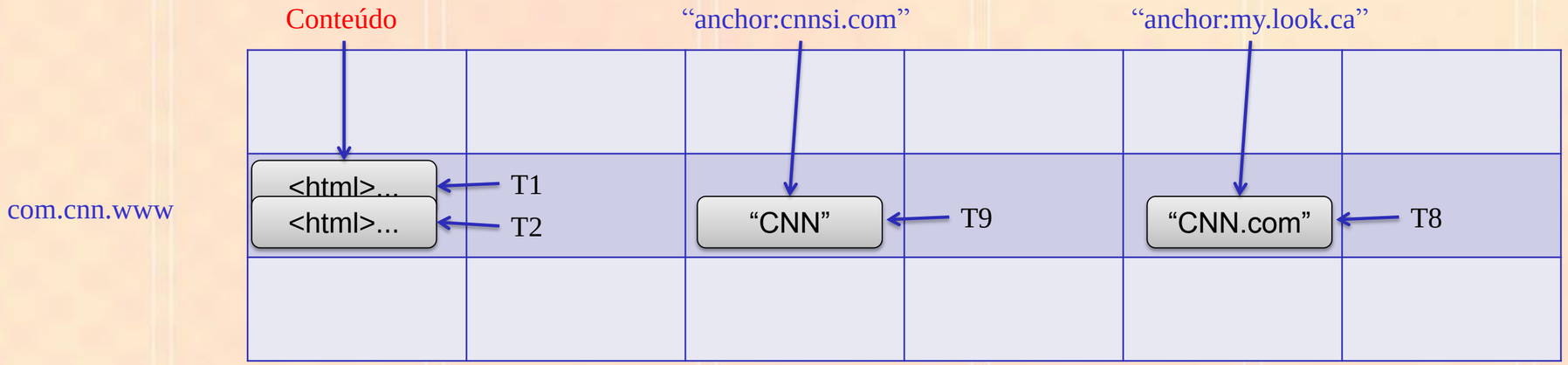
  Arquivos separados para cada localização de grupo para cada tablet



- *Compactação maior*
  - *Periodicamente compacta todas as SSTable para tablets em uma nova base SSTable no GFS*

  Neste ponto é possível recuperar arquivos apagados

Colunas



- *Colunas possuem uma estrutura de 2 níveis de nomes*
  - *família:qualificador_opcional*

- *Coluna família*
  - *Unidade de controle de acesso*
  - *Tem a informação do tipo de associação*

- *Qualificador*
  - *Nível adicional de indexação, se desejável*

TimeStamp

- *Usado para armazenar diferentes versões dos dados em uma célula*
  - *Novas escritas por padrão utilizam o tempo atual*
  - *Mas podem ser colocados explicitamente por clientes*

- *Opções de pesquisa*
  - *“Retornar os k mais recentes”*
  - *“Retornar todos valores em um intervalo”*

- Coluna *família* pode ser marcada com atributos
  - “Apenas mantenha os k valores mais recentes”
  - “Mantenha valores até que sejam mais velhos que k segundos”

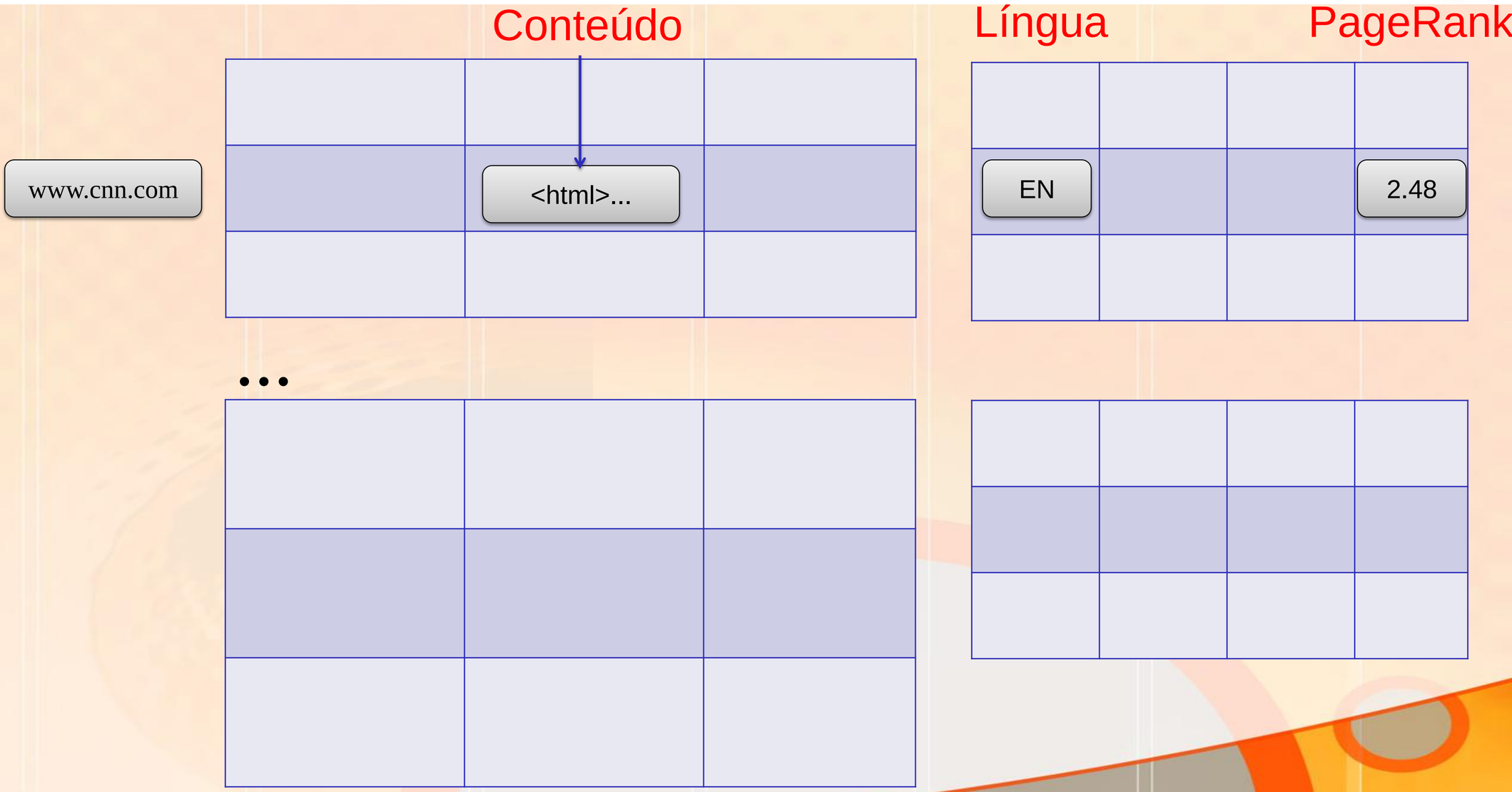
Grupos locais

- *Quando parte da tabela precisa ser pesquisada várias vezes cria-se os grupos locais*

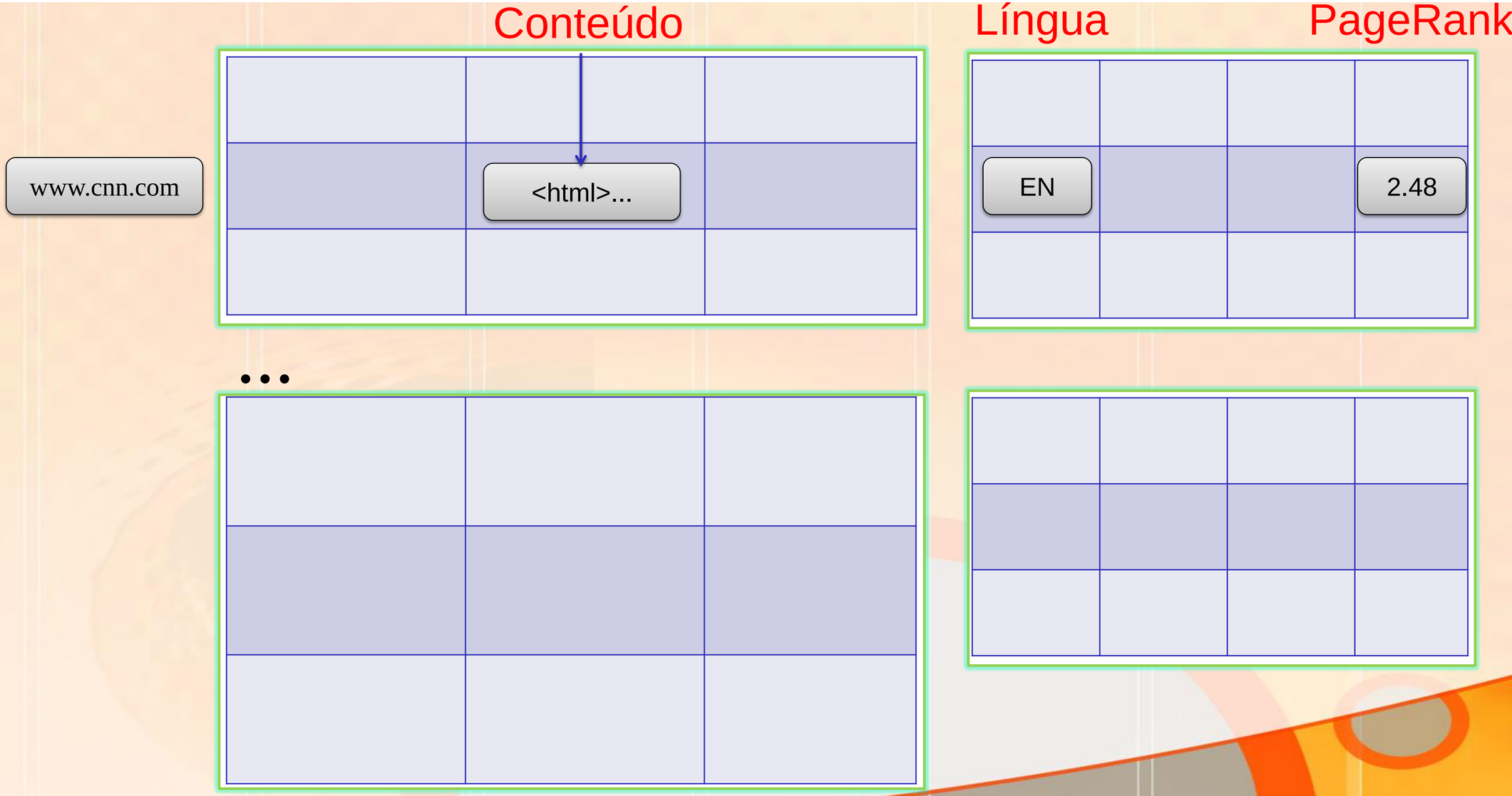
- *A coluna família pode ser designada para grupos locais*
  - *Usado para organizar uma representação de armazenamento subjacentes para desempenho*
  - *Pesquisar sobre em grupos locais é*
    O(bytes_nos_grupos_locais), e não O(bytes_na_tabela)

- *Dados nos grupos locais podem ser explicitamente mapeamentos em memória*

Grupos locais



Grupos locais



API

- *Operações de MetaDados*
  - *Criar / Apagar tabelas, colunas, famílias, mudar metadados*

- *Escritas (atômicas)*
  - *Set() : escreve célula em uma linha*
  - *DeleteCells() : apaga células em uma linha*
  - *DeleteRow() : apaga todas as células em uma linha*

- *Leituras*
  - *Scanner() : lê arbitrariamente células na BigTable*
    - *Cada leitura na linha é atômica*
    - *Pode limitar o intervalo de retorno de linhas*
    - *Pode perguntar pode apenas 1 linha, todas as linhas, etc.*
    - *Pode perguntar por todas as colunas, por certas família de colunas ou colunas específicas*

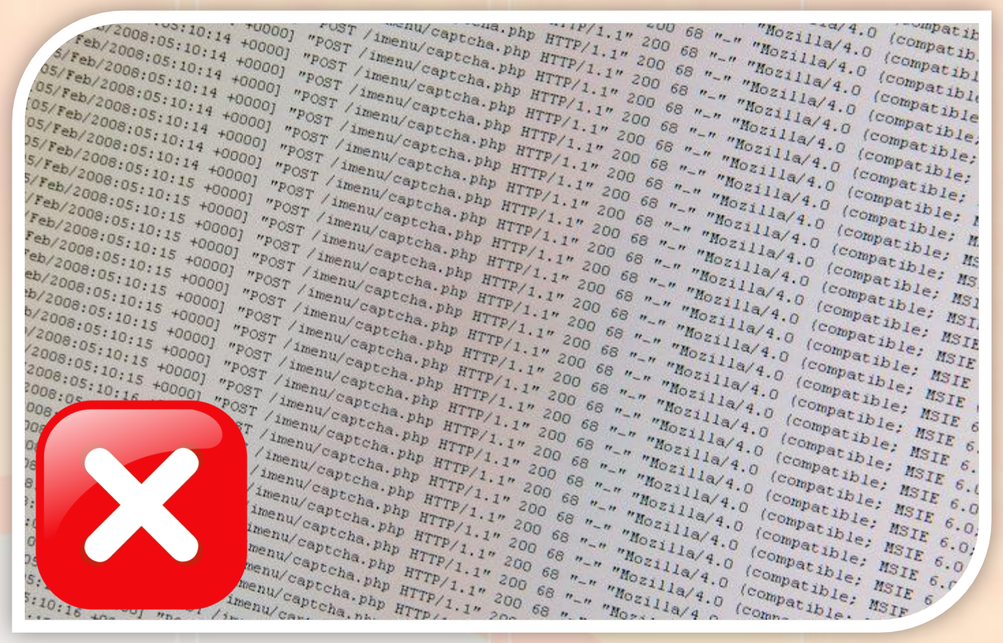
Logs compartilhados

- *Projetado para 1 milhão de tablets, milhares de servidores de tablets*
  - *1 milhão de logs escrevendo simultaneamente tem péssimo desempenho*

- *Solução: Log compartilhado*
  - *Escreve o log por servidor de tablet e não por tablet*

- *Inicia um novo chunk frequentemente (64MB)*

- *Problema*
  - *Durante a recuperação o servidor precisa ler o log para aplicar as mudanças no tablet*



  - *Pode perder muitas I/O se várias máquinas precisam ler dados de diferentes tablets do mesmo chunk*

Logs compartilhados

- *Recuperação*
  - *Servidores informam ao mestre o(s) chuk(s) que eles precisam ler*
  - *Mestres gerenciam e ordenam chunk necessários*
    - *Atribui os chunk dos logs ordenados para diferentes servidores tablets*
    - *Servidor ordena chunk por tablet*
    - *Escreve dados ordenados no disco local*
  - *Outro tablets perguntam ao mestre qual servidor possui o chunk ordenado que eles precisam*
  - *Agora servidores tablets leem dos servidores de tablets*

Compressões

- *Muitas oportunidades para compressões*
  - *Valores similares em uma mesma linha / coluna em diferentes timestamps*
  - *Valores similares em colunas diferentes*
  - *Valores similares entre linhas adjacentes*

- *Dentre de cada SSTable para um grupo local codifica os blocos compressados*
  - *Manter os blocos pequenos para acesso aleatório (+- 64KB dados compressados)*
  - *Explora o fato de muitos dados serem similares*
  - *Precisa que tenha um custo baixo de CPU para codificar / decodificar*

- *Dois blocos são construídos*
  - *BMDiff*
  - *Zippy*

BMDiff

- *Paper: McIlroy Bentley DCC em 1999 : “Data Compression using long common strings”*
- *Entrada : dicionário , fonte*
- *Sáida : sequência de*

  CÓPIA : <x> bytes do deslocamento <y>

  LITERAL:<texto literal>

- *Armazena o hash a cada 32-bytes alinhados que possuem uma fronteira no*
  - *Dicionário*
  - *Fonte processada até o momento*

- *Para cada nova fonte de byte*
  - *Computa um hash incremental dos últimos 32 bytes*
  - *Pesquisa em uma hash table*
  - *Quando encontra, expande o resultado para frente e para trás e emite uma cópia*
- *Codificado : 100MB /segundo*
- *Decodifica : 1000MB /segundo*

Zippy

- *Armazena o hash dos últimos quatro bytes em uma tabela de entrada de 16k*
- *Para todos byte de entrada*
  - *Computar o hash para os últimos quatro bytes*
  - *Pesquisar na tabela*
  - *Emitir CÓPIA ou Literal*

- *Diferenças para o BMDiff*
  - *Janela muito menor de compressão (repetições locais)*
  - *Tabela Hash não é associativa*
  - *Cuidadoso com a codificação CÓPIA / LITERAL tags e tamanhos*

- *Desleixado mas rápido*

Zippy

- *Desleixado mas rápido:*

Algoritmo	% restantes	Codificação	Decodificação
Gzip	13.4 %	21 MB/s	118 MB/s
LZO	20.5 %	135 MB/s	410 MB/s
Zippy	22.2 %	172 MB/s	409 MB/s

Compressão BigTable

- *Chaves*
  - *Strings ordenadas de linha, coluna, timestamp : prefixo da compressão*
- *Valores*
  - *Agrupa valores pelo tipo (ex.: nome da coluna família)*
  - *BMDiff saída para valores 1 a N é o dicionário para N+1*
- *Zippy como último passo sobre o bloco compactado*
  - *Obtém mais repetições localizadas*
  - *Obtém também atráves das repetições da coluna família, compressa as chaves*

Compressão eficiente

- *Experimento*
  - *Armazenar conteúdos de 2.1 bilhões páginas em uma instância da BigTable*
  - *Chave*
    - *URL das páginas com a porção do host-name invertido*
      - *com.cnn.www/index.html:http*
  - *Agrupa páginas do mesmo site*
    - *O que é bom para comprenssão (linhas vizinhas tendem a ter conteúdos similares)*
    - *Bom para clientes, pesquisa rápida sobre todas as páginas em um web site*
  - *Uma estratégia de compressão*
    - *Gzip para cada página : -28 % por byte restante*
  - *BigTable: BMDiff + Zippy*
  - *Velocidade em vez de tamanho*

Compressão eficiente

Tipo	Contagem (byte)	Espaço (TeraByte)	Compressado	% restantes
Conteúdo de páginas web	2.1	45.1 TB	4.2 TB	9.2 %
Links	1.5	11.2 TB	1.6 TB	13.9 %
Âncoras	126.3	22.8 TB	2.9 TB	12.7 %

Experimentos

Experimento	Quantidade de servidores de tablets			
	1	50	250	500
Leituras aletórias	1212	593	479	241
Leituras aletórias (mem)	10811	8511	8000	6250
Escritas aleatórias	8850	3745	3425	2000
Leituras sequênciais	4425	2463	2625	2469
Escritas sequênciais	8547	3623	2451	1905
Pesquisas	15385	10526	9524	7843

Experimentos

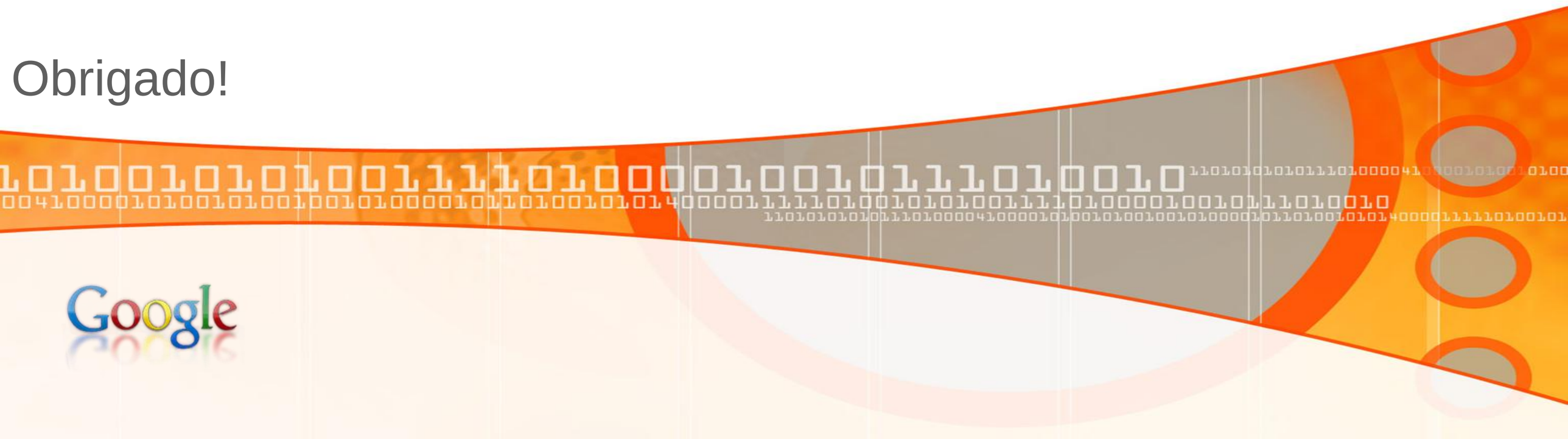
Quantidade de servidores de tablet	Quantidade de clusters
0 ... 19	259
20 ... 49	47
50 ... 99	20
100 ... 499	50
➢ 500	12

Em desenvolvimento / Planos para futuro

- *Manipulação/Acesso dos dados mais expressiva*
  - *Permitir o envio de pequeno scripts que realizem leitura / escrita para que eles executem no servidor ?*
- *Suportar transação de múltiplas linhas distríbuidas*
- *Desempenho em geral para céluas muito grandes*
- *BigTable as a service ?*
  - *Problemas com justiça entre os recursos sobre os diferentes clientes*

Perguntas e respostas

Obrigado!



Google