Seminarium Bazy Danych I

BigTable

Piotr Świgoń
Uniwersytet Warszawski

Rzędy wielkości

- Miliardy URL'i i linków, wiele wersji stron
- Setki milionów użytkowników
- Tysiące zapytań na sekundę

- 2.7 – 3.3 GB - rozmiar angielskiej wikipedii (tekst)
- 100+ TB - rozmiar zdjęć satelitarnych w Google Earth (2005)
- 5.625 PB – szacowany rozmiar przestrzeni dyskowej Google'a (2004)
- 200 PB – wszystkie wydrukowane dane świata

Dlaczego nie komercyjna BD

- Skala za duża dla większości rozwiązań
- Nawet jeśli by nie była, koszty wdrażania w kolejnych projektach byłyby bardzo duże:
  - Wydajny system zbudowany wewnętrznie może być udostępniony jako usługa
- Komercyjne rozwiązania nie współpracowały by dobrze z infrastrukturą Google'a
- Niskopoziomowe optymalizacje pomagają znacząco poprawić wydajność
- *„Fun and challenging to build large-scale systems ;)”*

Cele

- Chcemy żeby asynchroniczne procesy na bieżąco uaktualniały różne fragmenty danych
- Wsparcie dla:
  - Wydajne I/O (miliony operacji na sekundę)
  - Wydajne przeglądanie całości lub podzbioru danych
- Częsta potrzeba sprawdzania zmiany danych w czasie
  - np. zmiany na stronach www

BigTable

- Rozproszona wielowymiarowa mapa
- Wysoka stabilność
- Skalowalna:
  - Tysiące serwerów
  - Terabajty danych w pamięci
  - Petabajty danych na dyskach
  - Miliony operacji I/O na sekundę
- Samo-zarządzająca się:
  - Serwery mogą być dodawane/usuwane dynamicznie
  - Serwery same równoważą obciążenie

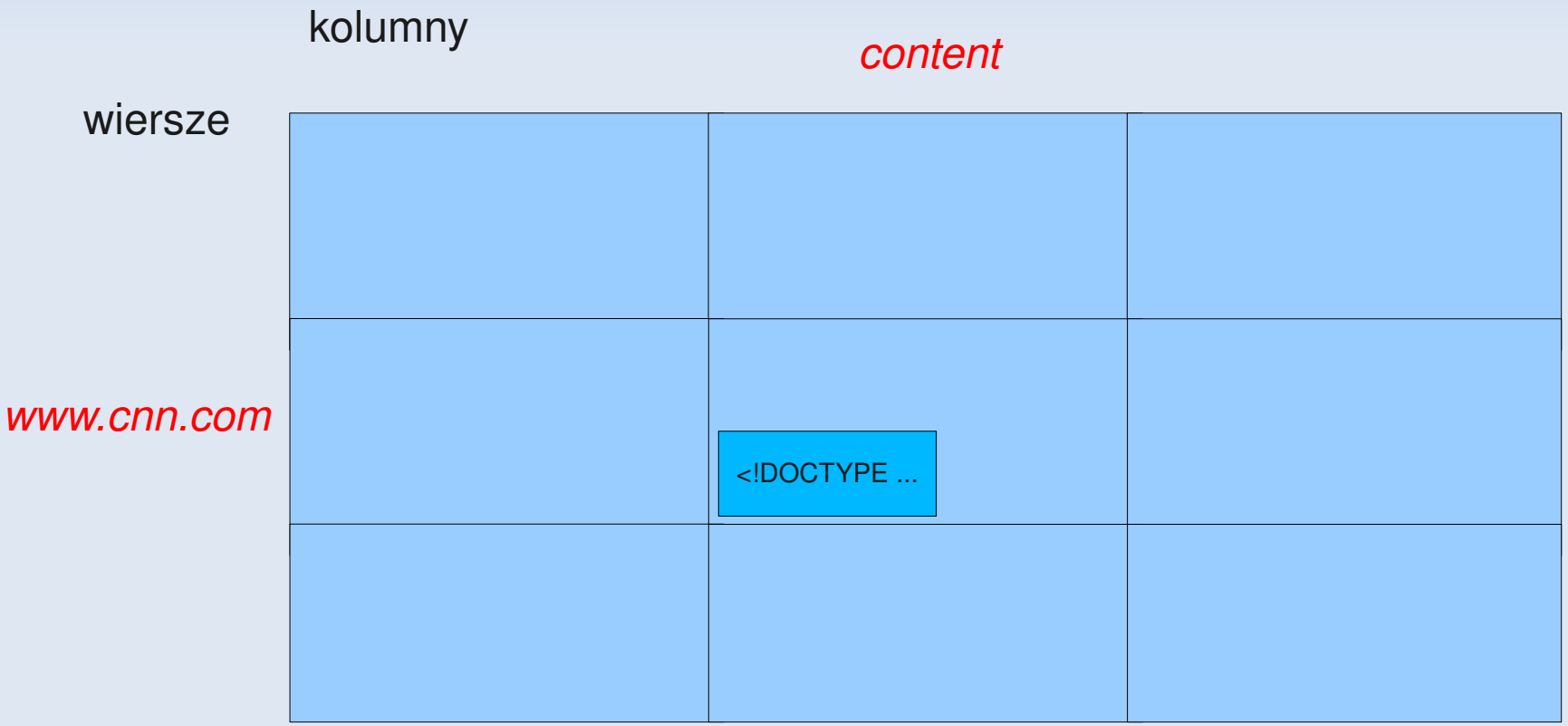
Użyte Technologie

- Google File System (GFS) :
  - Niezawodne przechowywanie danych
- Lock Service (Chubby) :
  - Odpowiada za wybór serwera głównego
- MapReduce :
  - Najczęściej używany do operacji I/O na BT

Model Danych

- Rozproszona wielowymiarowa mapa:

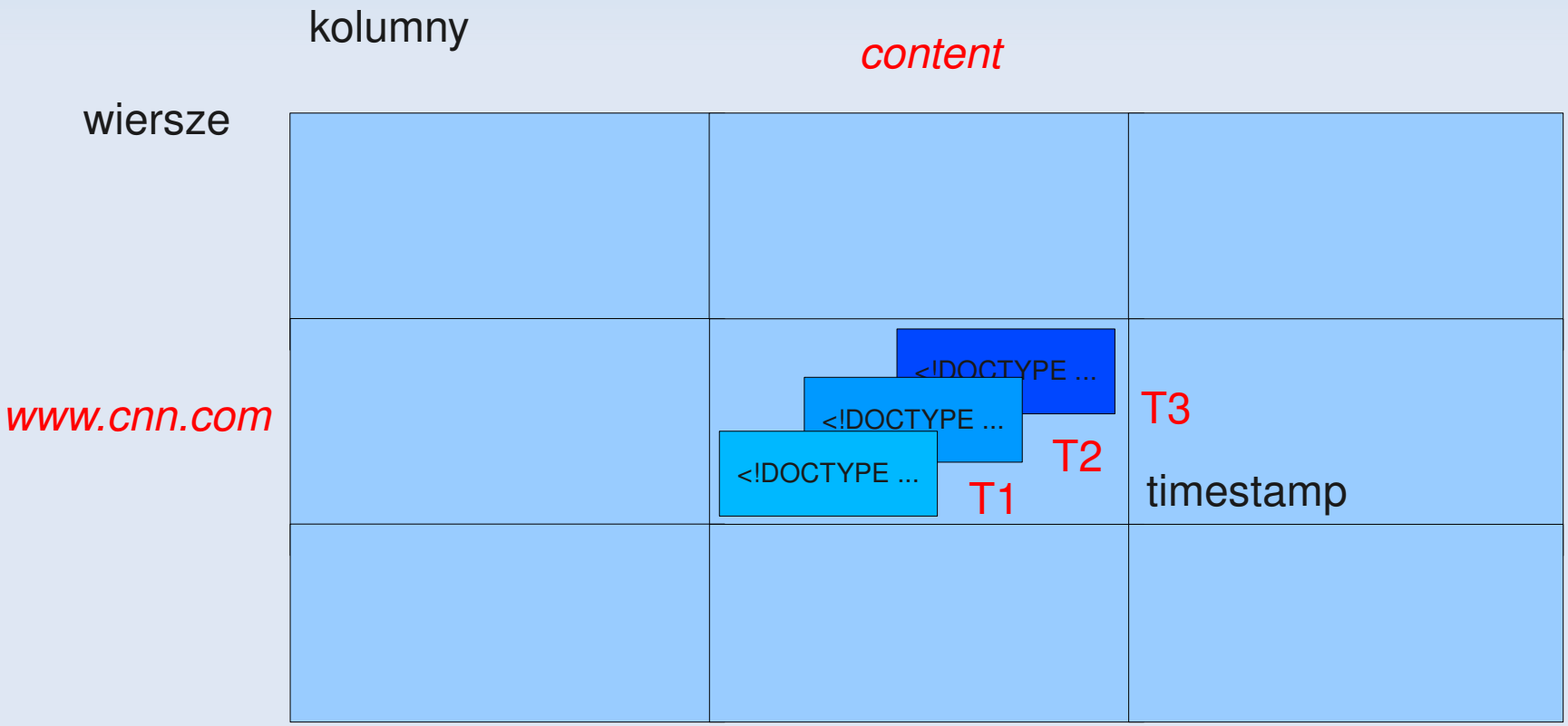
  (wiersz, kolumna, timestamp) → zawartość komórki



Model Danych

- Rozproszona wielowymiarowa mapa:

  (wiersz, kolumna, timestamp) → zawartość komórki



Wiersze

- Identyfikator wiersza jest dowolny stringiem
- Dostęp do wierszy jest atomowy
- Tworzenie wierszy jest automatyczne przy zapisywaniu danych – jak w mapie
- Wiersze są posortowane leksykograficznie
  - Oznacza to że wiersze o zbliżonym kluczu są na jednej lub małej liczbie maszyn

Kolumny

- Kolumny posiadają dwu-poziomową strukturę nazw
  - Rodzina:opcjonalny_kwalifikator
- Do rodziny jest przypisany typ danych
- Kwalifikator daje możliwość posiadania nieograniczonej liczby kolumn – dodatkowy poziom indeksowania

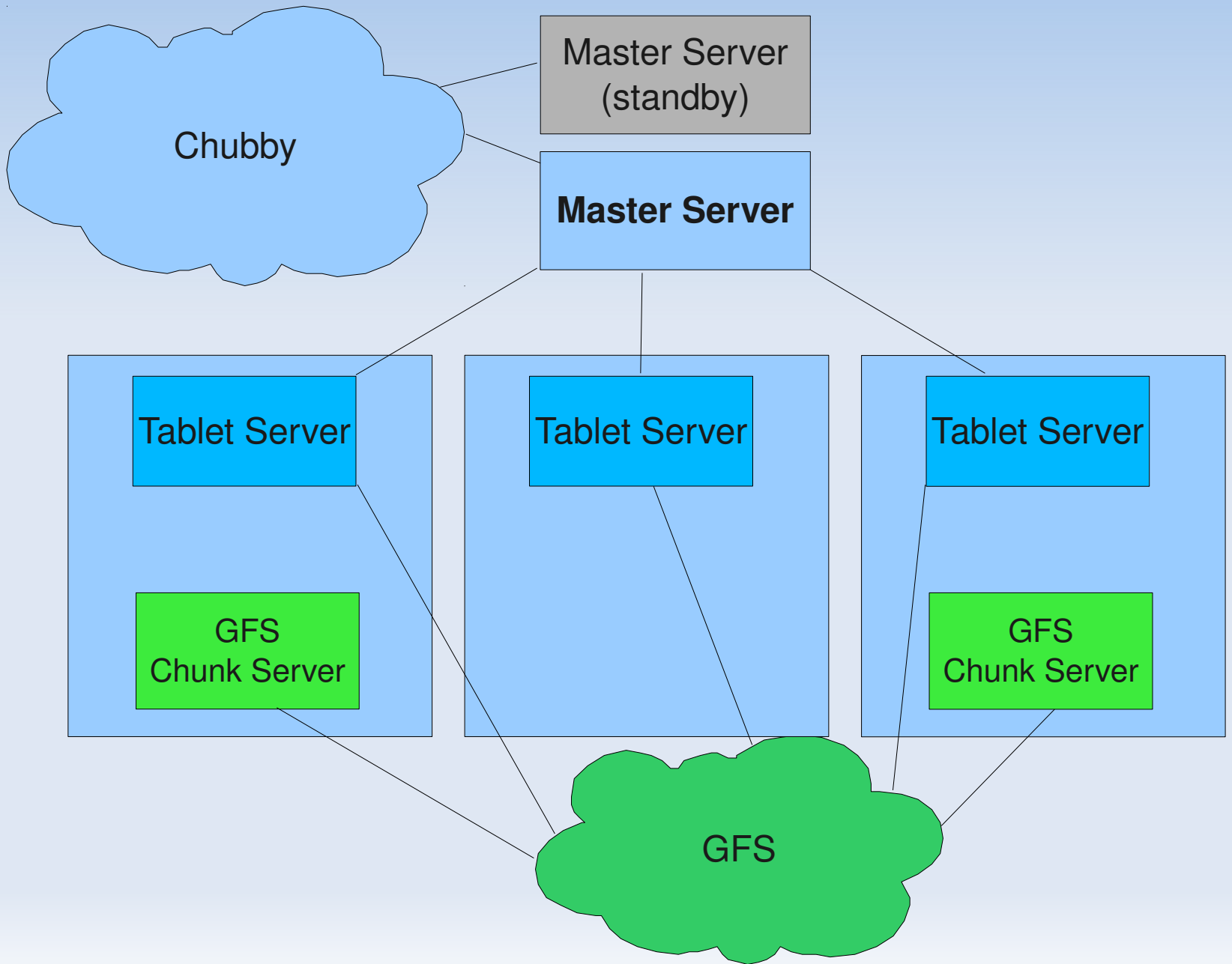
Timestamp

- Używany aby zapisywać różne wersje danych w tabeli
  - Domyślnie przy zapisie używany jest aktualny timestamp, ale można ustawić dowolny
- Opcje przeszukiwania:
  - *Zwróć K najświeższych wartości*
  - *Zwróć wszystkie wartości w przedziale (lub wszystkie w ogóle)*
- Rodziny mogą być dodatkowo oznaczone atrybutami:
  - *Przechowuj ostatnie K wartości*
  - *Przechowuj wartości które są młodsze niż K sekund*

Bloki (Tablets)

- Duże tablice są dzielone na bloki
  - Bloki przechowują ciągły podzbiór wierszy
  - Właściwy rozmiar bloku to 100-200 MB
- Pojedynczy serwer BT jest odpowiedzialny za ~100 bloków
  - Szybka obsługa awarii – 100 maszyn przejmuje po 1 bloku od uszkodzonej maszyny
  - Bloki migrują między maszynami w celu równoważenia obciążenia

Architektura



Serwery Bloków – Tablet Servers

- Odpowiedzialne za żądania zapisu/odczytu od klientów. Dane nie przechodzą przez serwer główny
- Jednemu serwerowi może być przypisane kilkaset bloków
- Nowe serwery bloków mogą być z łatwością dołączane/odłączane do systemu

Serwer Główny – Master Server

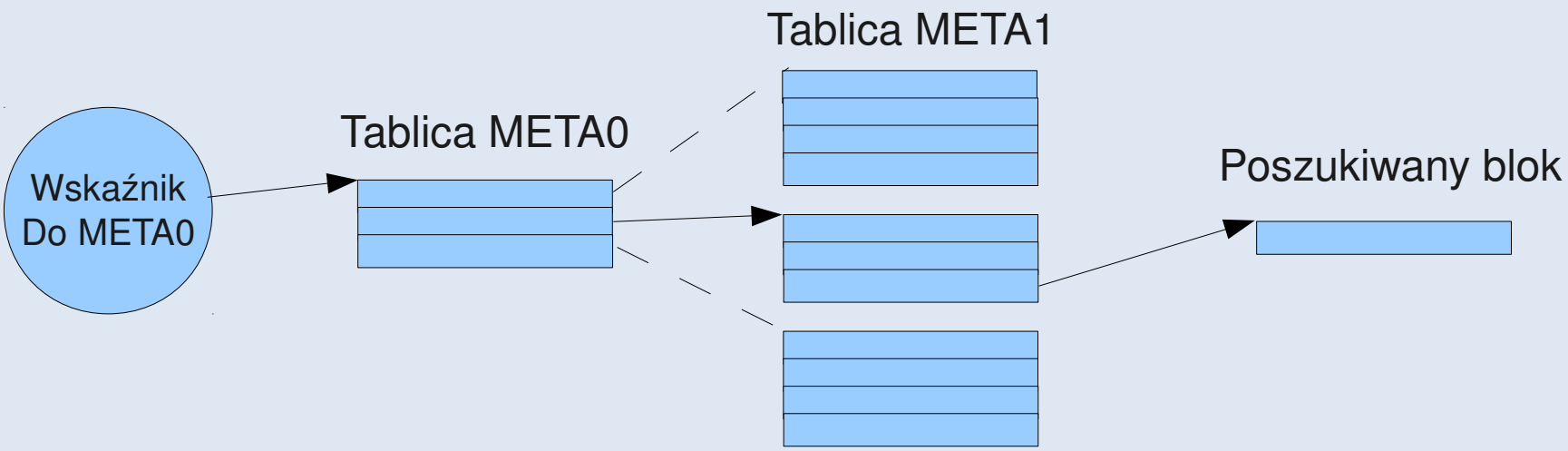
- Przydziela bloki do serwerów bloków
- Wykrywa i obsługuje dołączanie i odłączenie serwerów bloków
- Wykrywa i obsługuje problemy z maszynami
- Rozprasza obciążenie między serwery

Lokalizowanie Bloków

- Bloki migrują między serwerami, jak więc znaleźć maszynę, która obsługuje zadany wiersz?
  - Trzeba znaleźć blok, którego przedział wierszy pokrywa szukany wiersz
- Jedno podejście: Serwer główny mógłby być za to odpowiedzialny
  - Na pewno okazałby się wąskim gardłem
- Zamiast tego: Utrzymywać w BT specjalną tabelę zawierającą informacje o lokalizacji bloków

Lokalizowanie Bloków

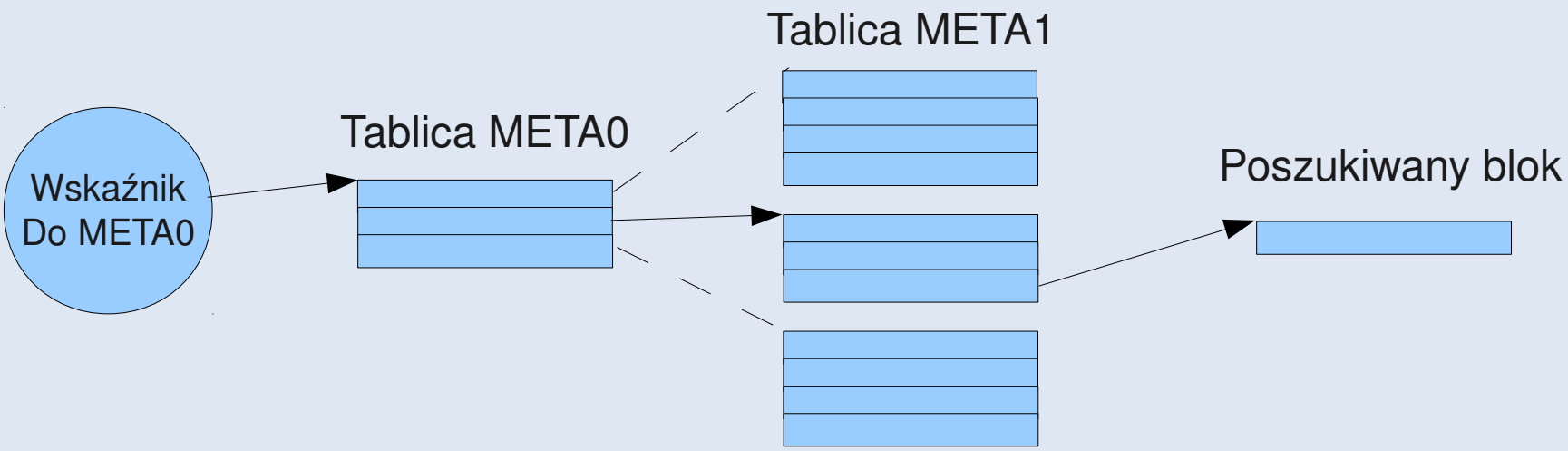
- Podejście użyte w BT – trzy-poziomowa hierarchia wyszukiwania
  - Lokalizacja bloku to *ip:port* serwera na którym blok się znajduje
  - 1 poziom: Znajdujący się w lock service wskaźnik do lokalizacji META0
  - 2 poziom (META0): Jeden wiersz dla każdego bloku tablicy META1
  - 3 poziom (META1): Jeden wiersz dla każdego bloku w systemie



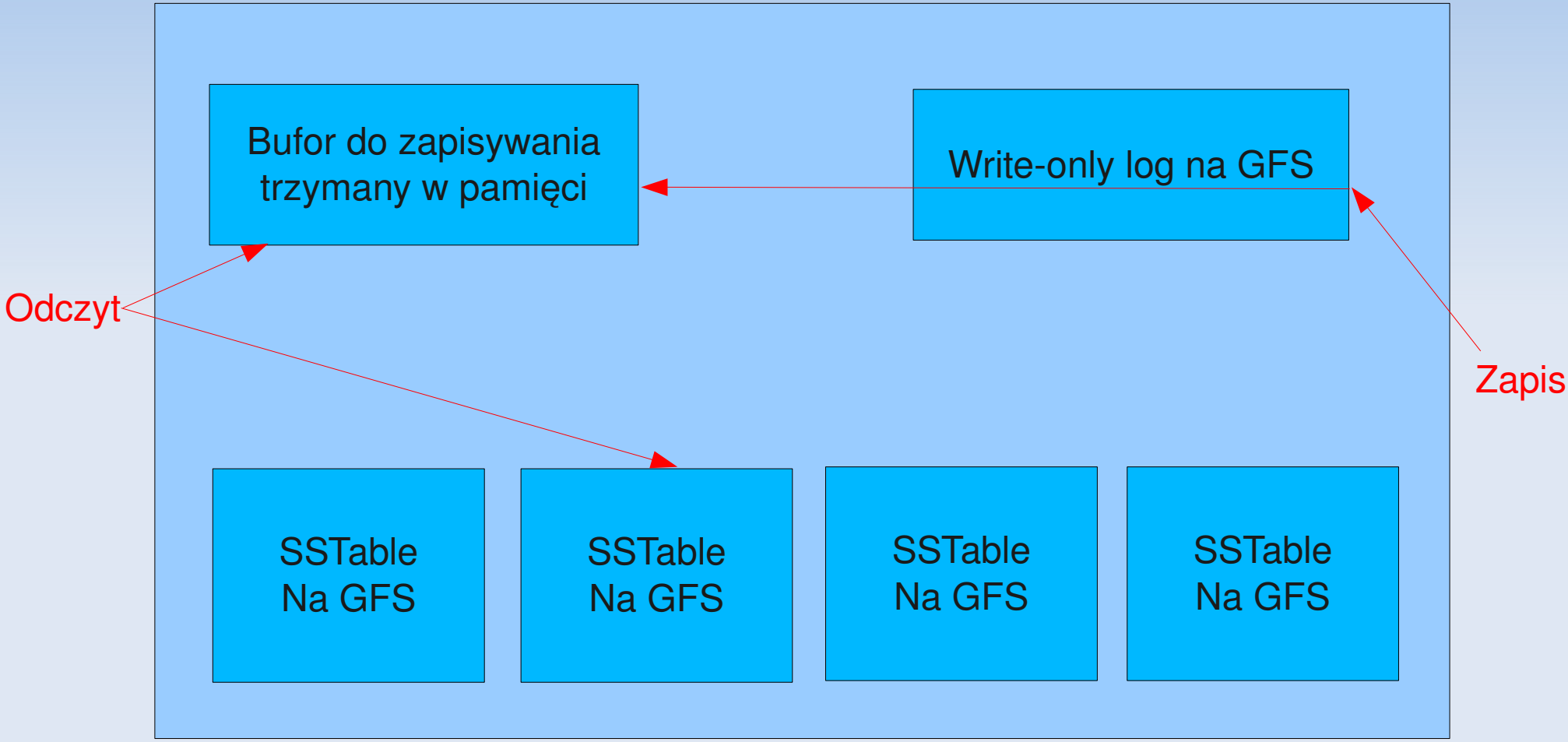
Lokalizowanie Bloków

- Podejście użyte w BT – trzy-poziomowa hierarchia wyszukiwania
  - Lokalizacja bloku to *ip:port* serwera na którym blok się znajduje
  - 1 poziom: Znajdujący się w lock service wskaźnik do lokalizacji META0
  - 2 poziom (META0): Jeden wiersz dla każdego bloku tablicy META1
  - 3 poziom (META1): Jeden wiersz dla każdego bloku w systemie

Agresywny prefetching i caching
Większość zapytań idzie od razu do dobrej maszyny



Blok – Organizacja Danych



SSTable: Niezmienna posortowana mapa String → String leżąca na dysku.
Klucze w tej mapie to krotki <wiersz, kolumna, timestamp>

Blok – Organizacja Danych

- Stan bloku reprezentowany jest przez zbiór niezmienny plików SSTable oraz „ogona” zalogowanych zmian trzymanych w pamięci.
- Małe „przepakowanie”
  - Gdy wyczerpie się limit pamięci, wybieramy plik SSTable dla którego zarejestrowano najwięcej zmian, i go przepisujemy.
- Duże „przepakowanie”
  - Okresowe przepisywanie wszystkich plików SSTable w ramach bloku, w celu przywrócenia im pożądanych rozmiarów oraz zwolnienia miejsca usuniętych wierszy.

Grupy Dostępu

- Rodziny kolumn mogą mieć wspólną grupę dostępu (*ang. Locality Group*)
- Używane aby optymalizować reprezentację danych na dysku
- Rodziny kolumn z tą samą grupą dostępu są trzymane razem w plikach SSTable
  - Gdy każda kolumna jest w osobnej grupie dostępu – mamy do czynienia z kolumnowo zorientowaną bazą danych
  - Gdy wszystkie kolumny są w jednej grupie dostępu – mamy do czynienia z wierszowo zorientowaną bazą danych
- Warto oddzielać dane które nie będą odczytywane razem

Kompresja

- Klienci mogą zdefiniować czy i jak dana grupa dostępu (plik SSTable) jest kompresowana.
- Każdy plik jest kompresowany osobno (powoduje to gorszy współczynnik kompresji, ale odczyty mogą zostać dokonane bez dekompresji całego bloku)
- Algorytm kompresji jest pomyślany, aby był po pierwsze szybki, a dopiero po drugie wydajny.
- kompresja dokonuje się z prędkością 100-200 MB/s, a dekompresja 400-1000 MB/s.

Ciekawostki

- Istnieje open-source'owa implementacja zarówno BigTable jak i MapReduce
- W Google istnieje nakładka na BigTable która zapewnia ograniczoną tranzakcyjność i obsługuje join'y
- Niektóre projekty w Google używają MySQL (+ Hibernate) do danych które z założenia nie muszą być tak efektywnie obsługiwane

Q & A