Индексация документов

Лекция 8

БГУ ФПМИ, 2018

План

План

Динамический индекс

- Обновление порциями (batch)
- Две стратегии:
  1. REBUILD
  2. REMERGE

Динамический индекс

Пусть все документы одного размера. В индексе было d_{old} документов, на вставку пришло d_{ins}, на удаление — d_{del}.

$$Time(REBUILD) = c(d_{old} - d_{del} + d_{ins})$$

$$Time(REMERGE) = ?$$

Динамический индекс

- Можно искать по части, находящейся в оперативной памяти, но еще не подлитой в индекс.
- Сложности с удалением: применяются списки инвалидации и сборка мусора.

Logarithmic Merge

```
LMERGEADDTOKEN(indexes, Z_0, token)
 1  Z_0 ← MERGE(Z_0, {token})
 2  if |Z_0| = n
 3     then for i ← 0 to ∞
 4          do if I_i ∈ indexes
 5                then Z_{i+1} ← MERGE(I_i, Z_i)
 6                     (Z_{i+1} is a temporary index on disk.)
 7                     indexes ← indexes − {I_i}
 8                else I_i ← Z_i    (Z_i becomes the permanent index I_i.)
 9                     indexes ← indexes ∪ {I_i}
10                     BREAK
11          Z_0 ← ∅
```

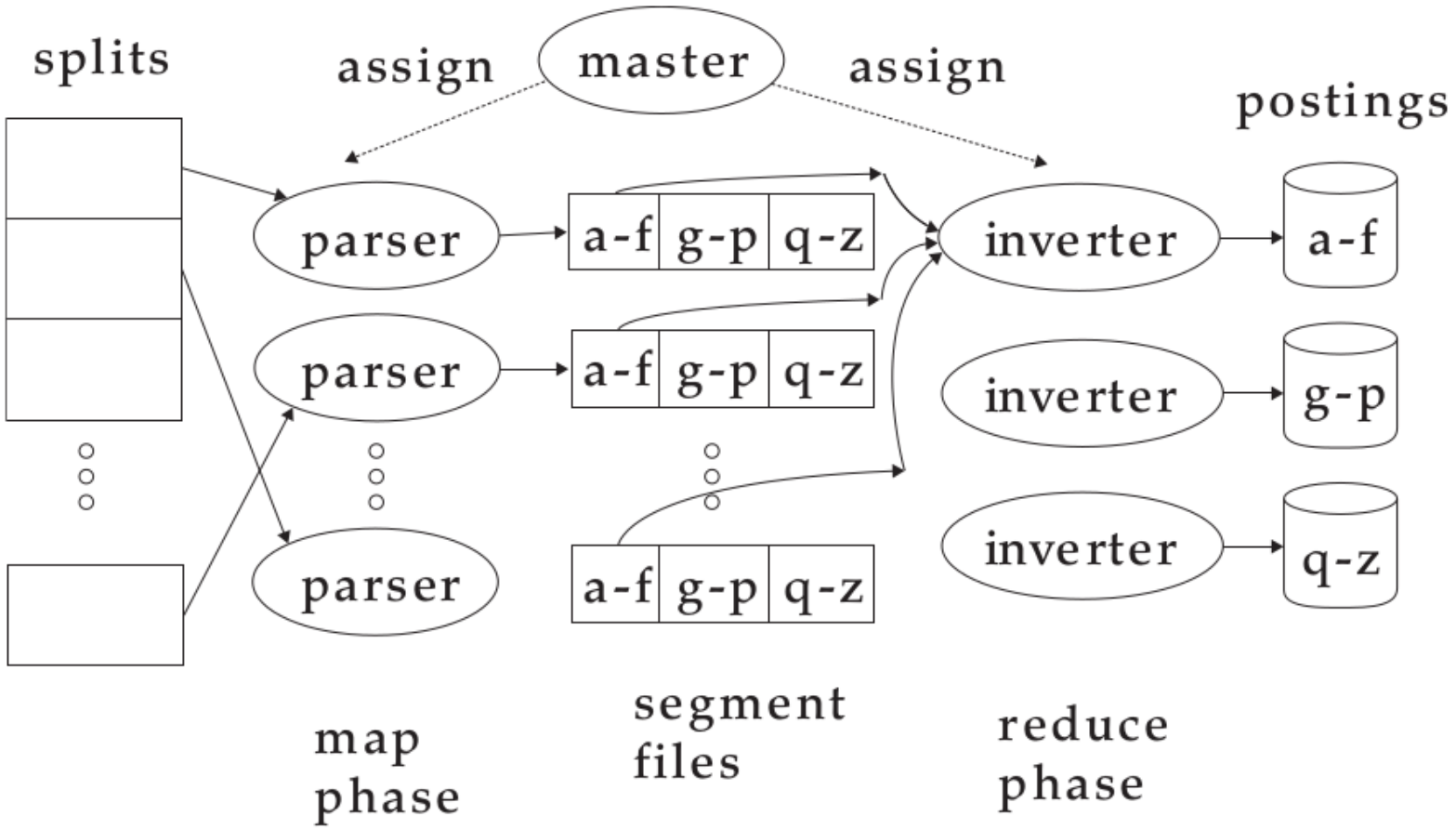
```
LOGARITHMICMERGE()
1  Z_0 ← ∅    (Z_0 is the in-memory index.)
2  indexes ← ∅
3  while true
4  do LMERGEADDTOKEN(indexes, Z_0, GETNEXTTOKEN())
```

Сложность — $\Theta(T \log(T/n))$.

План

Распределенный индекс

- Строится распределенно на кластере (например, посредством MapReduce).
- Распределенное хранение, по шардам:
  - term-sharded
  - doc-sharded


splits
assign
master
assign
postings
parser
parser
parser
a-f g-p q-z
a-f g-p q-z
a-f g-p q-z
inverter
inverter
inverter
a-f
g-p
q-z
map phase
segment files
reduce phase

Term VS Doc sharding

Term-sharding

- Расходы на пересылку по сети списков словопозиций по различным термам оказываются слишком велики.
- Сложности с балансировкой нагрузки из-за перераспределения совместной встречаемости.

Term VS Doc sharding

Term-sharding

- Расходы на пересылку по сети списков словопозиций по различным термам оказываются слишком велики.
- Сложности с балансировкой нагрузки из-за перераспределения совместной встречаемости.

В Вебе применяется подокументное шардирование.

- Разбиение документов соответствует равномерной нагрузке на кластер.
- При поиске top-k результатов с каждого шарда объединяются.

План

При поиске запроса в индексе каждому документу можно сопоставить счетчик, в котором суммируются вклады термов запросов (значения tf-idf).

Term-at-a-time

Списки словопозиций термов запроса обрабатываются последовательно, от терма к терму.

Document-at-a-time

Сортированные по docId списки словопозиций обрабатываются параллельно.

Оптимизации

- Упорядоченные по статическому критерию списки словопозиций.
- Упорядочивание словопозиций по убыванию tf (только term-at-a-time).
- Пропускать высокочастотные термы (term-at-a-time).

Tiering

Индекс состоит из нескольких уровней в зависимости от частоты термов или популярности документов. Обычно tier-ы сильно отличаются по размерам.
Если поиск по tier1 не дал удовлетворительных результатов, то повторяем на tier2 и т.д.

Трехуровневое кэширование

- Результаты поиска
- Пересечение списков
- Списки словопозиций

Применяется LRU или LFU кэширование.

Следующая лекция

Оценка качества поисковых систем