Detecting near-duplicates for web crawling

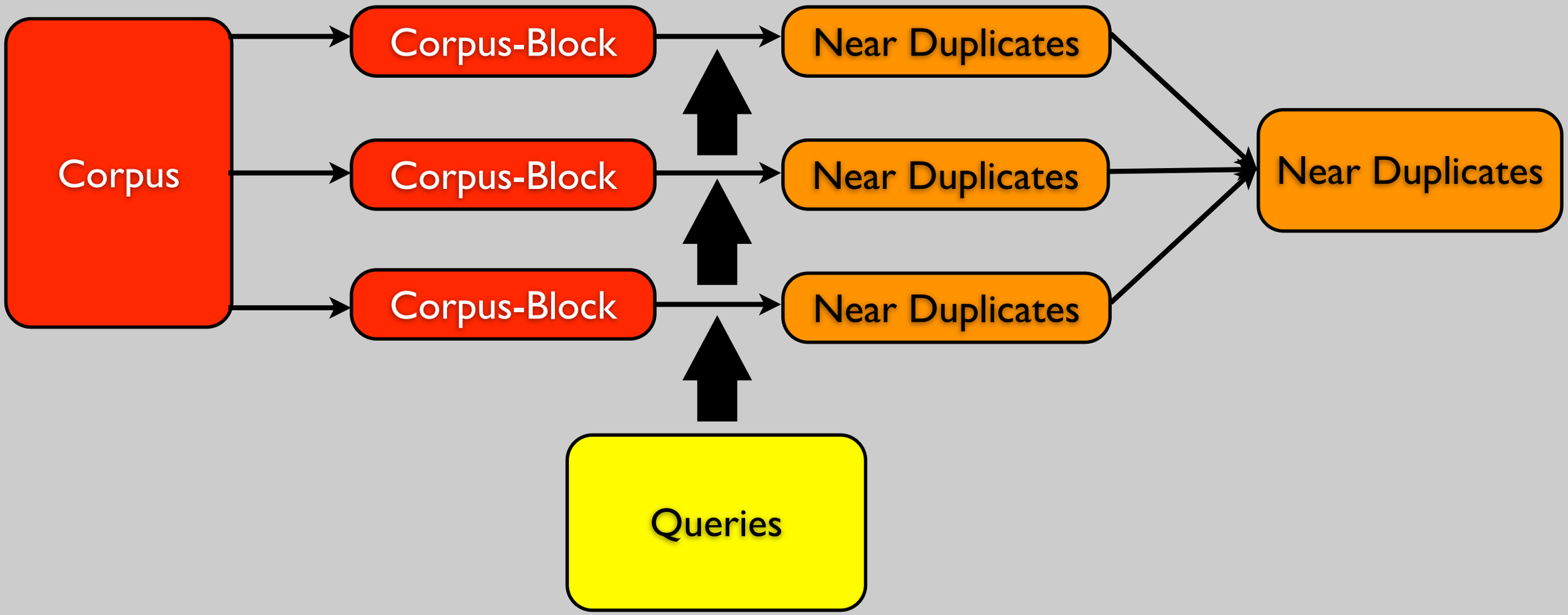
Ziawasch Abedjan
Tobias Flach

Agenda

- Problemstellung
- Generierung der Daten
- Implementierung
- Nächste Schritte

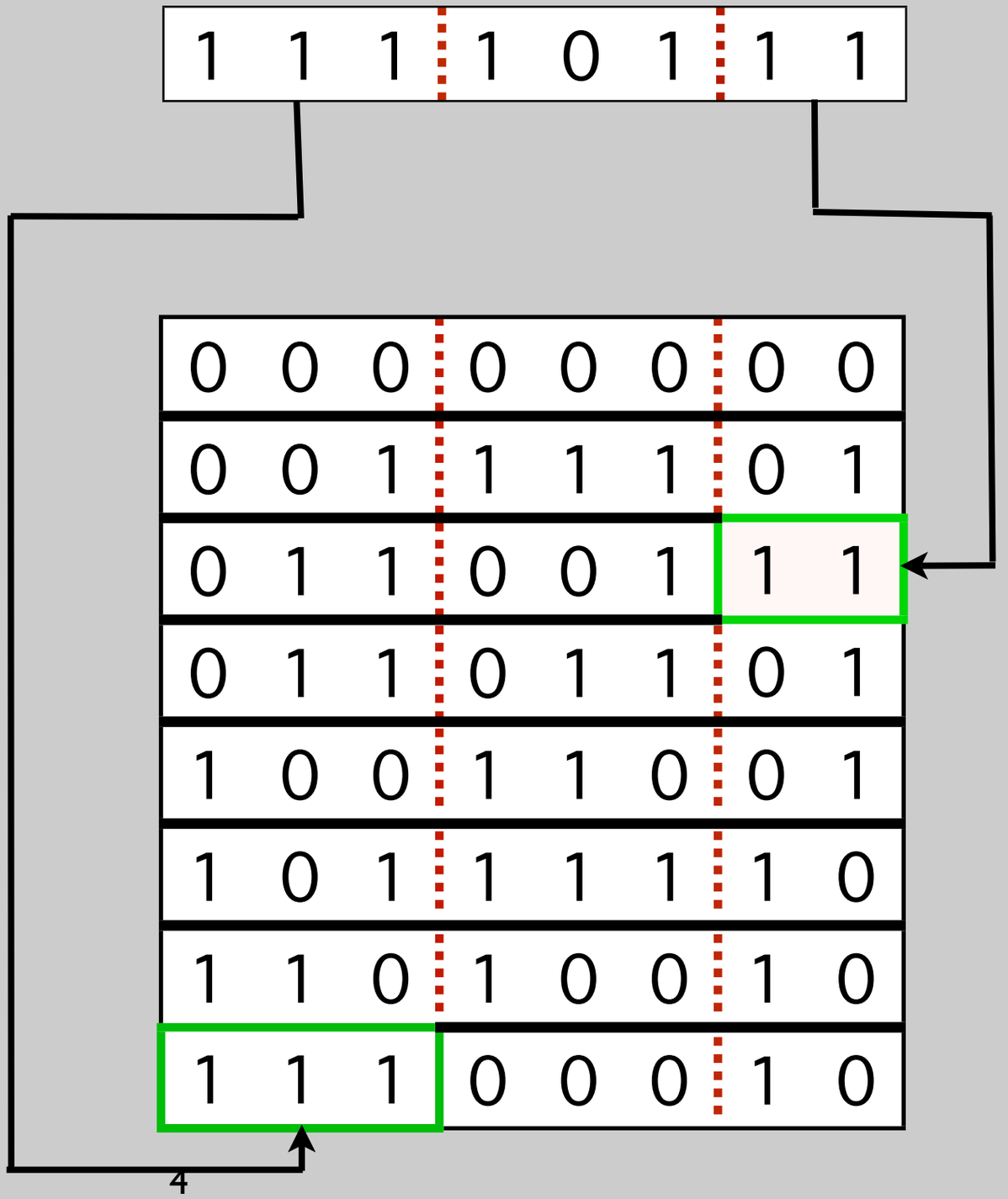
Problemstellung

- Aufgabe: Ähnliche Dokumente finden (near-duplicates)
- Eingabeformat: vorberechnete Hashwerte durch Simhash-Funktion



Vorgehensweise

- Blockaufteilung
- Indizierung
- Identische Suche nach Indexschlüsseln
- Vergleich der Bits



Datengenerierung

- Query-Hashwerte: Zufällige Generierung
- Corpus-Hashwerte:
  - Duplikate: basieren auf Query-Hashwerten, wobei max. k Positionen zufällig invertiert werden
  - Übrige Hashwerte zufällig generiert
- Speicherung als Sequenz von Long-Werten in zwei Dateien

Implementierte Komponenten

- FileInputFormat und RecordReader
  - Einlesen der Daten
  - Aufteilung auf die Mapper
- Map
  - Erzeugung der Masken und Indizes
  - Finden der near-duplicates
- Reduce
  - Verschmelzen der Listen
  - Ausgabe

Finden von near-duplicates

```
    List<Long> pdups;
// run over all fingerprints

for(int i = 0; i < fingerprints.length; i++) {
    long fp = fingerprints[i];

    // run over all indices and check for equality for the set mask bits
    for(int j = 0; j < MASKS.length; j++) {
        pdups = indices.get(j).get(MASKS[j] & fp);
        if(pdups != null) {

            // check hamming distance criterion for possible near-duplicates
            for(int k = 0; k < pdups.size(); k++) {
                if(isNearDuplicate(pdups.get(k), fp)) {
                    List<Long> entries = map.get(pdups.get(k));
                    if(!entries.contains(fp)) {
                        entries.add(fp);
                    }
                    map.put(pdups.get(k), entries);
                }
            }
        }
    }
}
```

Finden von near-duplicates

```
public static boolean isNearDuplicate(long x, long y) {
long hamming_bits = x ^ y;

int diff = 0;
for(int i = 0; i < Long.SIZE; i++ ) {
    diff += hamming_bits & 1;
    hamming_bits >>>= 1;
    if(diff > MAX_DISTANCE) {
        return false;
    }
}

    return true;
}
```

Ausgabe

File: /user/toby/out/part-00000

Goto : /user/toby/out go

Go back to dir listing
Advanced view/download options

-7253127574651717233	-7253126475140092529	-7253127849529755250	-2641441556223280757
-5491288702235061828	-5491288702235324004	-6639706607214554692	-5491288702503496708
-5486317444252353411	-4909293741995508483	-5487144276996440963	-5486317444244161411
-4797612092326257425	-4815626490835739161	-4793099696605864705	-5085842468478223121
-4180025748777018677	-4180025748743456565	-4180025748772757813	-4184529485977560373
-3544617457957310509	-3544617440240570477	-3544615258866946095	-3544617457424633902
-3222778774478489740	-3222778774478490288	-3222778774461646988	-3223904622845724812
-1894294158393768186	-1894294158670723322	-1750178970319026426	-1930323024131684602
-1847393393625710773	-1802357397352006325	-1847393410806104241	-1847428577997801621
638059102511094 89	638059101839985 85	189906562378529905	12167273461384797 29
409528470880629098	409528473027064171	27153626840013005 86	-8813843565907562134
30705916831867940 97	19176701783115116 65	30705908585530732 49	764748673975196785
43252935372161366 32	43275431380065662 96	43252935201704933 68	38929479729885695 28
50424478850264696 17	50424468027618198 89	50424473406393648 49	50424478848923164 01
57529021650935576 64	57530429025808685 12	57348877322243414 08	57529027145809360 97
77248051934492525 38	77248051934492187 38	77248051935834620 74	77608339910051529 54
78777117394542454 29	79161330737752499 73	78777469581944612 37	78777469240947697 17
82573549931439991 54	85455809712489376 50	82753693916534811 40	82216076689542621 94
83883878013484997 26	83703734715584955 02	83883875264747881 10	89648485536519233 38
85592035006537291 83	91356642529571199 67	85592034320010991 67	85051778973113276 47

Nächste Schritte

- Korrektur des RecordReaders
- Performance-Analyse des Algorithmus
- Testen der Abhängigkeit von der Werteverteilung (u.a. ähnliche Hashwerte, Einfluss einer hohen Hamming Distance)
- Mögliche Erweiterungen:
  - Verteilung der Indizes auf verschiedene Nodes (mit anschließender Duplikatentfernung)
  - Verteilte Berechnung des Simhash-Wertes