Order Inversion

(K_1, K_2) $V_1, V_2, ..., V_n$

Composite key	Values
(K_1, K_{2a})	$\{A_1, A_2, ..., A_m\}$
(K_1, K_{2b})	$\{B_1, B_2, ..., B_p\}$
(K_1, K_{2c})	$\{C_1, C_2, ..., C_q\}$

- $m + p + q = n$
- The sort order is as follows:

  $K_{2a} < K_{2b} < K_{2c}$ (ascending order)

  or:

  $K_{2a} > K_{2b} > K_{2c}$ (descending order)

- $A_i, B_j, C_k \in \{V_1, V_2, ..., V_n\}$

$A_1, A_2, ..., A_m, B_1, B_2, ..., B_p, C_1, C_2, ..., C_q$

Key	Values as integer numbers
$(W, *)$	$A_1, A_2, ..., A_m$
(W, W_1)	$B_1, B_2, ..., B_p$
(W, W_2)	$C_1, C_2, ..., C_q$
...	...

Key	Values
`(dog, *)`	`100, 129, 500, ...`
`(dog, bite)`	`2, 1, 1, 1`
`(dog, best)`	`1, 1, 1, 1, 1`
`...`	`...`

example

```
# hadoop fs -cat /order_inversion/input/sample_input.txt
java is a great language
java is a programming language
java is green fun language
java is great
programming with java is fun
```

```
(great, a)        0.2
(great, is)        0.4
(great, java)        0.2
(great, language)        0.2
(with, is)        0.3333333333333333
(with, java)        0.3333333333333333
(with, programming)        0.3333333333333333
(fun, green)        0.2
(fun, is)        0.4
(fun, java)        0.2
(fun, language)        0.2
(programming, a)        0.2
(programming, is)        0.2
(programming, java)        0.2
(programming, language)        0.2
(programming, with)        0.2
```

구현

Table 5-7. Implementation classes in Hadoop

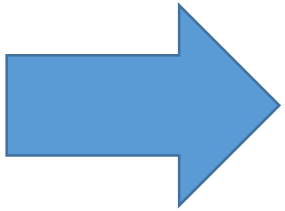
Class name	Class description
`RelativeFrequencyDriver`	Driver to submit job
`RelativeFrequencyMapper`	Defines `map()`
`RelativeFrequencyReducer`	Defines `reduce()`
`RelativeFrequencyCombiner`	Defines `combine()`
`OrderInversionPartitioner`	Custom partitioner: how to partition natural keys
`PairOfWords`	Represents pair of words (*Word1*, *Word2*)

Custom partitioner

- Natural key에 해당하는 모든 값은 같은 reducer로 전송되어야 함.
- {(man, tall), (man, strong), (man, moon), ...}

mapper

w1 w2 w3 w4 w5 w6



Key	Value
(w1, w2)	1
(w1, w3)	1
(w1, *)	2
(w2, w1)	1
(w2, w3)	1
(w2, w4)	1
(w2, *)	3
(w3, w1)	1
(w3, w2)	1
(w3, w4)	1
(w3, w5)	1
(w3, *)	4

Key	Value
(w4, w2)	1
(w4, w3)	1
(w4, w5)	1
(w4, w6)	1
(w4, *)	4
(w5, w3)	1
(w5, w4)	1
(w5, w6)	1
(w5, *)	3
(w6, w4)	1
(w6, w5)	1
(w6, *)	2

reducer

Key	Value
(w1, *),(w1, w2),(w1, w3)	2,1,1
(w2, *),(w2, w1),(w2, w3),(w2, w4)	3,1,1,1
(w3, *),(w3, w1),(w3, w2),(w3, w4),(w3, w5)	4,1,1,1,1
(w4, *),(w4, w2),(w4, w3),(w4, w5),(w4, w6)	4,1,1,1,1
(w5, *),(w5, w3),(w5, w4),(w5, w6)	3,1,1,1
(w6, *),(w6, w4),(w6, w5)	2,1,1