Types

A1

c1,SUM(c2)

A1

c1,SUM(c2)

c1,SUM(c2)

pipelined

materialized

Tables

|  |  |
| --- | --- |
| PK | col1 |
| x1 | 3 |
| x2 | 3 |

|  |  |
| --- | --- |
| PK | col1 |
| x1 | 5 |
| x2 | 2 |

|  |  |
| --- | --- |
| PK | col1 |
| x1 | 30 |
| x2 | 17 |

|  |  |  |
| --- | --- | --- |
| PK | col1 | col2 |
| k1 | x1 | 5 |
| k2 | x1 | 12 |
| k3 | x2 | 7 |
| k4 | x2 | 8 |
| k5 | x2 | 2 |
| k6 | x1 | 13 |

Type 1(materialized, delta input)

|  |  |  |
| --- | --- | --- |
| Seq | Op | Tuples |
| 1 | put | (x1,15) |
| 2 | put | (x2,8) |
| 3 | delete | (x2) |
|  | put | (x1,23) |
| 4 | put | (x1,18) |
| 5 | put | (x1,10) |
| 6 | put | (x1,38) |
| 7 | put | (x1,10) |
| 8 | delete | (x1) |
|  | put | (x3,22) |

Output stream

|  |  |  |
| --- | --- | --- |
| Seq | Op | Tuples |
| 1 | get, put | (x1,15) |
| 2 | get, put | (x2,8) |
| 3 | get, delete | (x2) |
|  | get, put | (x1,23) |
| 4 | get, put | (x1,18) |
| 5 | get, put | (x1,10) |
| 6 | get, put | (x1,38) |
| 7 | get, put | (x1,10) |
| 8 | get, delete | (x1) |
|  | get, put | (x3,22) |

|  |  |  |
| --- | --- | --- |
| Seq | Op | Tuples |
| 1 | put | (k1,(**null**, **null)**, (x1,15)) |
| 2 | put | (k2,(**null**,**null)**,(x2,8)) |
| 3 | put | (k2,(x2,8), (x1,8)) |
| 4 | put | (k1, (x1,15), (x1,10)) |
| 5 | put | (k2, (x1,8**)**, (**null**,**null)**) |
| 6 | put | (k3,(**null**,**null**), (x1,28)) |
| 7 | put | (k3, (x1,28**)**, (**null,null)**) |
| 8 | put | (k1, (x1,10), (x3,22)) |

delta input

A1

c1,SUM(c2)

Internal

Execution

Input stream

Type 1(cost)

**Example(**100 operations**):**

40 inserts

40 updates

20 deletes

G = 40 + 40 + 20 = 100

P = 40 + 40 + 10 + 15 = 105

D = 5 + 10 = 15

F = 100

Cost = 100 \* 0,1 + 100 \* 0,8 + 105 \* 0,5 + 15 + 0,5 = 150

Delete(NOTZERO) = delete(k1, x1, 15) **causes** update(x1, 25🡪10)

Delete(ZERO) = delete(k1, x1, 25) **causes** delete(x1, 25)

Update(AGGKEY;NOTZERO) = update(k1, x1🡪x2, 25) **causes** update(x1,45🡪20)

Update(AGGKEY;ZERO) = update(k1, x1🡪x2, 25) **causes** delete(x1,25)

G = inserts + updates + deletes

P = inserts + updates + updates(AGGKEY;NOTZERO) + deletes(NOTZERO)

D = deletes(ZERO) + updates(AGGKEY;ZERO)

Cost = G\* weight(G) + P\*weight(P) + D \* weight(D) + F\* weight(F)

c2 <= 10; [ST=0,5]

A1

100 F

Type 2(materialized, preagg input)

|  |  |  |
| --- | --- | --- |
| Seq | Op | Tuples |
| 1 | put | (x1,15) |
| 2 | put | (x2,8) |
| 3 | delete | (x2) |
|  | put | (x1,23) |
| 4 | put | (x1,18) |
| 5 | put | (x1,10) |
| 6 | put | (x1,38) |
| 7 | put | (x1,10) |
| 8 | delete | (x1) |
|  | put | (x3,22) |

Output stream

|  |  |  |
| --- | --- | --- |
| Seq | Op | Tuples |
| 1 | put | (x1,15) |
| 2 | put | (x2,8) |
| 3 | delete | (x2) |
|  | put | (x1,23) |
| 4 | put | (x1,18) |
| 5 | put | (x1,10) |
| 6 | put | (x1,38) |
| 7 | put | (x1,10) |
| 8 | delete | (x1) |
|  | put | (x3,22) |

preagg input

A1

c1,SUM(c2)

Internal

Execution

|  |  |  |
| --- | --- | --- |
| Seq | Op | Tuples |
| 1 | put | (x1,(k1,15)) |
| 2 | put | (x2,(k2,8)) |
| 3 | put | (x2,**null**) |
|  | put | (x1,(k1,15),(k2,8)) |
| 4 | put | (x1,(k1,10),(k2,8)) |
| 5 | put | (x1,(k1,10)) |
| 6 | put | (x1,(k1,10),(k3,28)) |
| 7 | put | (x1,(k1,10)) |
| 8 | put | (x1, **null**)) |
|  | put | (x3,(k1,22)) |

Cost Calculations:

Input stream

**Example(**100 operations**):**

40 inserts

40 updates

20 deletes

G = 40 + 40 + 20 = 100

P = 40 + 40 + 10 + 15 = 105

D = 5 + 10 = 15

F = 100

Cost = 100 \* 0,1 + 105 \* 0,5 + 15 + 0,5 = 70

P = inserts + updates + updates(AGGKEY;NOTZERO) + deletes(NOTZERO)

D = deletes(ZERO) + updates(AGGKEY;ZERO)

Cost = P\*weight(P) + D \* weight(D) + F\* weight(F)

Type 3(pipelined)

|  |  |  |
| --- | --- | --- |
| Seq | Op | Tuples |
| 1 | put | (x1,(k1,15)(sum,15)) |
| 2 | put | (x2,(k2,8),(sum,8)) |
| 3 | put | (x2,(sum,0)) |
|  | put | (x1,(k1,15),(k2,8),(sum,23)) |
| 4 | put | (x1,(k1,10),(k2,8),(sum,18)) |
| 5 | put | (x1,(k1,10),(sum,10)) |
| 6 | put | (x1,(k1,10),(k3,28),(sum,38)) |
| 7 | put | (x1,(k1,10),(sum,10)) |
| 8 | put | (x1, (sum,0))) |
|  | put | (x3,(k1,22),(sum,22)) |

Output stream

c1,SUM(c2)

pipelined

Internal

Execution

GROUP BY c1

|  |  |  |
| --- | --- | --- |
| Seq | Op | Tuples |
| 1 | put | (x1,(k1,15)) |
| 2 | put | (x2,(k2,8)) |
| 3 | put | (x2,**null**) |
|  | put | (x1,(k1,15),(k2,8)) |
| 4 | put | (x1,(k1,10),(k2,8)) |
| 5 | put | (x1,(k1,10)) |
| 6 | put | (x1,(k1,10),(k3,28)) |
| 7 | put | (x1,(k1,10)) |
| 8 | put | (x1, **null**) |
|  |  | (x3,(k1,22)) |

Puts = 0

Deletes = 0

Input stream

Cost Calculations:

G = inserts + updates + deletes

P = inserts + updates + updates(AGGKEY;NOTZERO) + deletes(NOTZERO)

D = deletes(ZERO) + updates(AGGKEY;ZERO)

Cost = G\* weight(G) + P\*weight(P) + D \* weight(D) + F\* weight(F)

Flow diagram

M1

A

V1

C1

150 GP

S1

c1,SUM(c2)

300 GP

100 F

100 F

100 GP

100 GP

100 F

100 F

A

(c1,c2,c3,c4)

A

S1

C1

M1

c1,SUM(c2)

c1,MIN(c2)

c1,COUNT(c2)

Using Preaggregation

S1

S1

C1

M1

S1

C1

300 P

c1,SUM(c2)

c1,MIN(c2)

c1,COUNT(c2)

100 F

100 GP

c1, PKA, c2

100 F

A

(c1,c2,c3,c4)

100 GP

100 F

A

Merging

V2

V3

V1

SMC1

V2

100 GP

A

100 F

100 GP

100 F

V3

V1

M1

C1

S1

c1,SUM(c2)

c1,MIN(c2)

c1,COUNT(c2)

c1,SUM(c2) ∧ c1,MIN(c2) ∧ c1,COUNT(c2)

(c1,c2,c3,c4)

A