University of Zurich[UZH]

Planning Ahead: Stream-Driven Linked-Data Access under Update-Budget Constraints

Shen Gao, Daniele Dell'Aglio, Soheila Dehghanzadeh,
Abraham Bernstein, Emanuele Della Valle, Alessandra Mileo

University of Zurich[UZH]

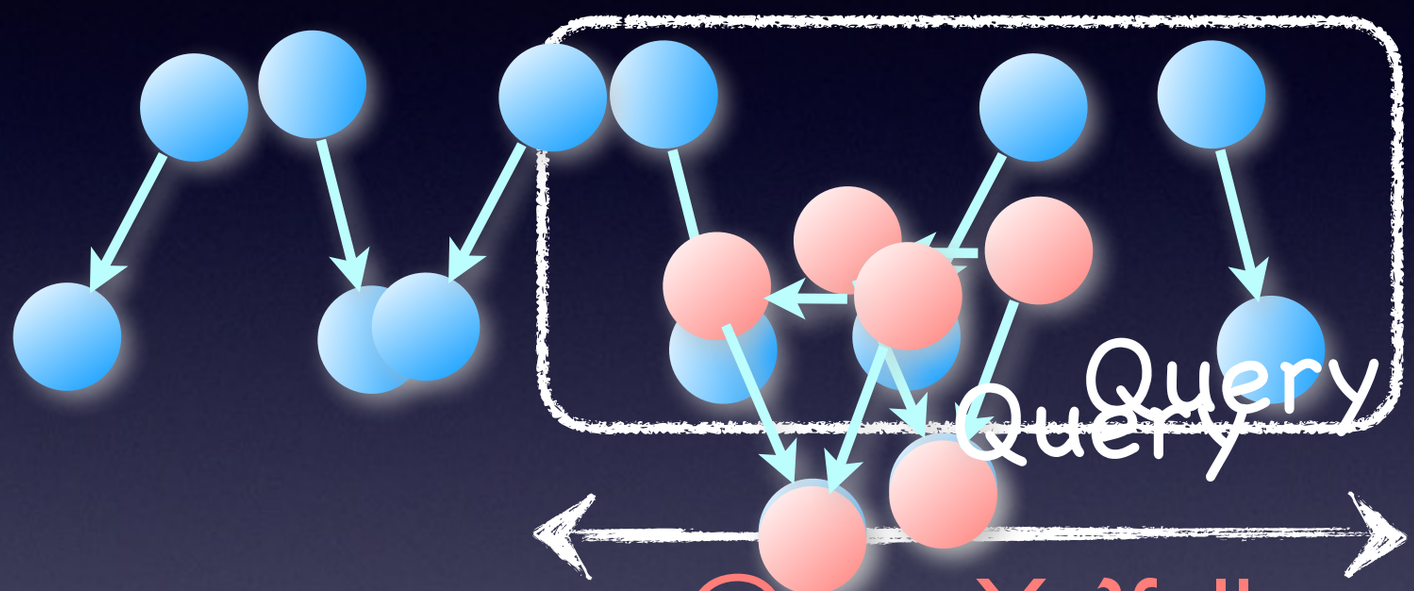
POLITECNICO DI MILANO

Insight

How can we efficiently access SPARQL endpoints in RDF *Stream* Processing (*RSP*)?

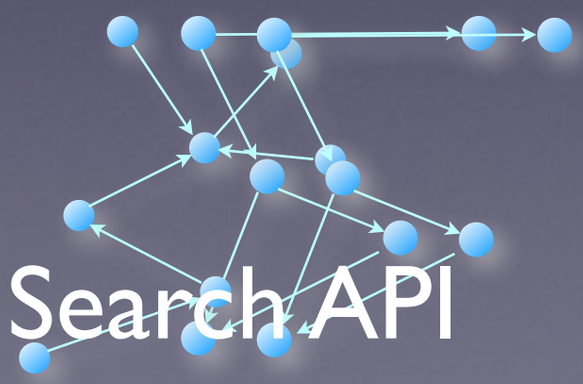
Finding Trendy Hashtags

@user1: #ISWC2016



Stream API

Join



Search API

. . . . . .

Accessing Background Knowledge is Costly

Local Cache?

Budget: Num. of accesses per window

How can we allocate budget efficiently?

- Model the query: Bipartite Graph
- Exploiting the bipartite graph for solutions

SELECT * WHERE
WINDOW (*Stream*) {P^W}
SERVICE (*BKG*) {P^S}

P^W

P^S

Stream

Local cache

s_1

s_2

s_3

B_1

B_2

B_3

?

?

?

BKG

Budget = 2

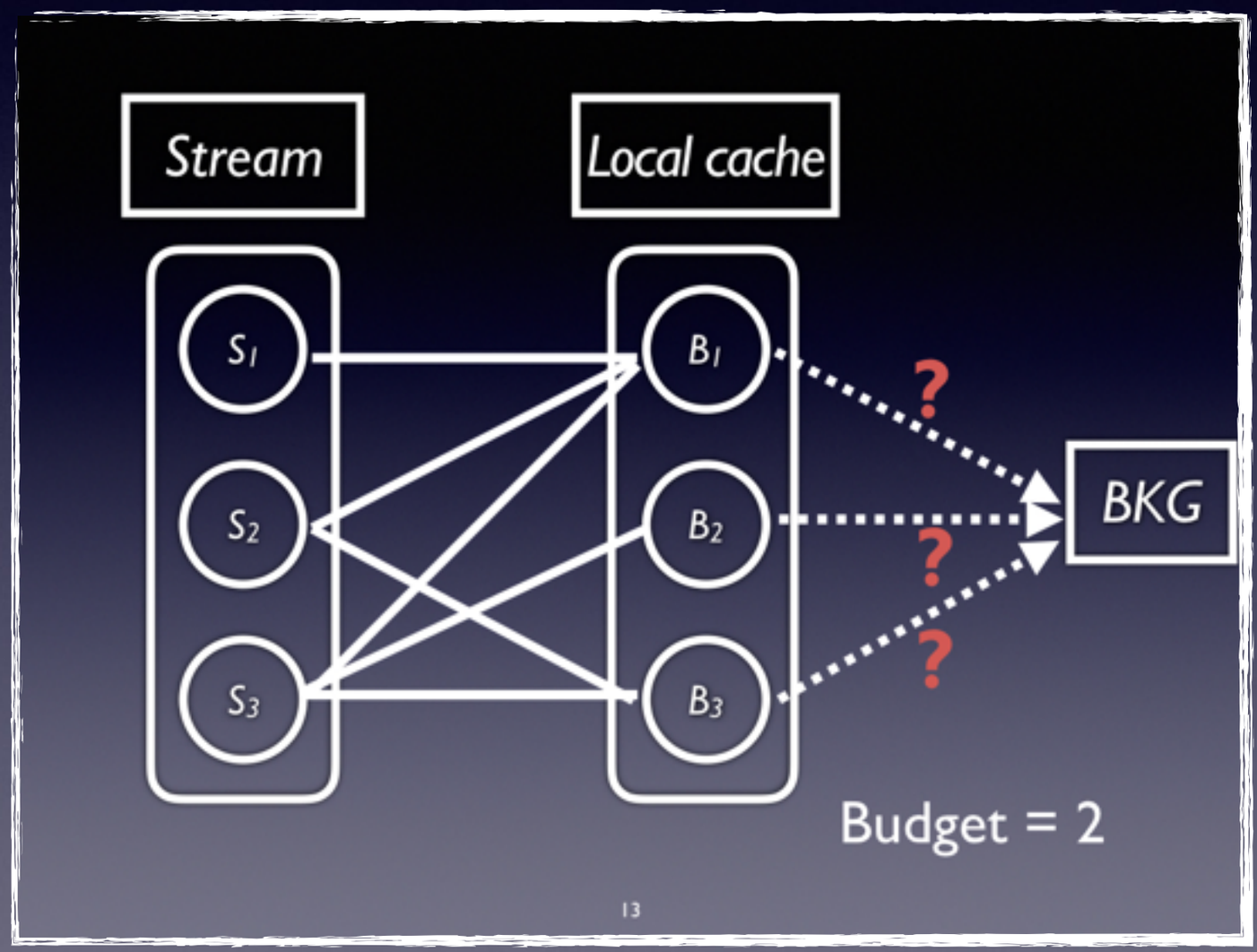
What is the optimisation goal?

```
SELECT *   WHERE
           WINDOW (Stream) {P^W}
           SERVICE  (BKG) {P^S}
```

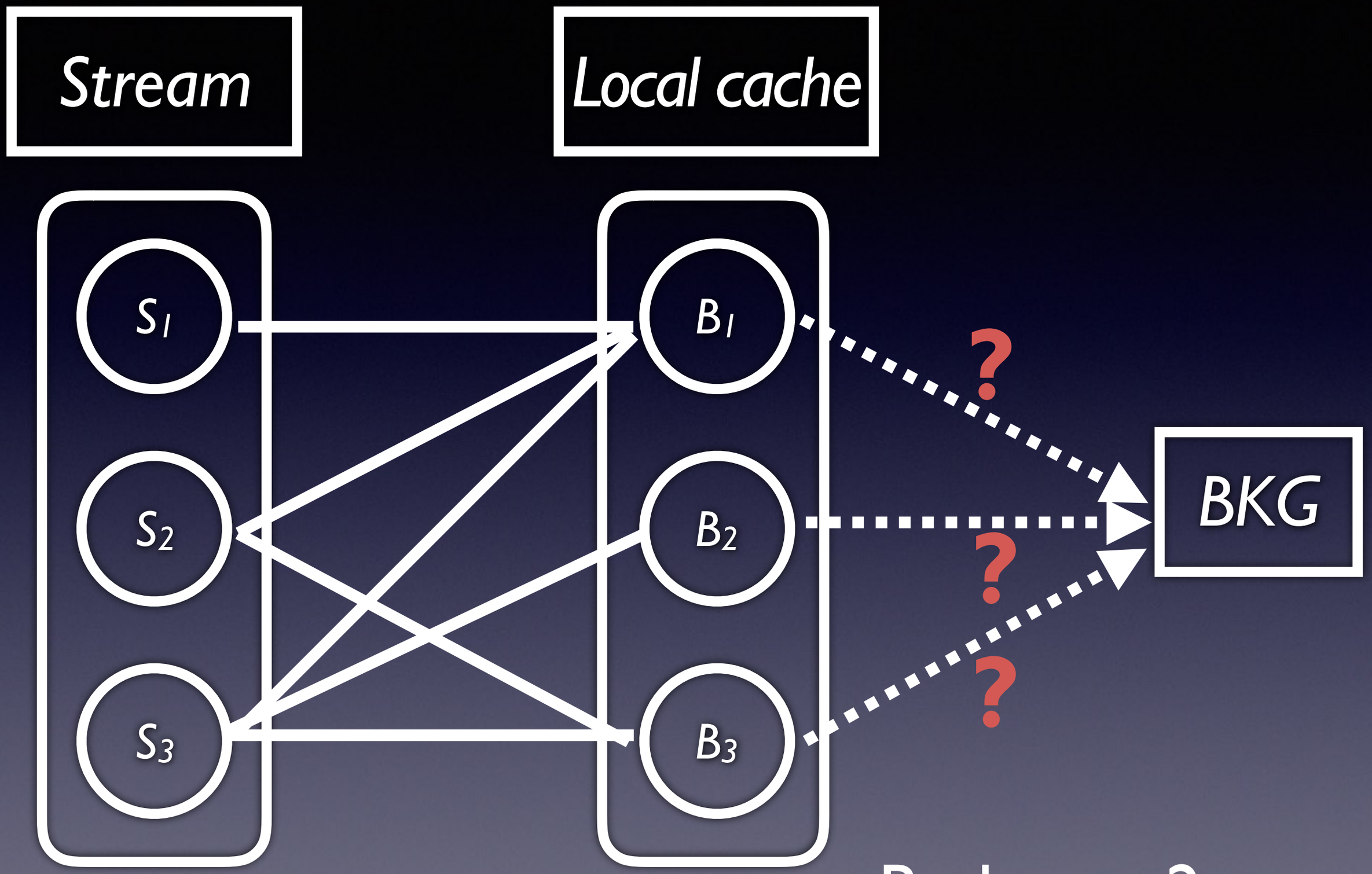
- SERVICE clause P^S:
  - Basic Graph Pattern (BGP)
  - Aggregate query

Service Clause P^S - BGP

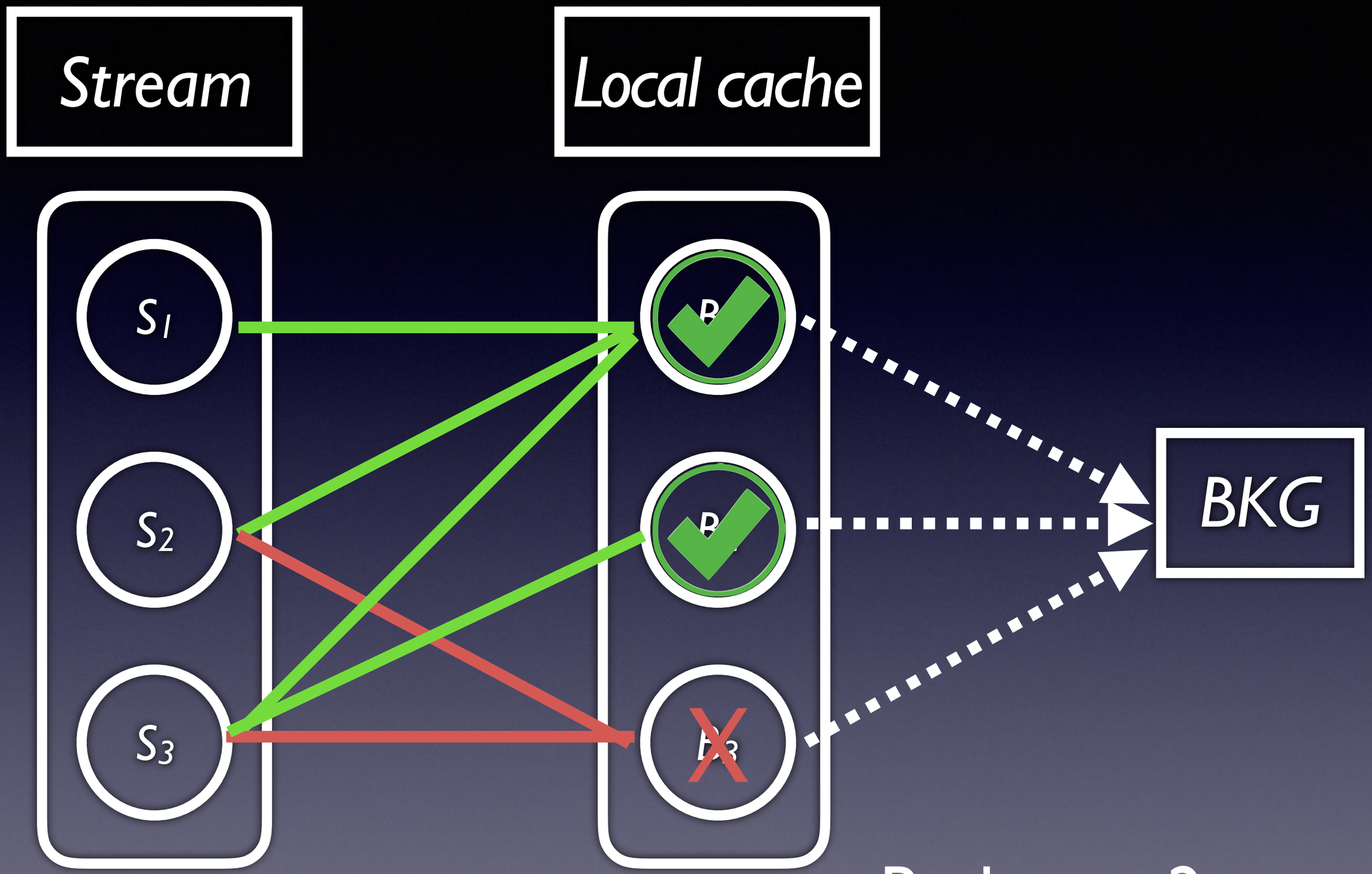
<S_1 :p B_1>
<S_2 :p B_1>
<S_2 :p B_3>
<S_3 :p B_1>
<S_3 :p B_2>
<S_3 :p B_3>



Max(# of edges)

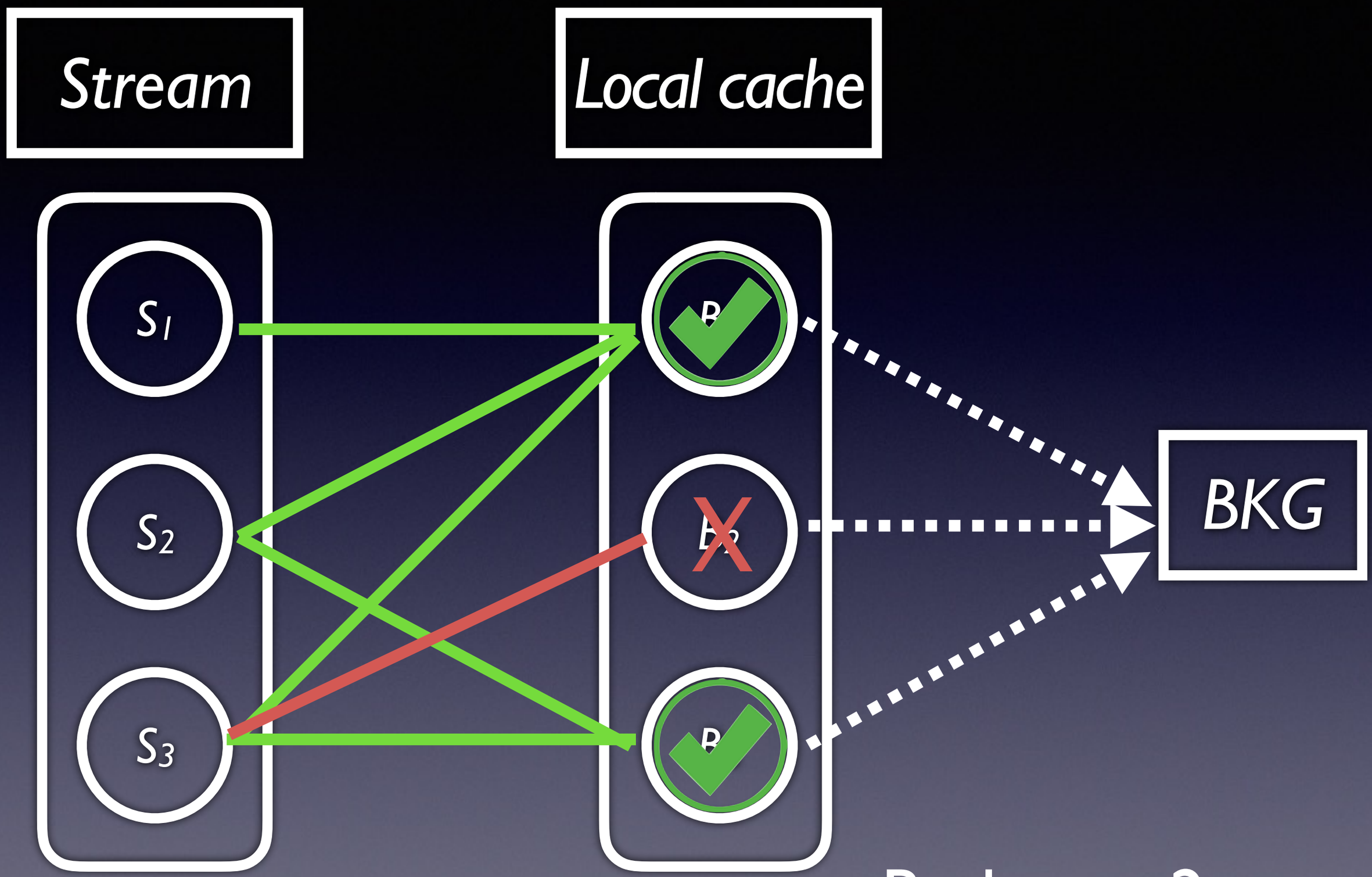


Budget = 2



Budget = 2

# of Fresh Results = 4



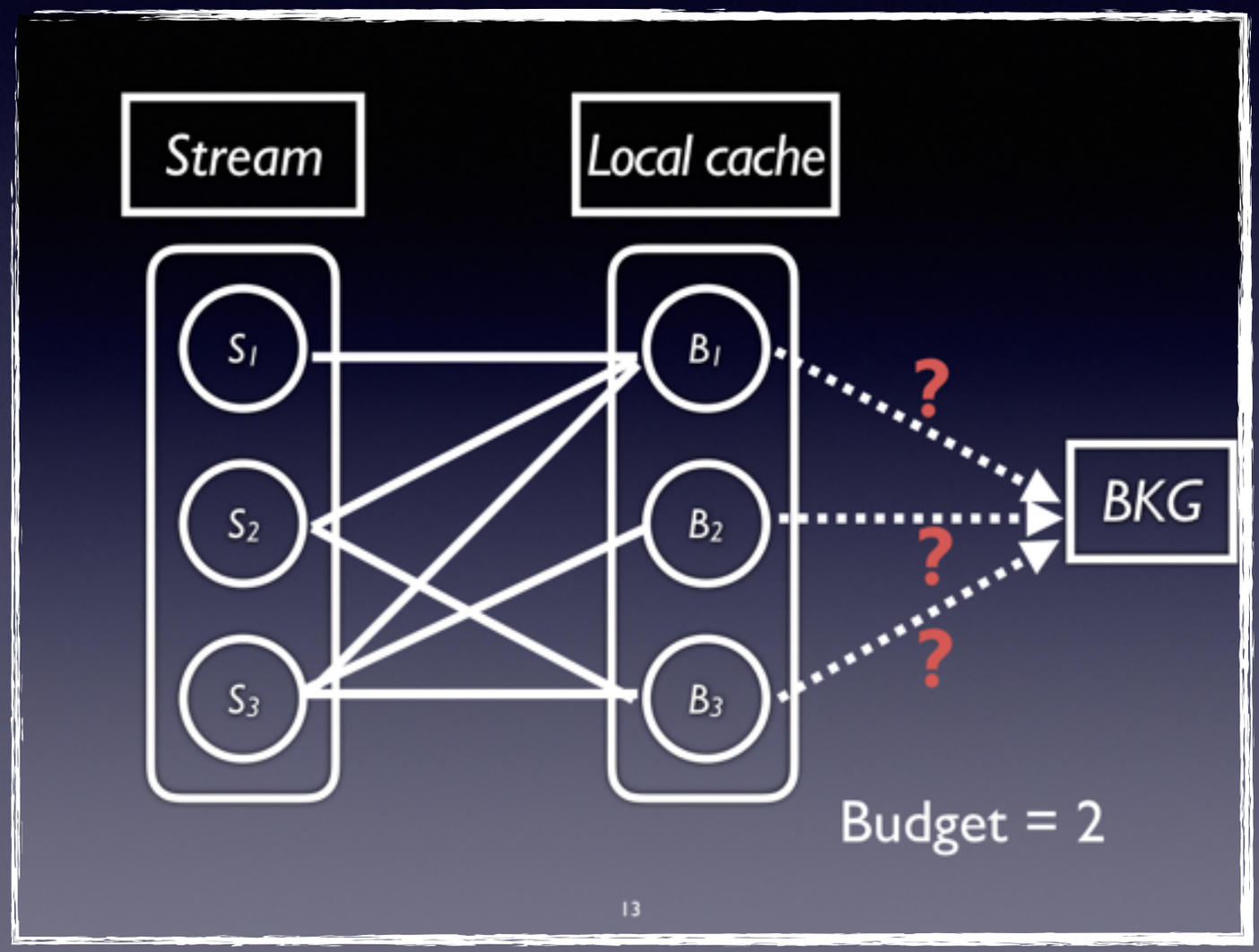
$SBM_{BGP} = Max(deg(B_i))$

Budget = 2

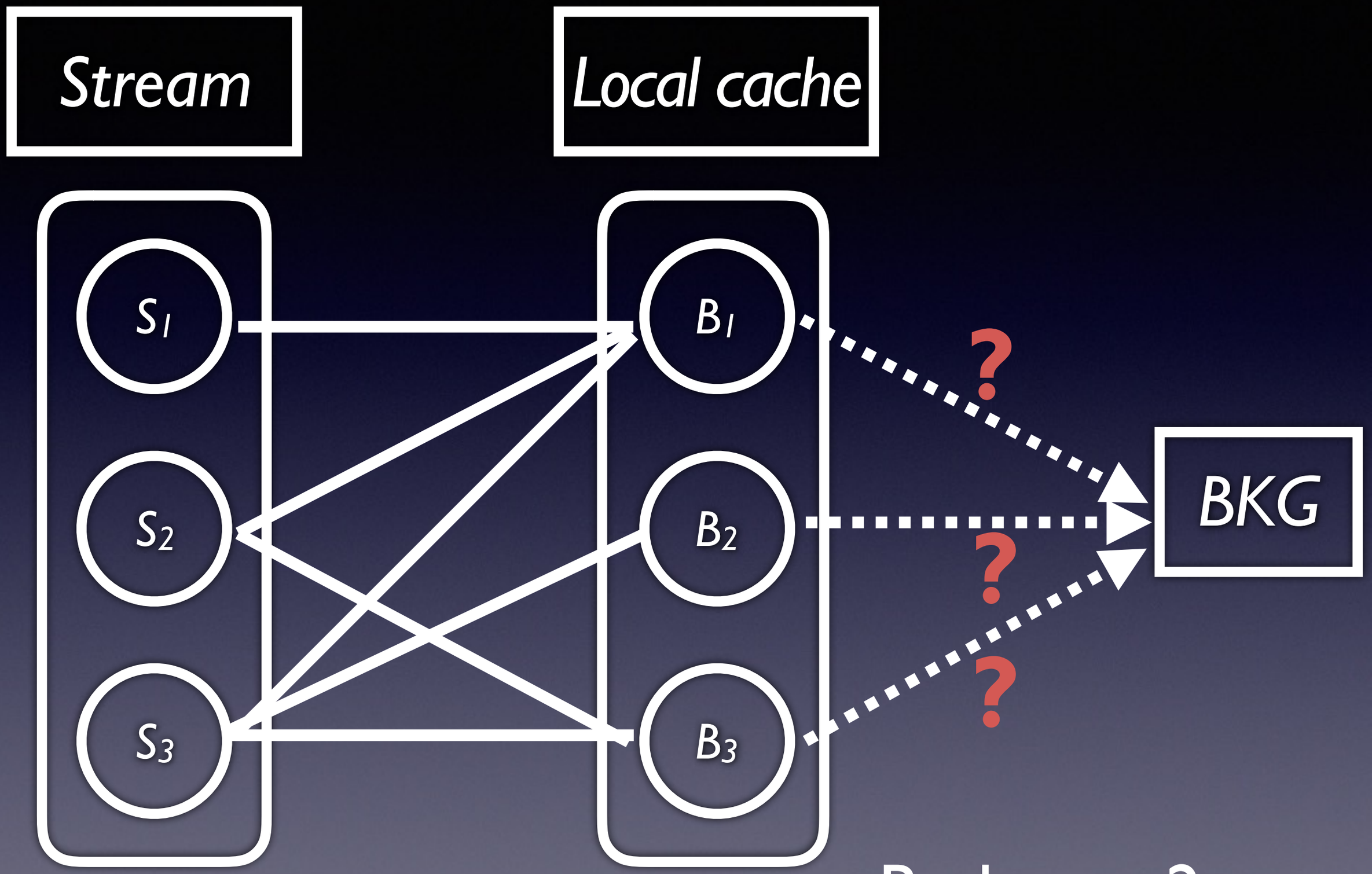
# of Fresh Results = 5

Service Clause P^S - Agg.

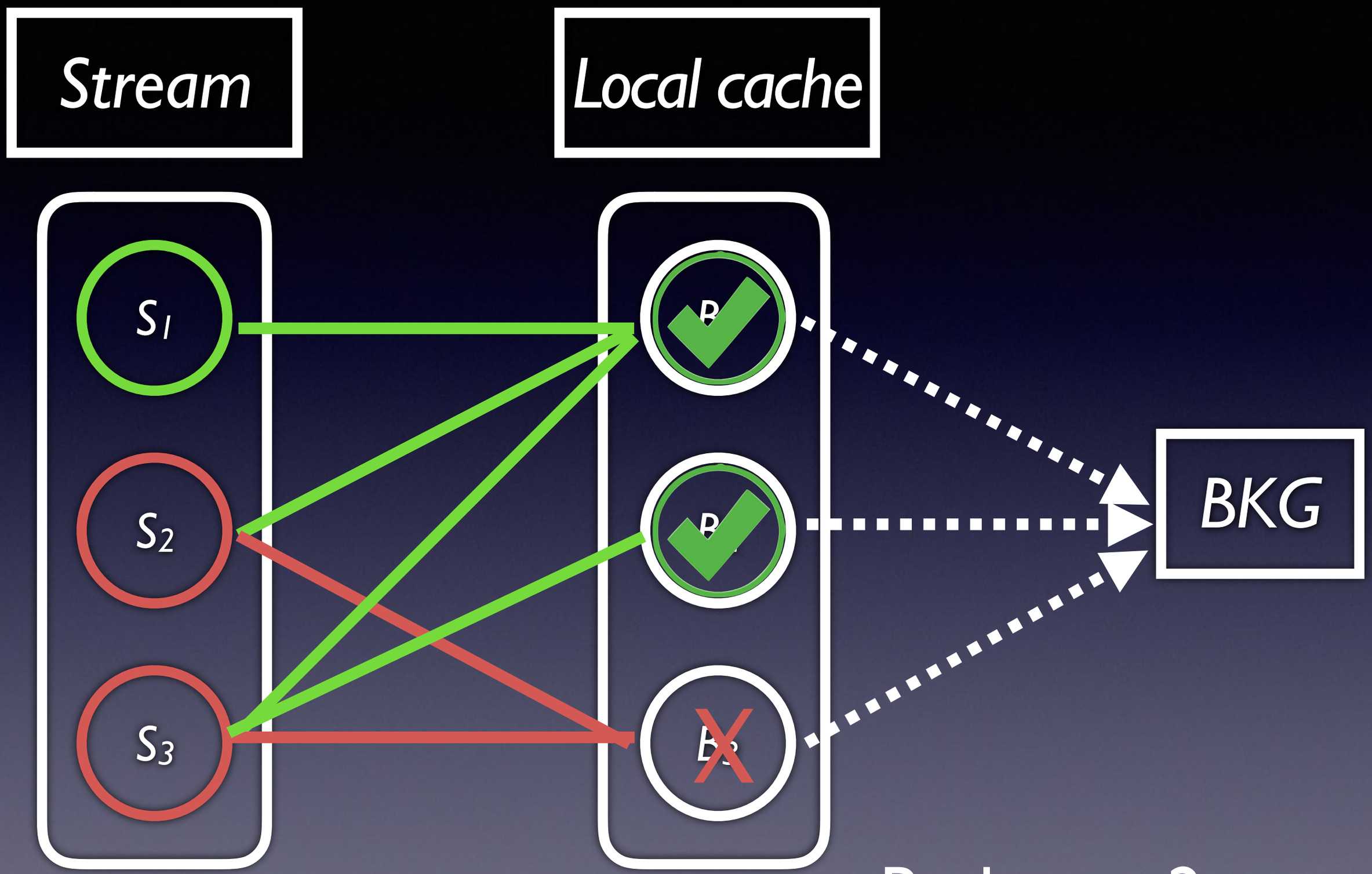
<S_1 :p Sum(B_1)>
<S_2 :p Sum(B_1,B_3)>
<S_3 :p Sum(B_1, B_2,B_3)>



Max(# of S_i, whose join partners are all fresh)

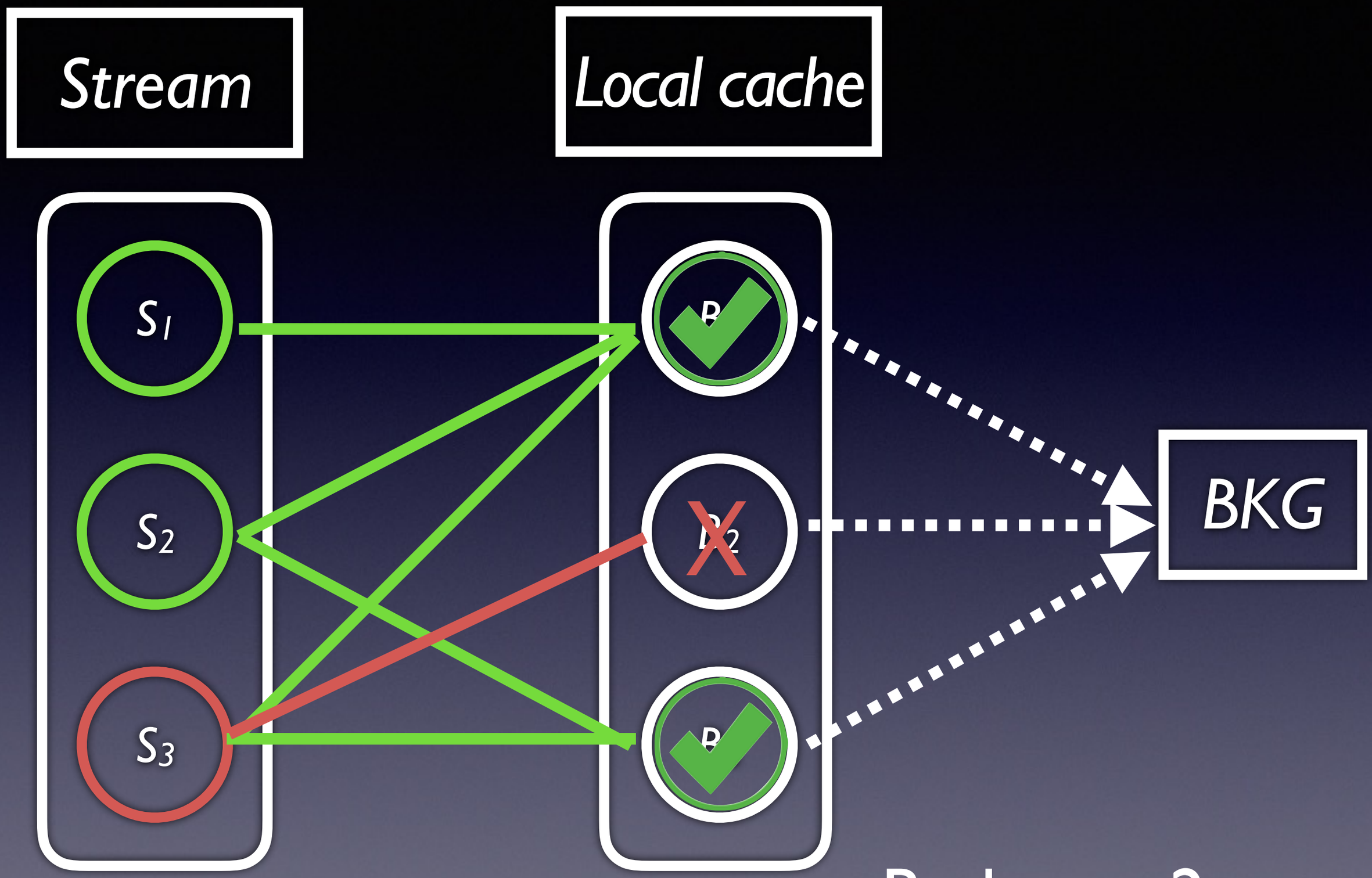


Budget = 2



Budget = 2

# of Fresh Results = 1



NP-Hard

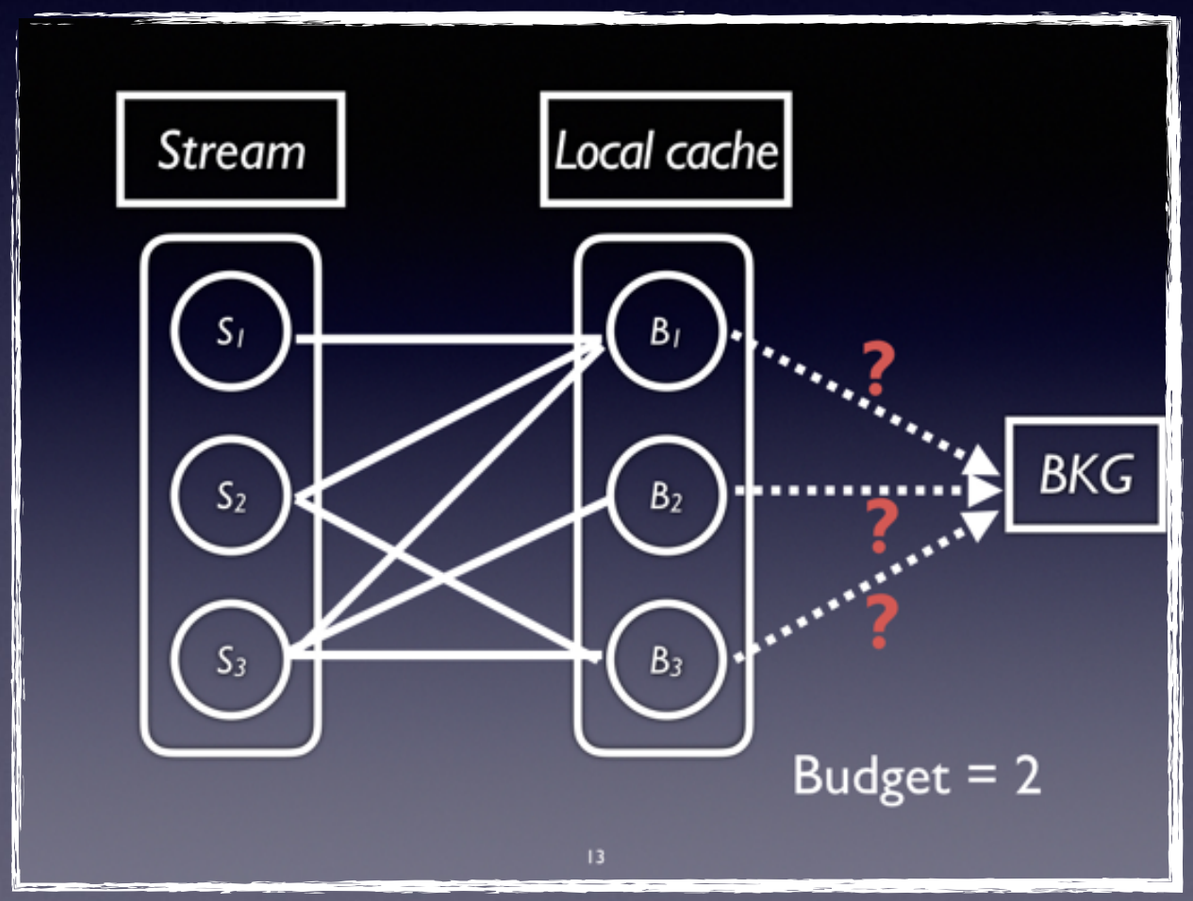
SBM_{agg} = B_i of Min(deg(S_i))

Budget = 2

# of Fresh Results = 2

SBM:

Selectivity Based Maintenance



BGP considers deg(B_i)

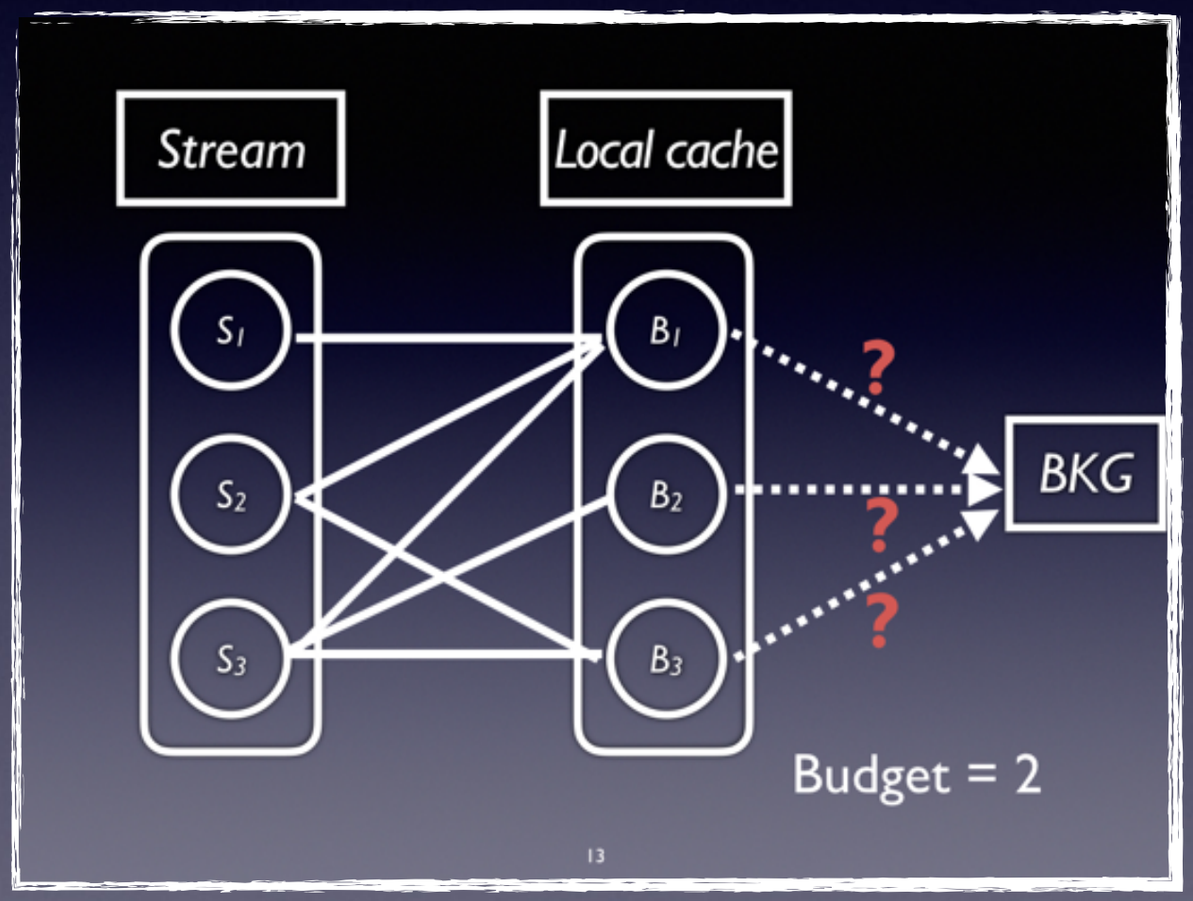
Agg. considers deg(S_i)

in the

current window

IBM:

Impact Based Maintenance



in the
current window

+

in the
following windows

Stream

Local cache

S_1

S_2

S_3

B_1

B_2

B_3

?

?

?

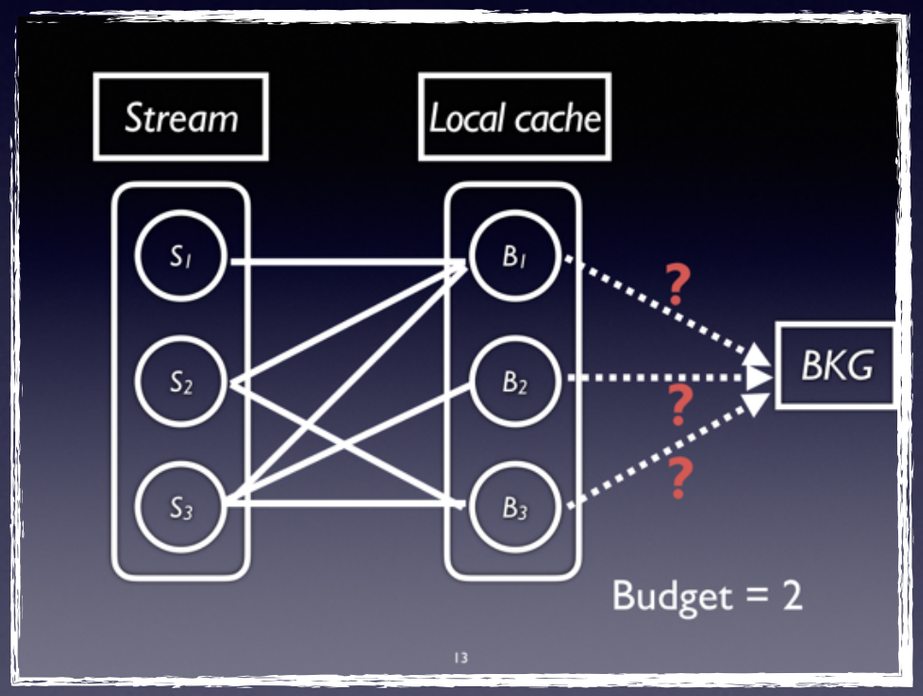
BKG

Win +1

Win +2

Impact = # of results
in current + future windows

FBA:
Flexible Budget Allocation



the budget

in the
current window

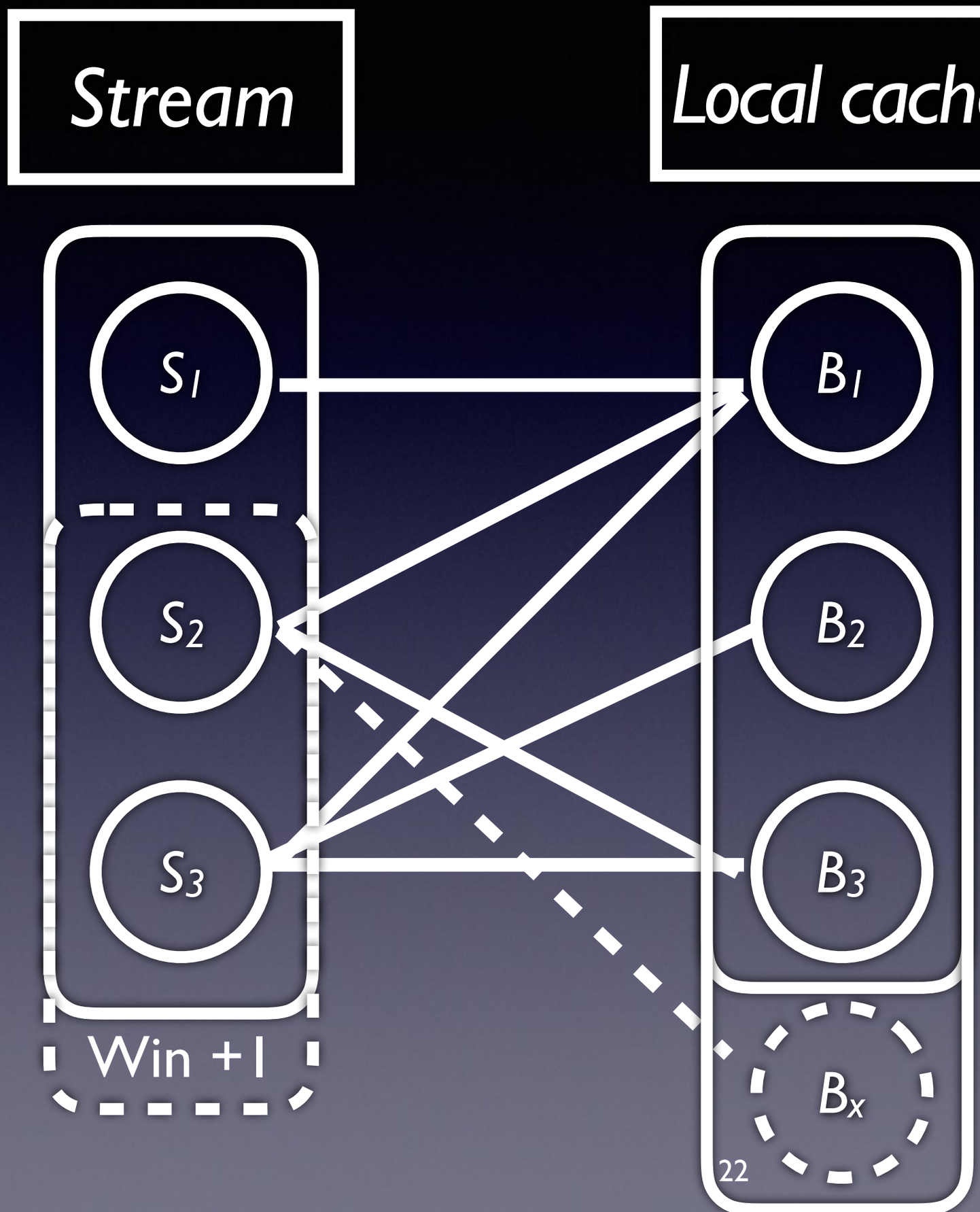
+

in the
following windows

Fixed Budget = 2

Fixed Budget = 2

Fixed Budget = 2


Stream
Local cache
S_1
S_2
S_3
Win +1
B_1
B_2
B_3
B_x

Fixed Budget = 2 - 1 = 1

Fixed Budget = 2 +1 = 3

How much budget should be saved for future windows?

How can we allocate the budget better?

```
SELECT *    WHERE
            WINDOW (Stream) {P^W}
            SERVICE  (BKG) {P^S}
```

Spend the budget on
the ones that have
the largest impact over the windows!

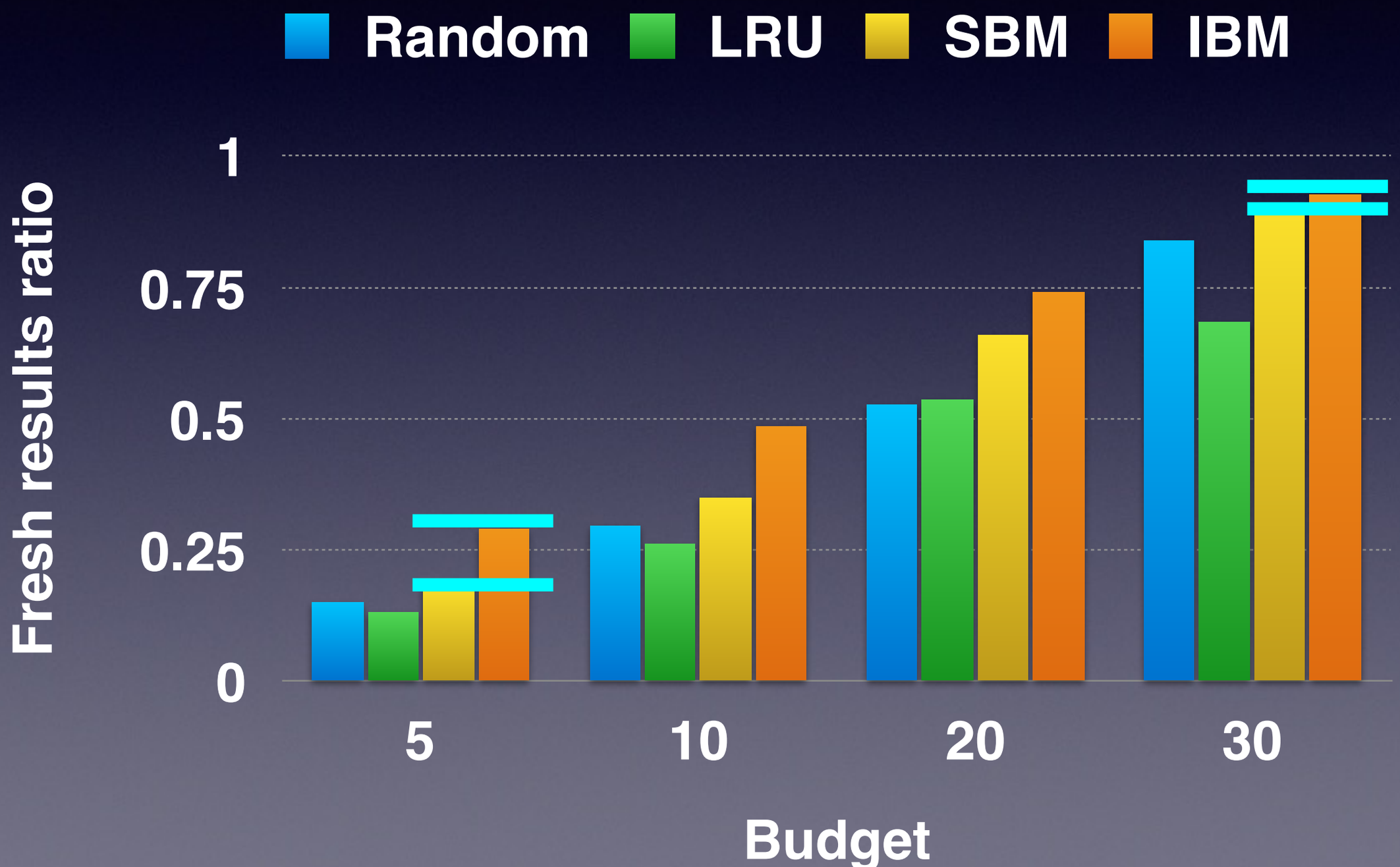
SBM => IBM => FBA

Experiment setup

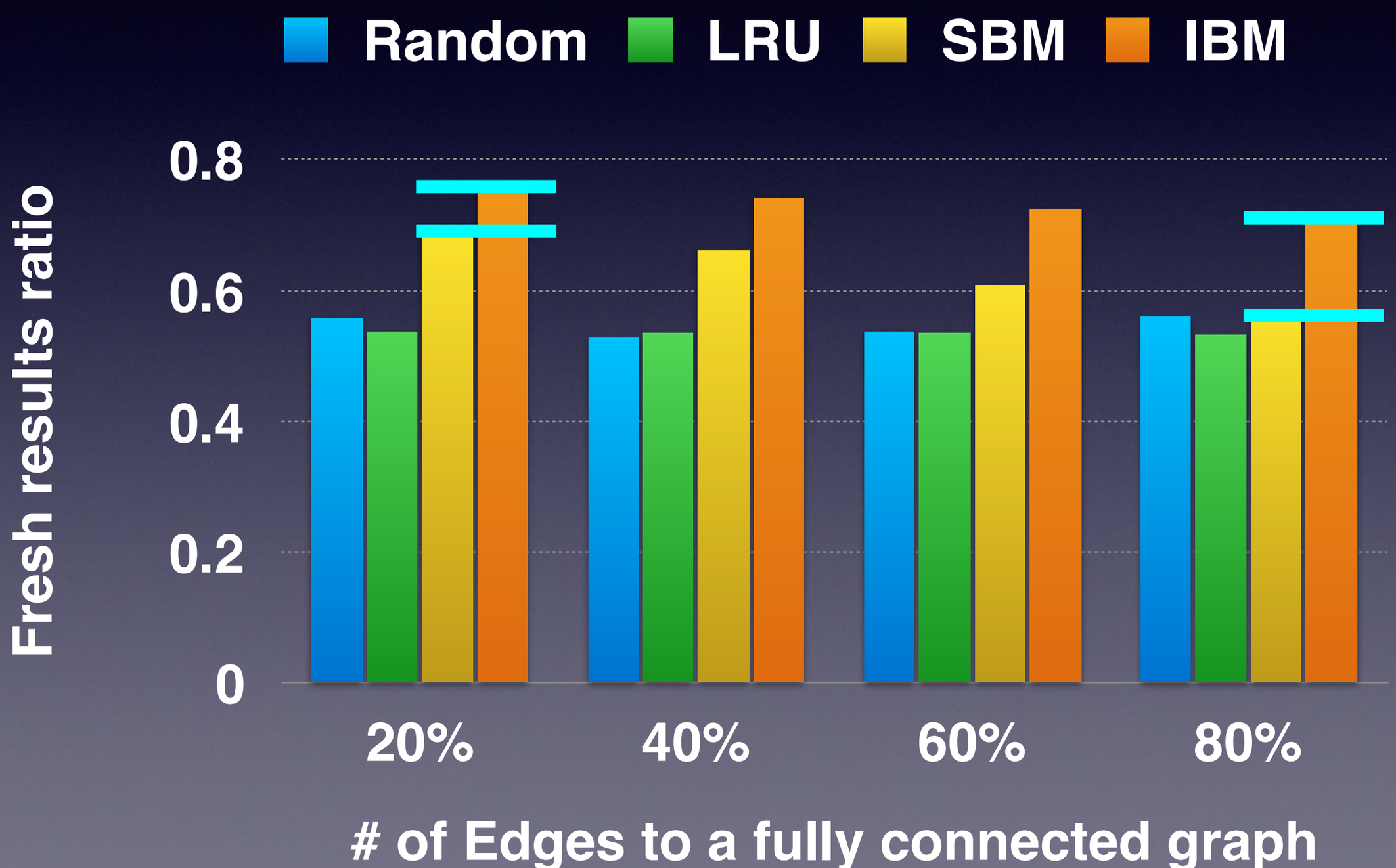
- Real implementation in C-SPARQL :
  - https://github.com/dellaglio/CSPARQL-engine

- Synthetic and Real data

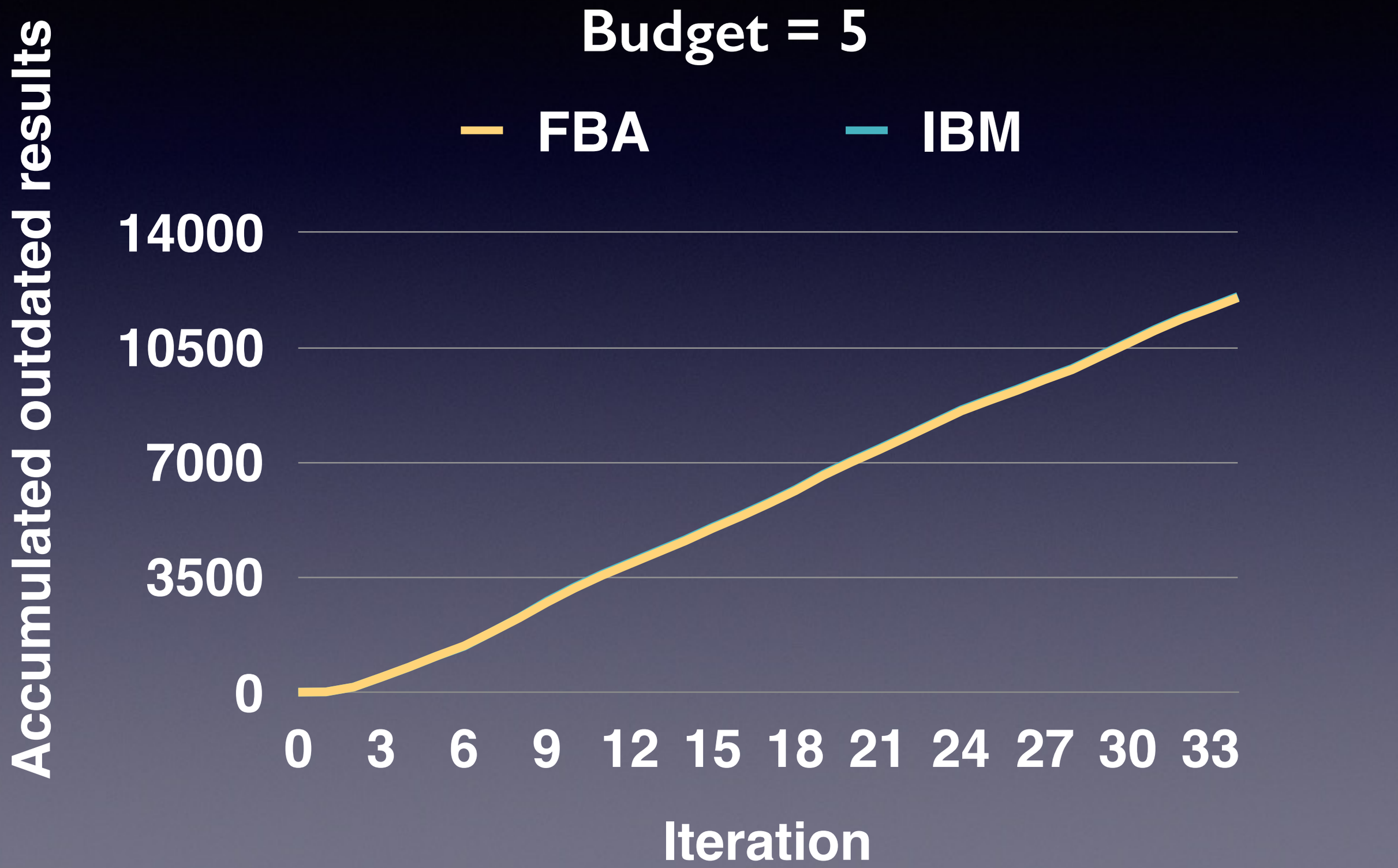
Increasing Budget



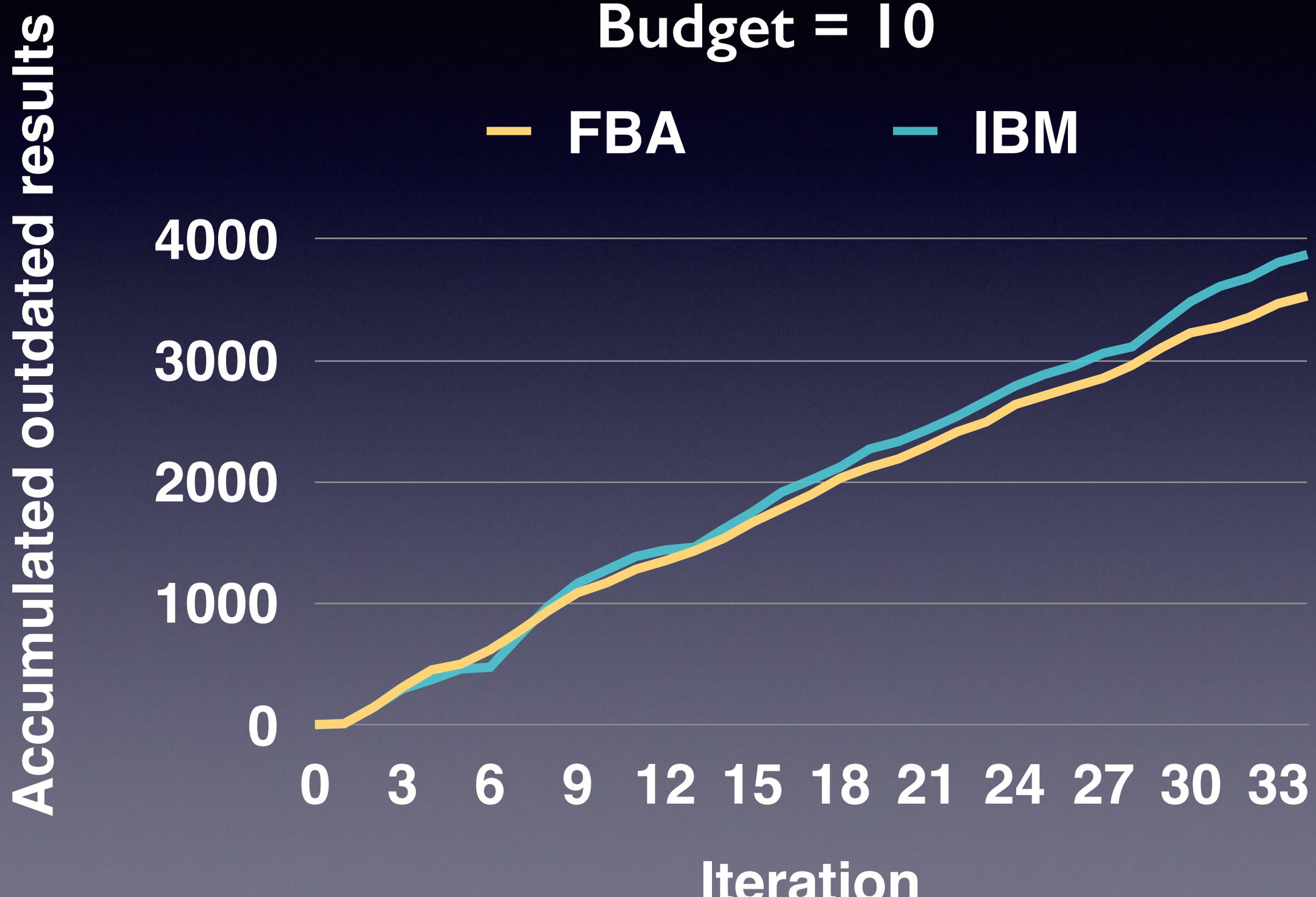
Increasing Graph Density



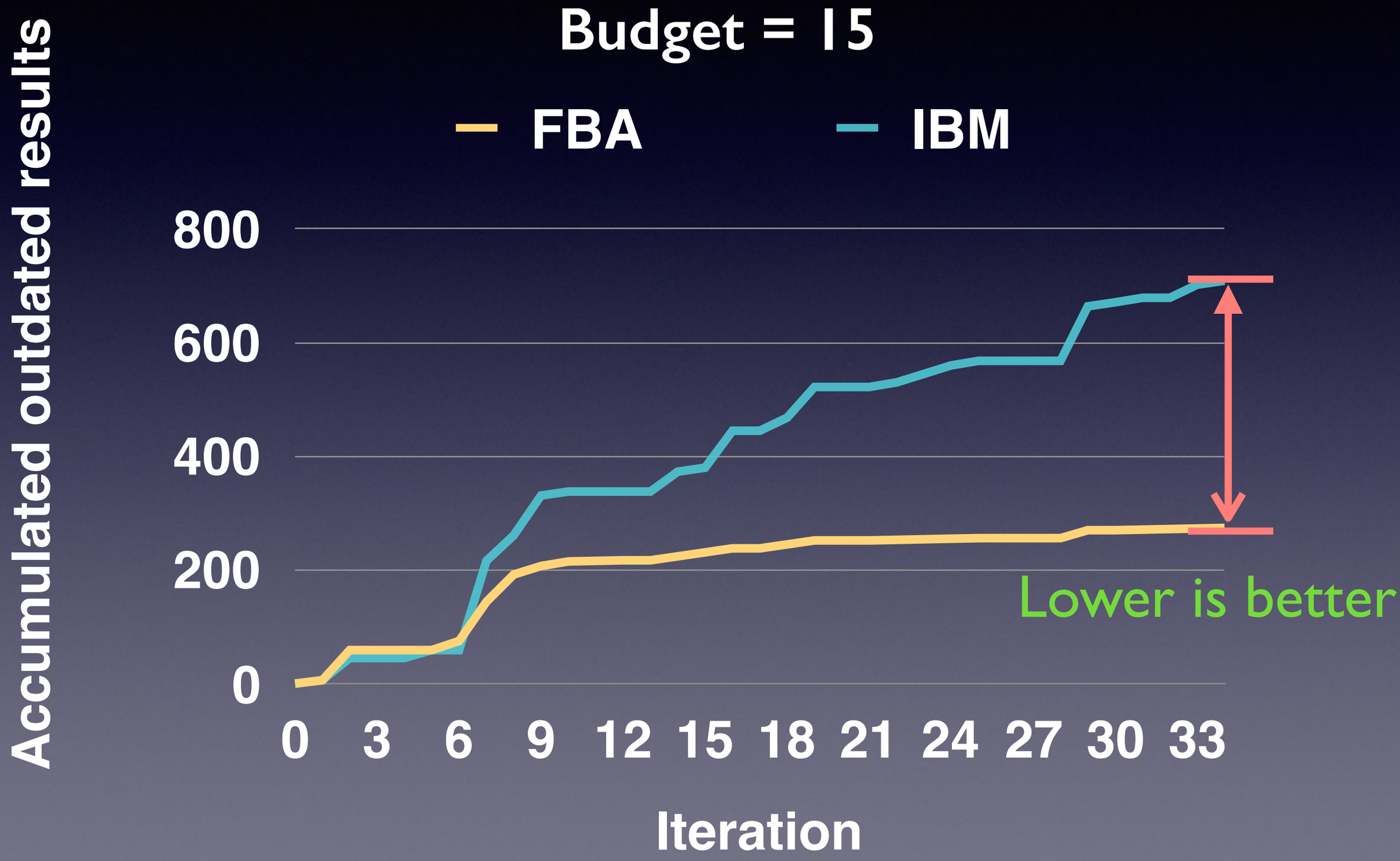
Increasing Budget



Increasing Budget



Increasing Budget



Stream

Local c

S_1

S_2

S_3

B_1

B_2

B_3

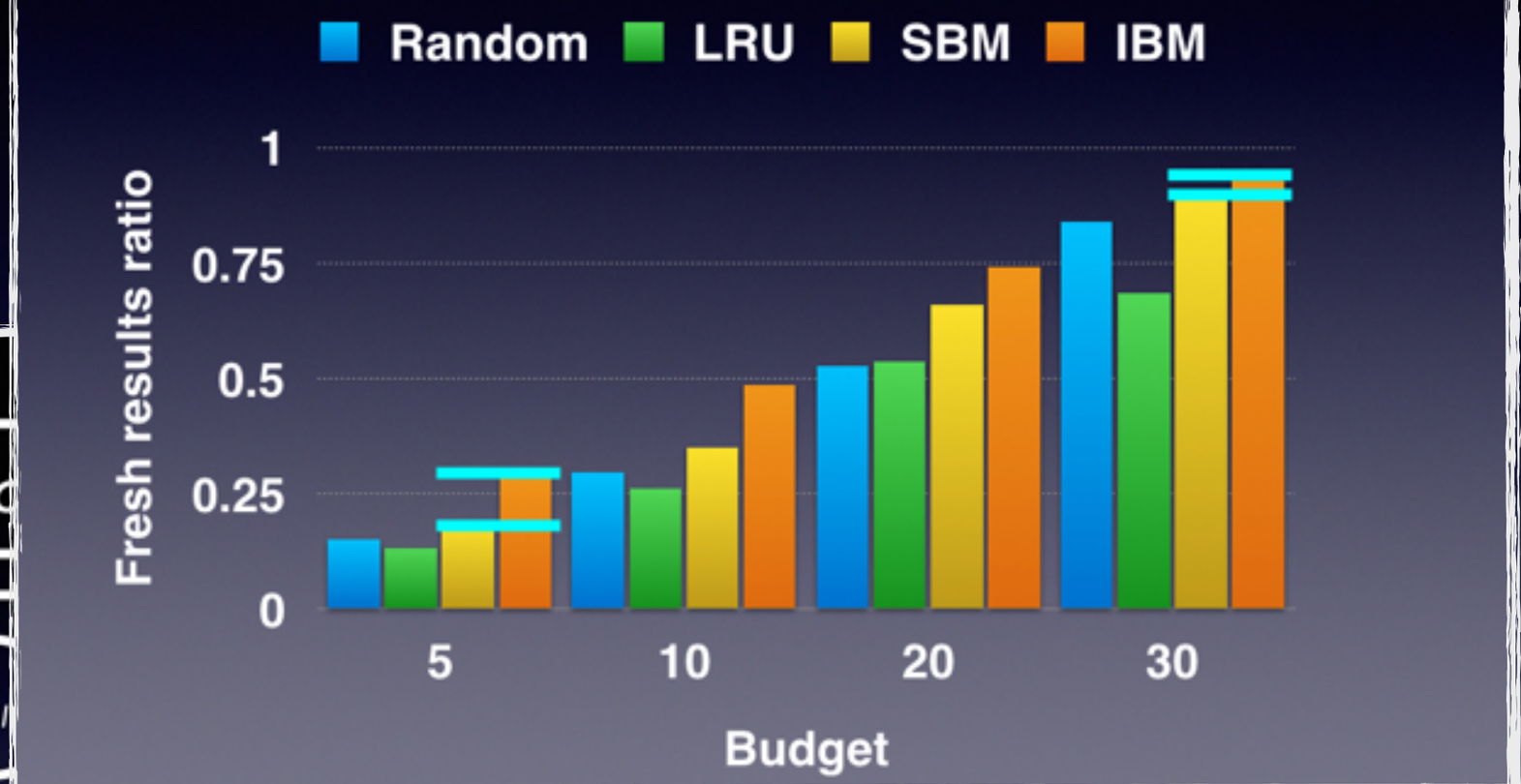
BKG

?

?

Budget = 2

Increasing Budget



Thank you very much!

Questions?

University of Zurich[UZH]