HEM: A Hardware-aware Event Matching Algorithm for Content-based Pub/Sub Systems

Wanghua Shi, Shiyou Qian*

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Outline

1 Background

- **Content-based publish/subscribe system**
  - **composed of publishers, subscribers, and brokers.**
  - **on-demand data distribution.**
  - **realizes the decoupling of time, space and synchronization of communication parties.**
- **Event Matching Algorithm**
  - **find all matching subscriptions for each event as soon as possible.**

2 Problem--Matching Semantics

- **Event: attribute-value pairs.**
- **Subscription: predicates/constraints.**
- **Predicate/Constraint: closed interval defined in an attribute.**

$E_1=\{(a_1, 3), (a_3, 7)\}$

$S_1=\{(a_1, [1,5]), (a_2, [5,8])\}$

$S_2=\{(a_1, [3,3]), (a_3, [1,10])\}$

E_1 **matches** S_2

Size of a subscription $\leq$ Size of an event $\leq$ Dimension

3 Motivation

- The **essence** of **backward marking** based event matching algorithm
  1. searches and marks all the unmatching subscriptions in each attribute
  2. unmarked subscriptions are the matches of the event
- The **inherent defects** of backward marking
  1. tends to mark the same unmatching subscription for multiple times → redundant operations
  2. heavily dependent on dimension

Fig. 1: The matching time distribution of REIN

3 Motivation

- Discovery
  - efficient OR operation on bitset with millions of bits
  - large memory capacity → trade space for time
- Breakthrough
  1. inserting: pre-mark subscriptions in bitsets according to their predicates' value distribution
  2. matching: select bitsets to execute several OR operations instead of millions of assignment operations

SubID: 1 2 3 4 5

(00110)

OR op

(10011)

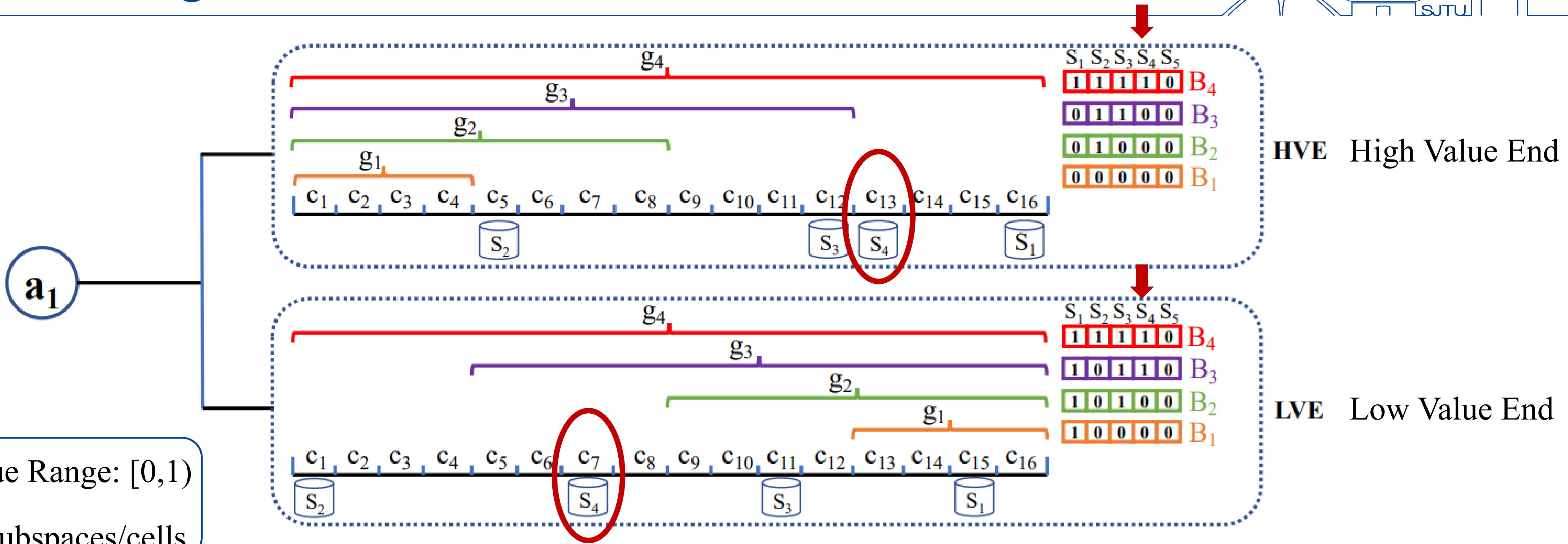
⇩

(10111)

↓

S_2 is the unique matches.

4 Design--Insertion



Value Range: [0,1)

16 subspaces/cells

Table 1: Sample subscriptions

ID	a_1	a_2	ID	a_1	a_2	ID	a_1	a_2
S_1	[0.9, 0.95]	[0.8, 0.9]	S_2	[0.0, 0.3]	[0.5, 0.7]	S_3	[0.63, 0.69]	[0.1, 0.2]
S_4	[0.38, 0.76]	-	S_5	-	[0.4, 0.57]	-		

S_4 : (a_1,[0.38,0.76])

LVE: $\lfloor 0.38 * 16 + 1 \rfloor = 7$

HVE: $\lfloor 0.76 * 16 + 1 \rfloor = 13$

4 Design—Matching without bitsets

- Event $E_2 = \{(a1, 0.48)\} \rightarrow c_8$
- High Value End

E_2

Unmatching HVE cell list

$\underbrace{c_1 \; c_2 \; c_3 \; c_4 \; c_5 \; c_6 \; c_7}$ | $c_8 \; c_9 \; c_{10} \; c_{11} \; c_{12} \; c_{13} \; c_{14} \; c_{15} \; c_{16}$

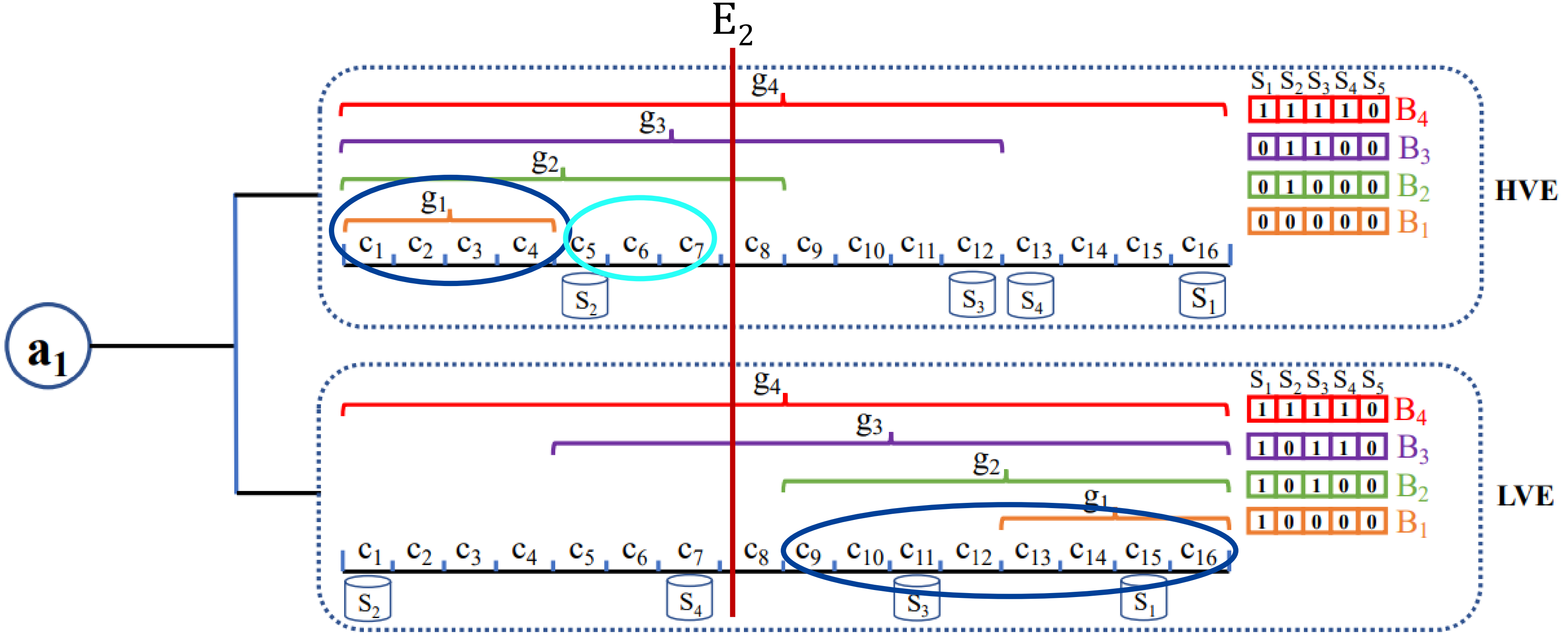
- Low Value End

Unmatching LVE cell list

$c_1 \; c_2 \; c_3 \; c_4 \; c_5 \; c_6 \; c_7$ | $c_8 \; \underbrace{c_9 \; c_{10} \; c_{11} \; c_{12} \; c_{13} \; c_{14} \; c_{15} \; c_{16}}$

4 Design—Matching with bitsets

- Event $\mathrm{E}_2 = \{(a1, 0.48)\} \rightarrow c_8$



SubID: 1 2 3 4 5

$HVE\ B_1(00000)$

OR op

$LVE\ B_2(10100)$

⇩

(10100)

⇩ Mark op

(11100)

S_4 and S_5 are the matches.

The number of marking cells is 3. (Cell Number/Group Number -1)

4 Design—Analysis

Theorem 1. *Given* n, ψ_E, ψ_S, d *and* g*, when* $\psi_E = d, g > 1$ *and the predicate values are uniformly distributed in the value domain* $[0, 1]$*, the improvement ratio of the marking time of HEM is* $1 - \frac{1}{g}$ *with the SPC method.*

Group Number $g = 32 \rightarrow 96.875\%$

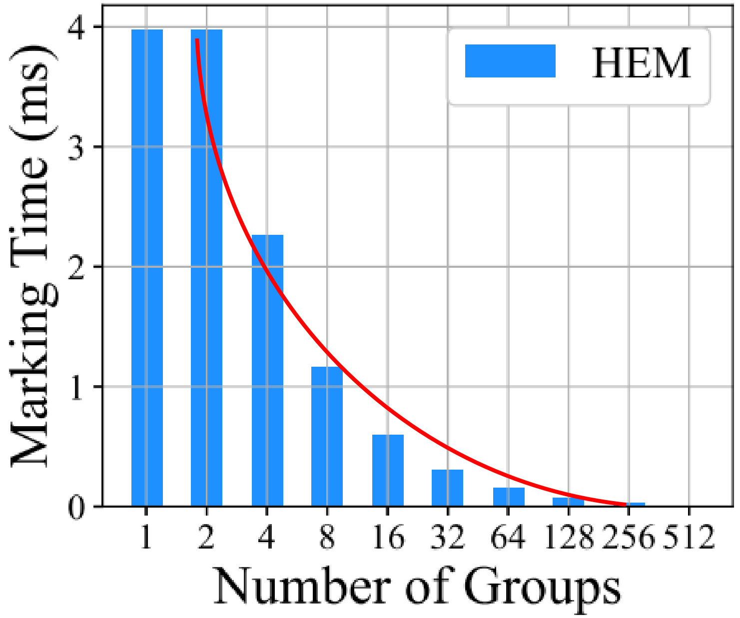
5 Evaluation--Setting

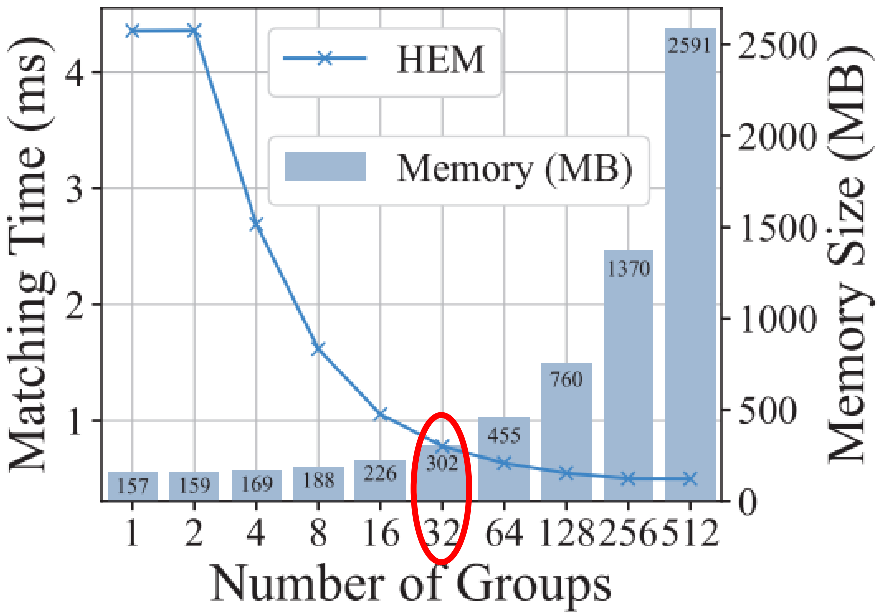
- Language: C++ / g++9.3.0 -O3
- Baseline: REIN, TAMA, Ada-REIN, OpIndex
- Testbed:
  - Ubuntu 20.04
  - AMD 3.7 GHz
  - 64GB RAM
- Codes: **https://github.com/shiwanghua/HEM**

Table 2: Parameter settings in the experiments

Name	Description	Experimental Values
$\mathcal{R}$	The value domain of attribute.	$[1, 1M]$
α	Parameter of Zipf.	**0**, 1 $\sim$ 5
d	Number of attributes.	**20**, 30, 100, 300 $\sim$ 900
n	Number of subscriptions.	0.3M, **1M**, 3M $\sim$ 9M
ψ_E	Event Size.	**20**, 30 $\sim$ 80
ψ_S	Subscription size.	5, **10**, 15 $\sim$ 30
w	Predicate width.	0.1, 0.2, **0.3** $\sim$ 0.9

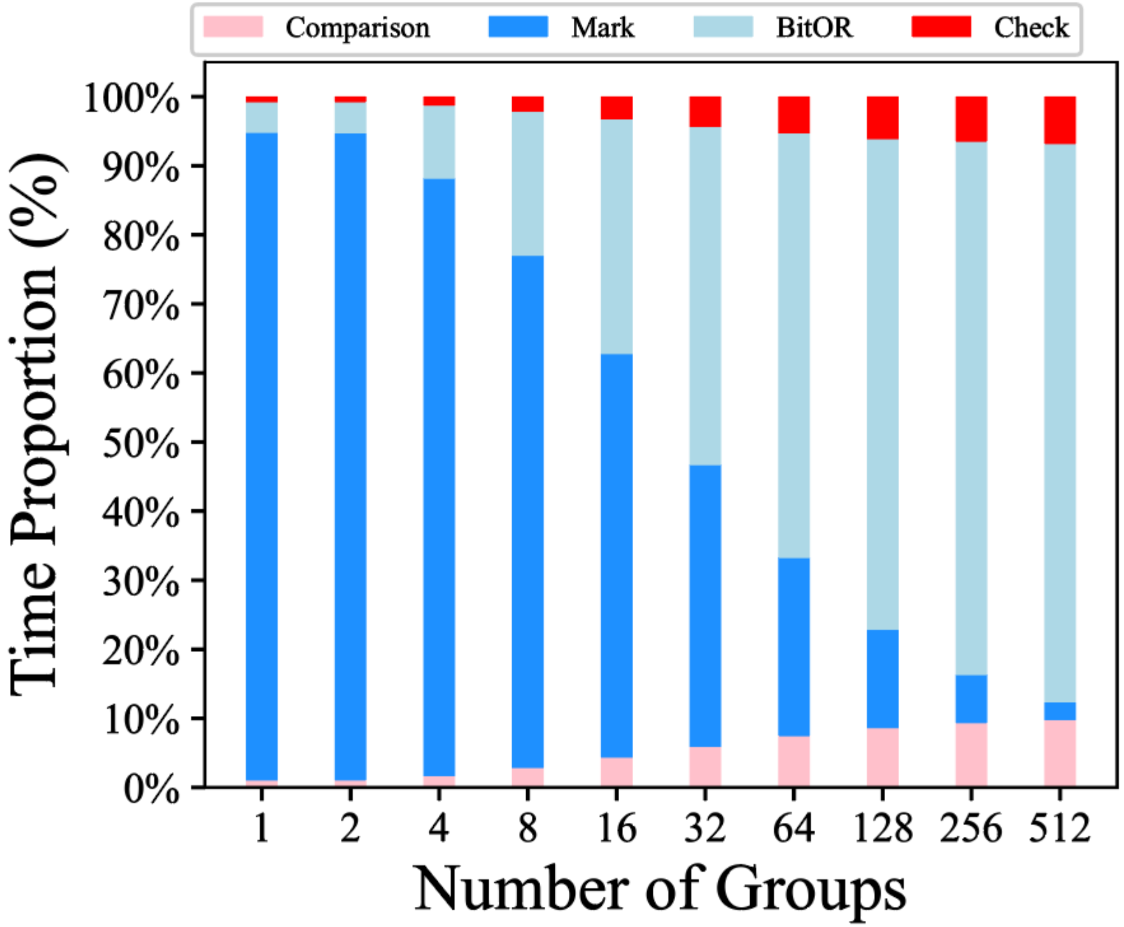
5 Evaluation-- Verification





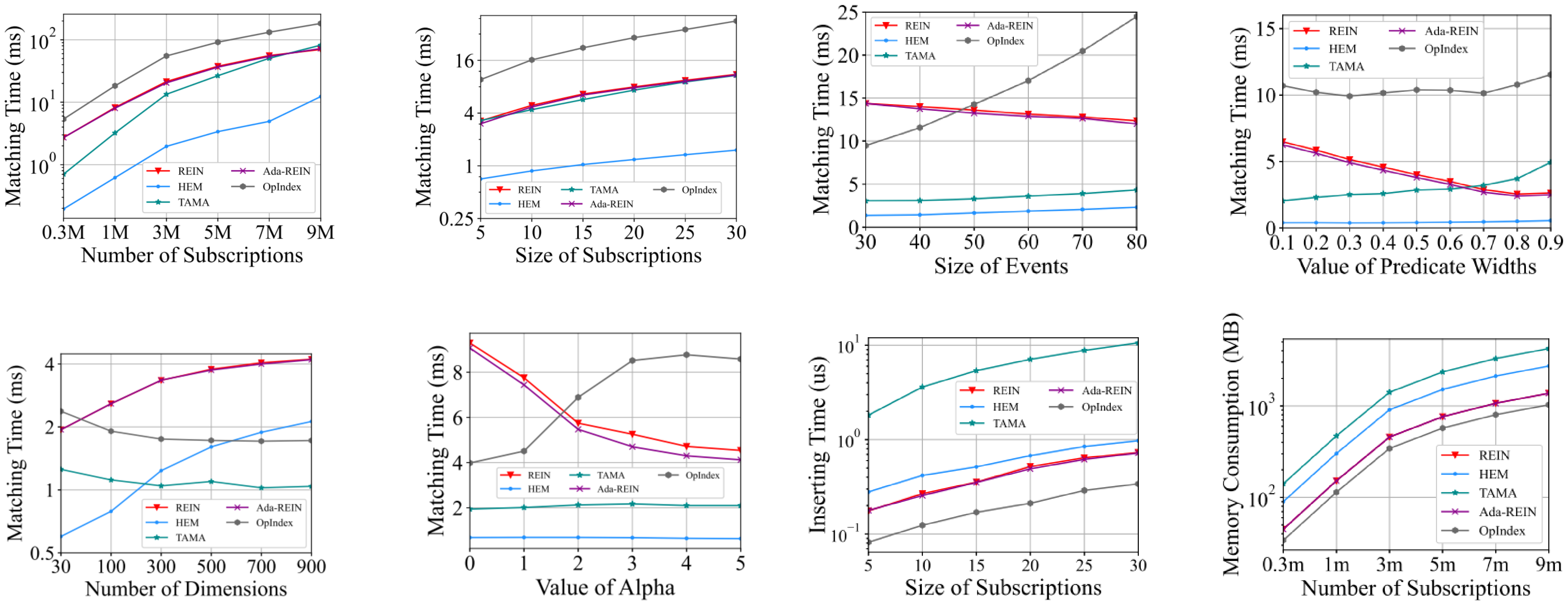
$$\text{Group Number g} = 32 \rightarrow 1 - \frac{0.31ms}{6.79ms} = 95.4\% \approx 96.875\%$$

5 Evaluation-- Verification



- Bottleneck is alleviated.
- Choose g = 32.

5 Evaluation--Metric



6 Future Work—series of optimizations

1. Double Reverse Optimization (Virtual Memory): how to select bitsets ?
2. Dynamic Group Optimization (Load Balancing): how to construct groups ?
3. Virtual Attribute Group Optimization
4. Real Attribute Group Optimization

high dimensional space

5. OR Operation Optimization (avx2, avx512): 512-bits OR op

- Other directions:

  approximate matching / response time / thread pool parallelization

Thanks for listening!

qshiyou@sjtu.edu.cn