Rank-at-a-Time Query Processing

Ahmed Elbagoury, **Matt Crane**, and Jimmy Lin
University of Waterloo

Processing Schemes

Query “*a b c*” — Example Posting Lists:

<docid, tf>

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:9>			

DAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

DAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

DAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

DAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

DaaT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

DaaT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

DaaT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

DAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TaaT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TaaT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TaaT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TaaT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TaaT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

TAAT

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

Quantization & Impact Ordering

- Pre-calculate term/document score contributions
- Quantize to integer range (impact score)
- Group terms by impact score
- Store descending impact, ascending docid within each group

Example Index

<docid, tf>

a	<1:6>	<2:2>	<5:6>	<6:10>	<9:2>	<11:9>
b	<2:3>	<6:12>	<11:15>			
c	<6:9>	<9:6>	<11:3>			

Quantized Impact Ordered Example Index

impact score: [docid, docid, …, docid]

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SaaT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SaaT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SaaT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

SAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

Rank-at-a-Time

- For a query "*a b c*" given example index, largest score possible is 34 ({a:10} ∩ {b:15} ∩ {c:9})
- Descend from score 36, performing intersections as necessary
- Score safe
- Anytime

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]
b	15: [11]	12: [6]	3: [2]	
c	9: [6]	6: [9]	3: [11]	

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RaaT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

RAAT

a	10: [6]	9: [11]	6: [1, 5]	2: [2, 9]		
b	15: [11]	12: [6]	3: [2]			
c	9: [6]	6: [9]	3: [11]			

Compositions

- A composition of *n* is a way of writing *n* as a sequence of strictly positive integers:
    - 10 + 11 + 15 is one composition of 36
- An *A*-restricted composition draws values from same range for all components
- Second-order restricted allows range of values to be different for each component
    - Drawn from the impact values for each term

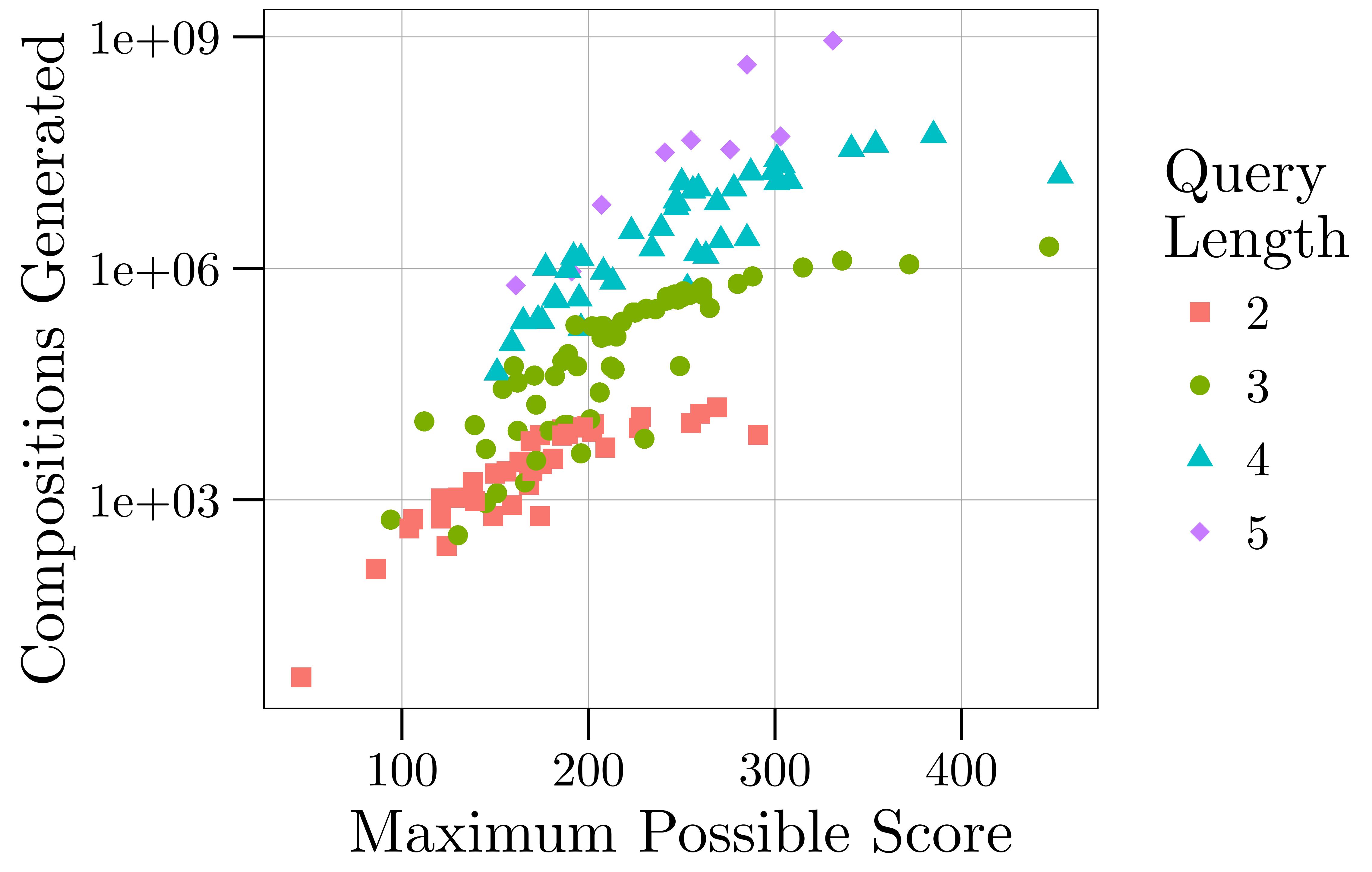
Compositions

- Combinatorial explosion
- (n-1)C(k -1) compositions of *n* into *k* parts
- Second-order restriction helps here, e.g. only one way to decompose max

Compositions vs Max Score

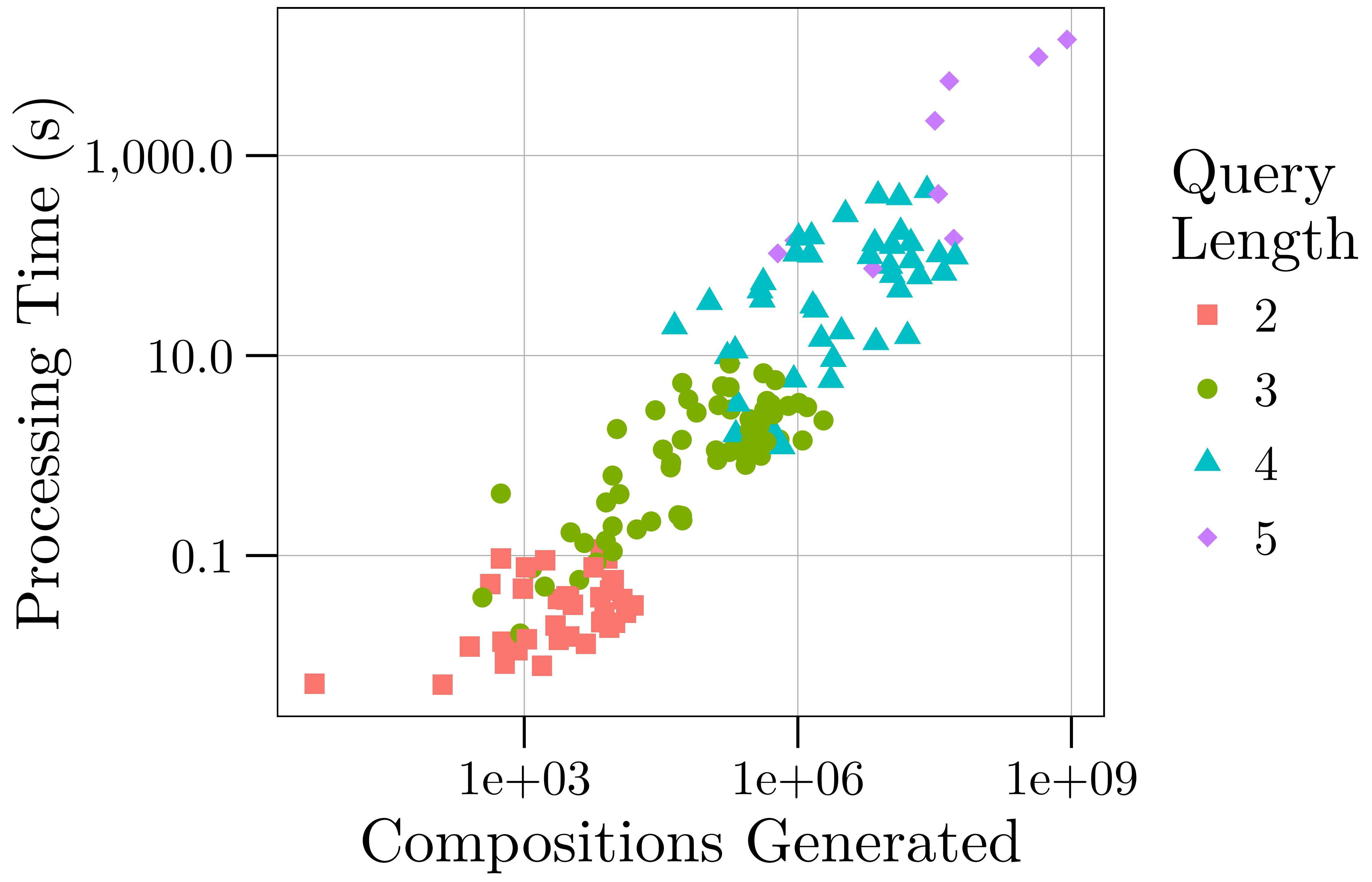
- Maximum score influences number of compositions (more scores to descend)

- Number of query terms influence number of compositions (more ways to generate a given score)



Time vs Compositions

- Each composition requires intersection testing
- Query time is therefore directly related



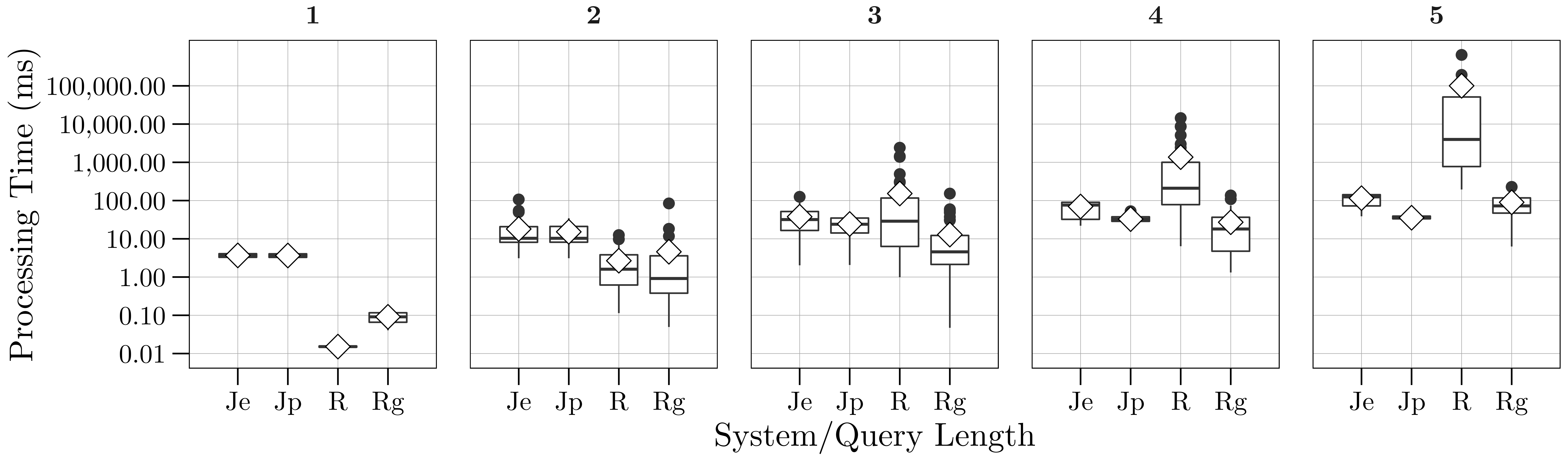
Grouped Rank-at-a-Time

- Number of intersections determine query time
- Reduce intersections by grouping impacts together
  - Requires the use of intermediary structure to store scores
- No longer score-safe

Grouped Rank-at-a-Time

- Three groups with 1/3 of impact scores each
- Intersect {high} ∩ {high} ∩ {high} first, then depth first back-off:
    - {high} ∩ {high} ∩ {medium}
    - {high} ∩ {high} ∩ {low}
    - …

Comparison


1
2
3
4
5
Processing Time (ms)
100,000.00
10,000.00
1,000.00
100.00
10.00
1.00
0.10
0.01
Je
Jp
R
Rg
System/Query Length

Summary

- Novel query processing strategy
- Naïve implementation impractically slow
- Score safe and anytime
    - To the best of our knowledge, only such processing strategy

Future Work

- Extension to disjunctive query processing
    - Weak compositions (allow a 0 value)
- Caching strategies for intersections
- Different grouping strategies
- Make use of anytime property to restrict time

Questions?
// Comments