Efficient Algorithms for Constrained Subspace Skyline in Structured Peer-to-Peer Systems

Khaled M. Banafaa and Ruixuan Li

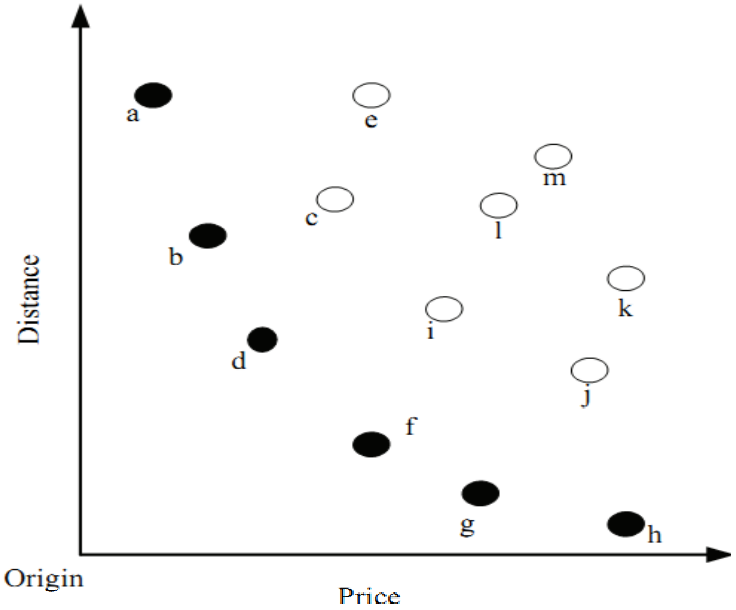
Huazhong University of Science and Technology

Outline

- Background
- Motivation and related work
- Constrained subspace skyline
  - Partitioning and distribution
  - Algorithms
- Experimental evaluation
- Conclusion

Skyline

- *A Skyline is the set of all non-dominated tuples*



Formal Definition

Skyline *S* of *T* tuples is

$\{s \in T: \neg\exists\, t \in T,\ \forall\, i \in [1..k]\ t_i \geq s_i \text{ and } \exists\, i \in [1..k]\ t_i > s_i\}$

Where t_i and s_i are the values of the i^{th} column of tuples *t* & *s* respectively

Constrained Subspace Skyline Definition

- Let
  - $S=\{d_1,d_2,\ldots,d_d\}$ be a d-dimensional space
  - PS be a set of points in S
  - $p=\{p_1,p_2,\ldots,p_d)$ a point with d dimensions
  - $S' \subseteq S$, S' is a subspace of S
  - $C=\{c_1,c_2,\ldots,c_k\}$ is a range constraints on S', expressed by $[c_{i,min}, c_{i,max}]$
- A point $p \in PS$ dominates $q \in PS$ ($p <_{S'} q$) on a subspace S' if

  1) on every $d_i \in S'$ $p_i \leq q_i$; and 2) on at least one dimension $d_j \in S'$, $p_j < q_j$
- The skyline of a subspace S' is a set $PS' \subseteq PS$ which are not dominated by any other point on the subspace S'

$$PS'=\{p \in PS \mid \neg \exists q \in PS: q <_{S'} p\}$$

- A constrained subspace skyline for $S' \subseteq S$ refers to a set of points $PS'_c = \{p \in PS_c \mid \neg \exists q \in PS_c: q <_{S'} p\}$, where $PS_c \subseteq PS$ and $PS_c=\{p \in PS \mid \forall d_i \in S': c_{i,min} \leq p_i \leq c_{i,max}\}$

Motivation

- Various users may issue queries with different subsets of attributes due to their interests and constraints
- An example:
  - A customer may be sensitive only on price and mileage regarding within some ranges while a car database may contain many attributes of cars, such as price, mileage, horsepower, age and fuel consumption

Related Work

Structured overlay does not support both constrained and subspace skyline

Unstructured overlay can support both constrained and subspace skyline but need contact all peers

Approach	Skyline	Subspace	Constrained	Overlay
DSL [18]	O		[illegible]	[illegible]
SSP/Skyframe [2, 3]	X		[illegible]	[illegible]
iSky [4 5]	X		[illegible]	[illegible]
SSW [6]	X		[illegible]	[illegible]
SFP [7]	X	O	O	Unstructured
DDS [8, 9]	X	O	X	Unstructured
SKYPEER/SKYPEER+ [10, 11]	O	X		Unstructured
BITPEER [12]	O	X		Unstructured
PaDSkyline [13, 14]	O	O	X	Unstructured
AGiDS [15]	X	O	O	Unstructured
FDS [16]	X	O	O	Unstructured
SkyPlan [17]	X	O	O	Unstructured

- Table: Different skyline query variants supported by distributed approaches (X: proposed for, O: also supports)

Our Objectives

- Using a simple network overlay such as Chord or Baton
- Accomplish full/subspace skyline queries efficiently
  1. Decrease visited peers
  2. Increase bandwidth saving
  3. Preserve progressiveness
  4. Exploit parallelism (while preserving progressiveness)

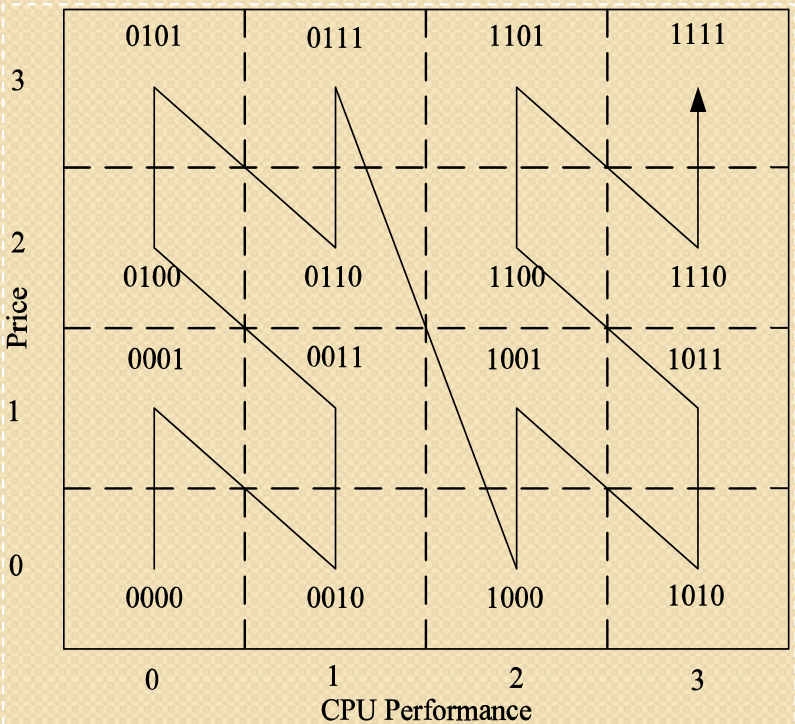
Baseline

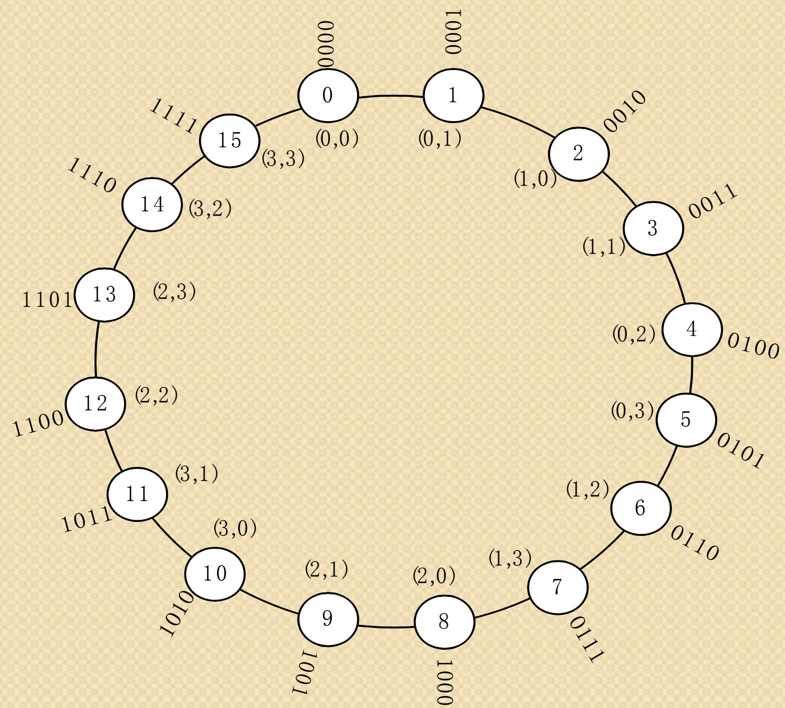
- We chose subsky [1] as a baseline for our work
  - It used Chord structure
  - It used a share-nothing structure
  - It did not support progressiveness
  - No parallelism
- It was modified to consider subspaces

Data Distribution

1-The data space is partitioned and Cells are given z-order addresses

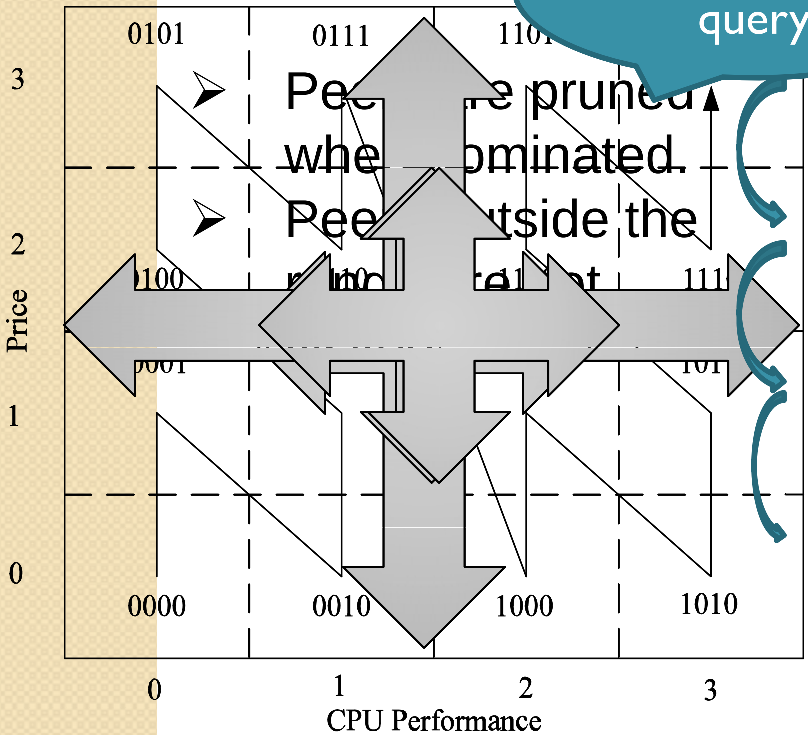
2- Partitions assigned to peers of a Chord or a Baton





Query Traversal and Example

- Peers' Traversal depends on query subspace and constraints.



2. Subspace query

Peers	CPU-performance			Price		
Z-address	Bits	CPU-performance (Z-address)	Traversal for subspace query	Bits	Price(Z-address)	Traversal for subspace query
0000	00	00 (0000)	00 (0000)	00	00 (0000)	00 (0000)
[illegible]	[illegible]	00 (0001)	00 (0001)	01	01 (0001)	00 (0010)
[illegible]	[illegible]	01(0010)	00 (0100)	00	00 (0010)	00 (1000)
[illegible]	01	01 (0011)	00 (0101)	01	01 (0011)	00 (1010)
0100	00	00 (0100)	01 (0010)	10	10 (0100)	01 (0001)
0101	00	00 (0101)	01 (0011)	11	11 (0101)	01 (0011)
0110	01	01 (0110)	01 (0110)	10	10 (0110)	01 (1001)
0111	01	01 (0111)	01 (0111)	11	11 (0111)	01 (1011)
1000	10	10 (1000)	10 (1000)	00	00 (1000)	10 (0100)
1001	10	10 (1001)	10 (1001)	01	01 (1001)	10 (0110)
1010	11	11 (1010)	10 (1100)	00	00 (1010)	10 (1100)
1011	11	11 (1011)	10 (1101)	01	01 (1011)	10 (1110)
1100	10	10 (1100)	11 (1010)	10	10 (1100)	11 (0101)
1101	10	10 (1101)	11 (1011)	11	11 (1101)	11 (0111)
1110	11	11 (1110)	11 (1110)	10	10 (1110)	11 (1101)
1111	11	11 (1111)	11 (1111)	11	11 (1111)	11 (1111)

Global Fullspace Skyline

Algorithm 1 Constrained Full Space Algorithm

Input:
RS: Received Skyline
Output:
DS: Discovered Skyline
BEGIN
LS: Constrained Local Skyline Points
$DS = \phi$
for all $P \in LS$ **do**
  if $\nexists Q \in RS$: Q $\prec P$ **then**
    $DS = DS \cup P$;
  end if
end for
Send DS to querying peer as final skyline points
$RS = RS \cup DS$
Send RS to next unpruned peer
END

Subspace Skyline

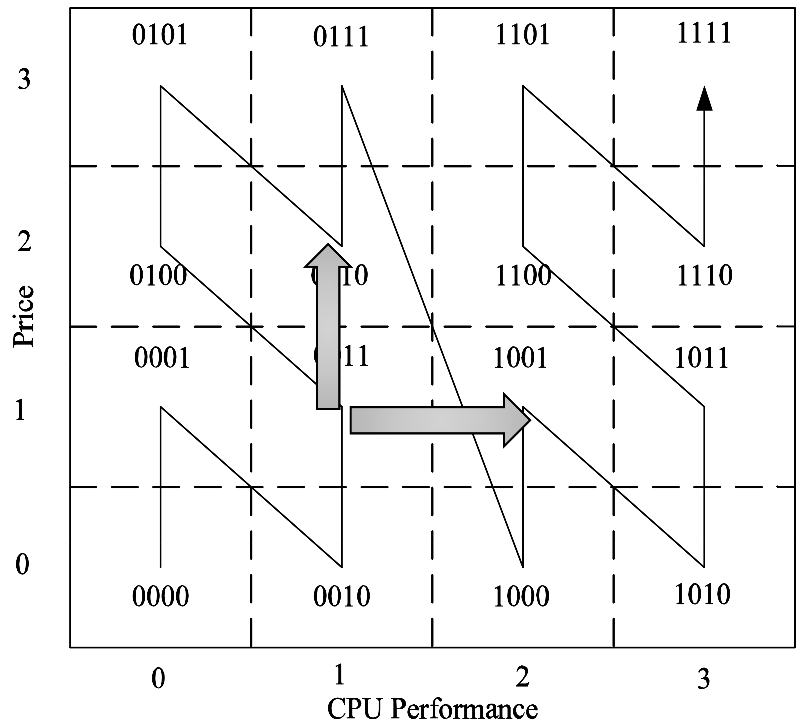
Algorithm 2 Constrained Subspace Skyline Algorithm

```
Input:
RFS: Rcvd Final Sky Pts
RGS: Rcvd Group Sky Pts
Output:
DS: Discovered Skyline
BEGIN
LS: Constrained Local Sky Pts
LS = findSkyline(LS ∪ RGS)
DS = φ
for all P ∈ LS do
   if ∄Q ∈ RFS : Q ≺ P then
      DS = DS ∪ P;
   end if
end for
if (Last-peer-of-subspace-group) then
   Send DS to query peer as a final sky
   RFS = RFS ∪ DS
   Send RFS to first unpruned peer in SG
else
   RGS = DS
   Send RFS and RGS to next peer in SG
end if
END
```

Subspace group(SG) is treated as a peer of the full space

Parallelism

- As parallelism is used to minimize total processing time:
  - Each node/group can be sent to d peers (for fullspace)/|subspace| peers (for subspace) for queries
  - Due to un-comparability, progressiveness are preserved



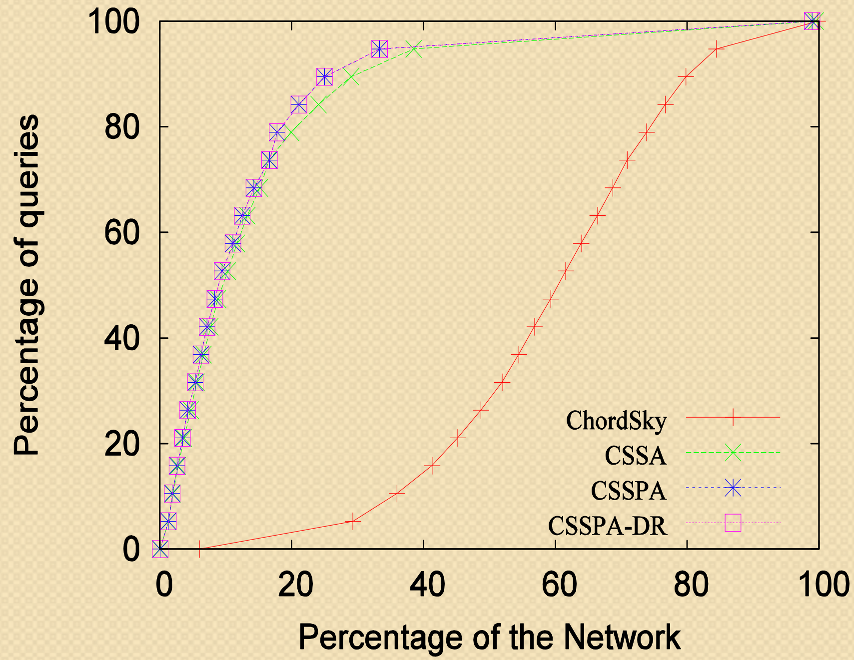
For More Data Transfer Reduction

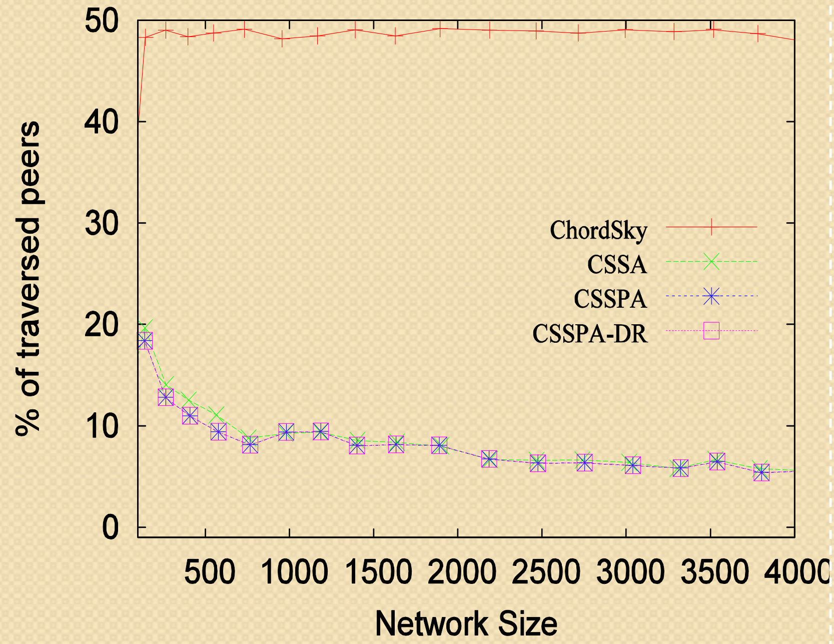
- In parallelism, notice:
  - Traversal is from lower to upper value in each dimension.
- Thus,
  - No need to send all so-far skyline points.
  - It is enough to only send the subspace skyline of all dimensions in consideration - the dimension we are sending through.

Results (I)

Queries vs. peers traversed

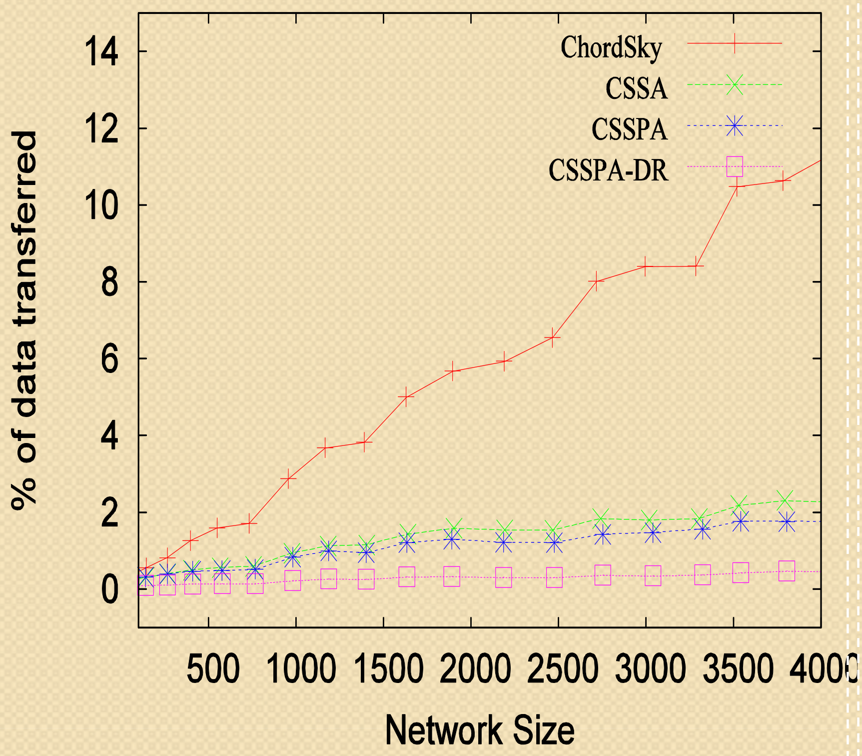
Traversed peers vs. network size



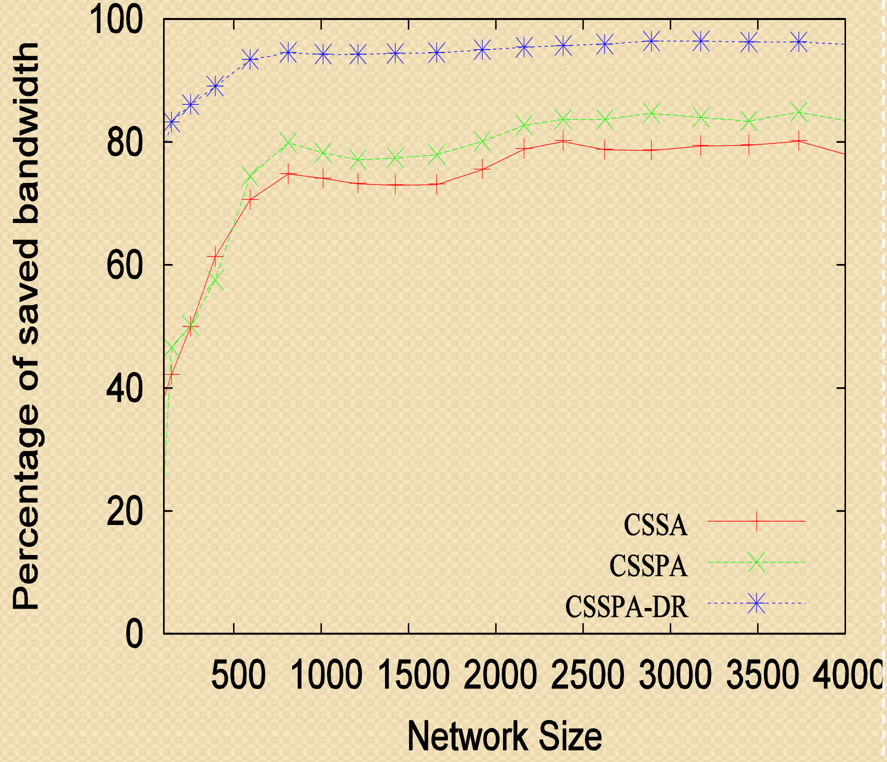


Results (II)

Data transferred vs network size

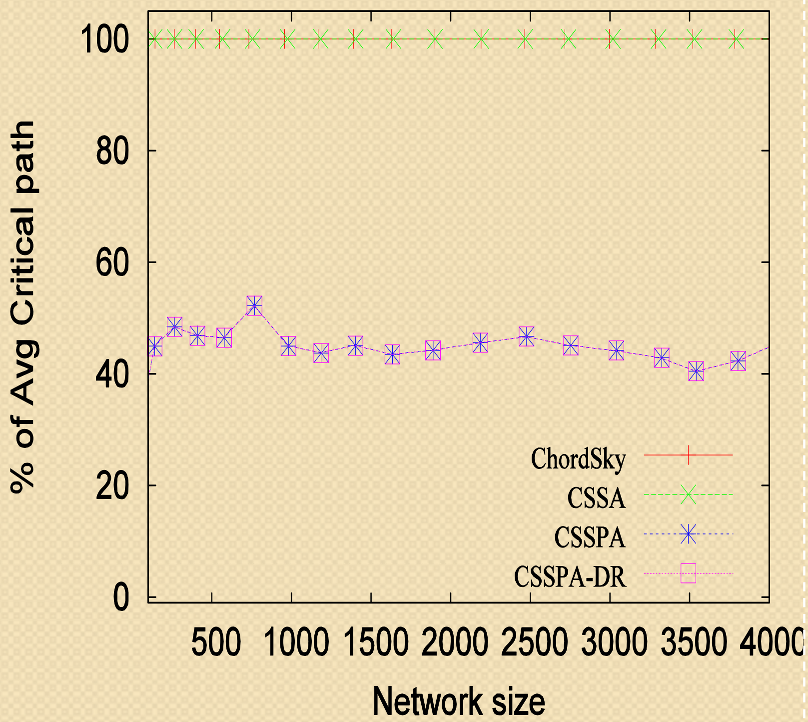


Saved Bandwidth

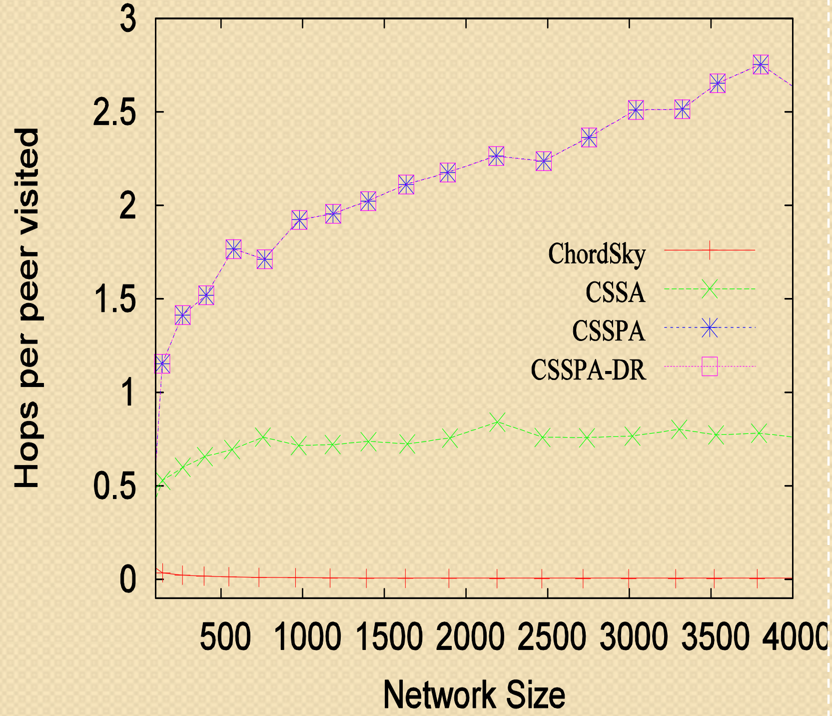


Results (III)

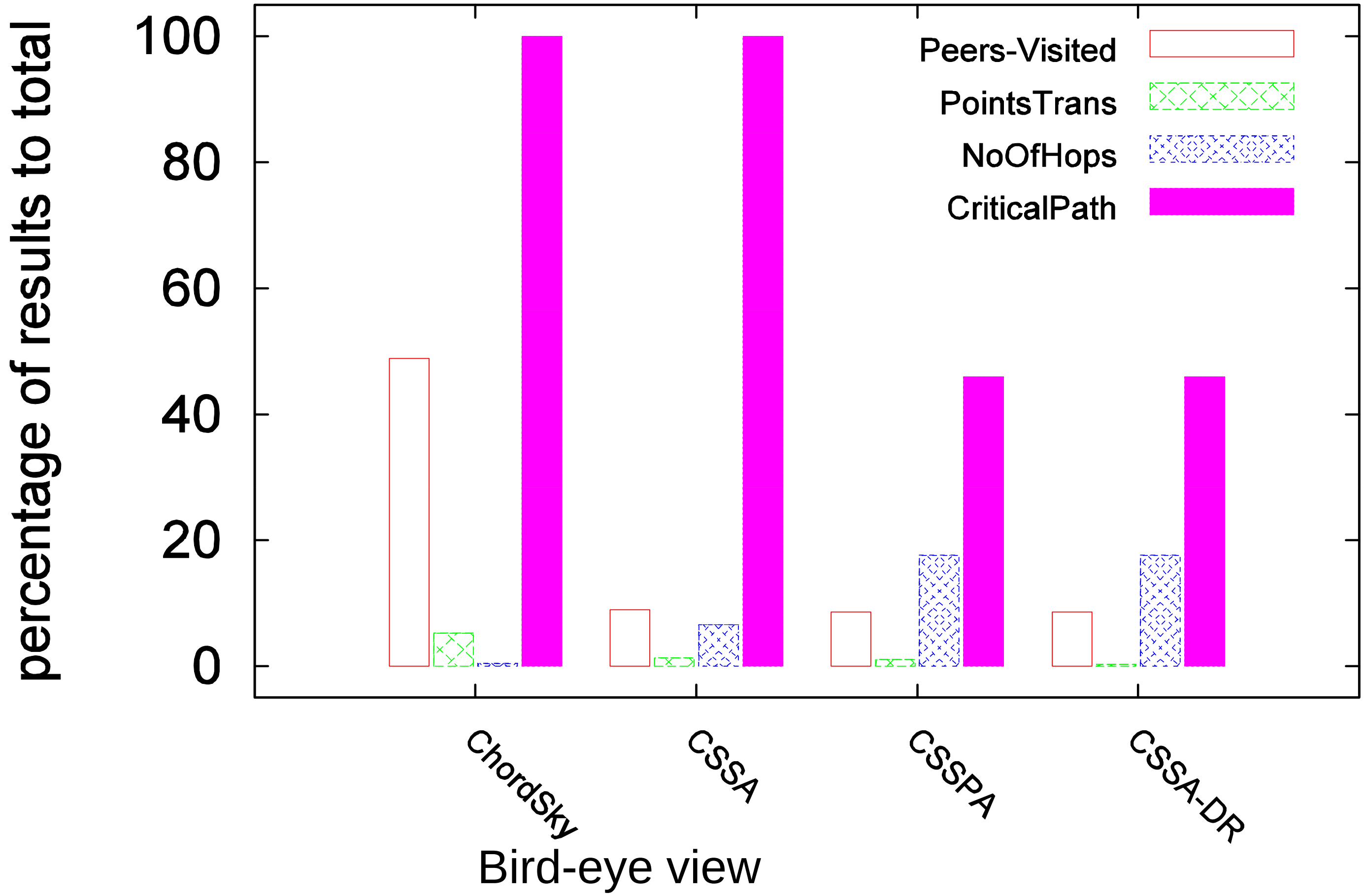
Parallel effects on critical path



Cost



Results (IV)



percentage of results to total
100
80
60
40
20
0
Peers-Visited
PointsTrans
NoOfHops
CriticalPath
ChordSky
CSSA
CSSPA
CSSA-DR
Bird-eye view

Thank you

Reference (I)

[1] Lin Zhu, Shuigeng Zhou, and Jihong Guan," Efficient skyline retrieval on peer-to-peer networks, " in FGCN. 2007, pp. 309–314, IEEE

[2] Wang, S., Ooi, B., Tung, A., Xu, L.: E□cient skyline query processing on peer-to-peer networks. In: Proceedings of Inter-national Conference on Data Engineering (ICDE), pp. 1126-1135 (2007)

[3] Wang, S., Vu, Q.H., Ooi, B.C., Tung, A.K., Xu, L.: Skyframe:a framework for skyline query processing in peer-to-peer sys-tems. The VLDB Journal 18(1), 345-362 (2009)

[4] Chen, L., Cui, B., Lu, H., Xu, L., Xu, Q.: iSky: Efficient and progressive skyline computing in a structured P2P net-work. In: Proceedings of the International Conference on Distributed Computing Systems (ICDCS), pp. 160-167 (2008)

[5] Cui, B., Chen, L., Xu, L., Lu, H., Song, G., Xu, Q.: Efficient skyline computation in structured peer-to-peer systems. IEEE Transactions on Knowledge and Data Engineering (TKDE) 21(7), 1059{1072 (2009)

[6] Li, H., Tan, Q., Lee, W.: Efficient progressive processing of skyline queries in peer-to-peer systems. In: Proceedings of theInternational Conference on Scalable Information Systems (Infoscale), p. 26 (2006)

[7] Huang, Z., Jensen, C.S., Lu, H., Ooi, B.C.: Skyline queries against mobile lightweight devices in manets. In: Proceedings of International Conference on Data Engineering (ICDE), p. 66 (2006)

[8] Hose, K., Lemke, C., Sattler, K.: Processing Relaxed Skylines in PDMS Using Distributed Data Summaries. In: Proceedings of International Conference on Information and Knowledge Management (CIKM), pp. 425-434 (2006)

Reference (II)

[9] Hose, K., Lemke, C., Sattler, K., Zinn, D.: A relaxed but not necessarily constrained way from the top to the sky. In: Proceedings of International Conference on Cooperative In-formation Systems (CoopIS), pp. 339{407 (2007)

[10] Vlachou, A., Doulkeridis, C., Kotidis, Y., Vazirgiannis, M.: SKYPEER: Efficient subspace skyline computation over distributed data. In: Proceedings of International Conference on Data Engineering (ICDE), pp. 416-425 (2007)

[11] Vlachou, A., Doulkeridis, C., Kotidis, Y., Vazirgiannis, M.: Efficient routing of subspace skyline queries over highly distributed data. IEEE Transactions on Knowledge and Data Engineering (TKDE) 22(12), 1694-1708 (2010)

[12] Fotiadou, K., Pitoura, E.: BITPEER: continuous subspace skyline computation with distributed bitmap indexes. In: Proceedings of International Workshop on Data Management in Peer-to-Peer Systems (DaMaP), pp. 35{42 (2008)

[13] Chen, L., Cui, B., Lu, H.: Constrained skyline query processing against distributed data sites. IEEE Transactions on Knowledge and Data Engineering (TKDE) 23(2), 204-217 (2011)

[14] Cui, B., Lu, H., Xu, Q., Chen, L., Dai, Y., Zhou, Y.: Parallel distributed processing of constrained skyline queries by filtering. In: Proceedings of International Conference on Data Engineering (ICDE), pp. 546-555 (2008) [15] Rocha-Junior, J.B., Vlachou, A., Doulkeridis, C., Norvag, K.: AGiDS: A grid-based strategy for distributed skyline query processing. In: Proceedings of International Conference on Data Management in Grid and Peer-to-Peer Systems (Globe), pp. 12-23 (2009)

Reference (III)

[16] Zhu, L., Tao, Y., Zhou, S.: Distributed skyline retrieval with low bandwidth consumption. IEEE Transactions on Knowledge and Data Engineering (TKDE) 21(3), 384-400 (2009)

[17] Rocha-Junior, J.B., Vlachou, A., Doulkeridis, C., Norvag, K.: Efficient execution plans for distributed skyline query pro-cessing. In: Proceedings of International Conference on Extending Database Technology (EDBT), pp. 271-282 (2011)

[18] Wu, P., Zhang, C., Feng, Y., Zhao, B., Agrawal, D., Ab-badi, A.: Parallelizing skyline queries for scalable distribu-tion. In: Proceedings of International Conference on Extend-ing Database Technology (EDBT), pp. 112-130 (2006)

[19] Lijiang Chen, Bin Cui , Linhao Xu, and Heng Tao Shen , "Distributed cache indexing for efficient subspace skyline computation in P2P networks, " in DASFAA (1), Hiroyuki Kitagawa, Yoshiharu Ishikawa, Qing Li , and Chiemi Watanabe , Eds. 2010, vol. 5981 of Lecture Notes in Computer Science, pp . 3–18, Springer.