



Relax and Let the Database do the Partitioning Online

Alekh Jindal, Jens Dittrich

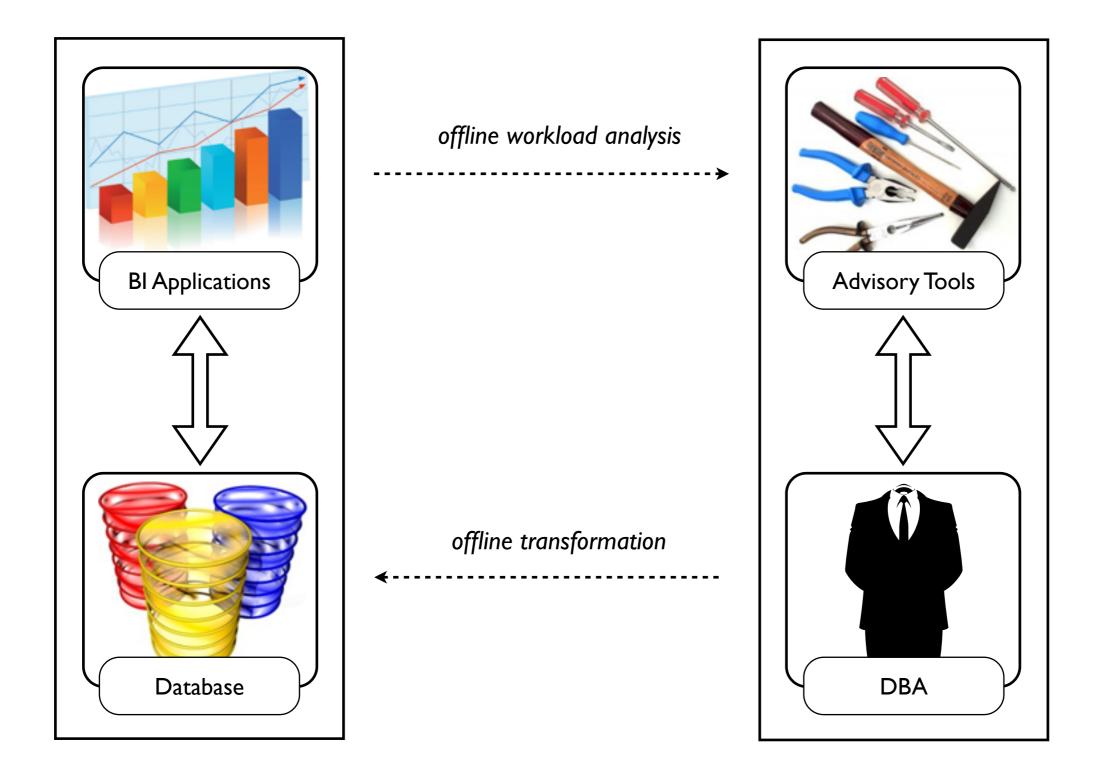
- presented by Stefan Schuh

VLDB International Workshop on Real-Time Business Intelligence

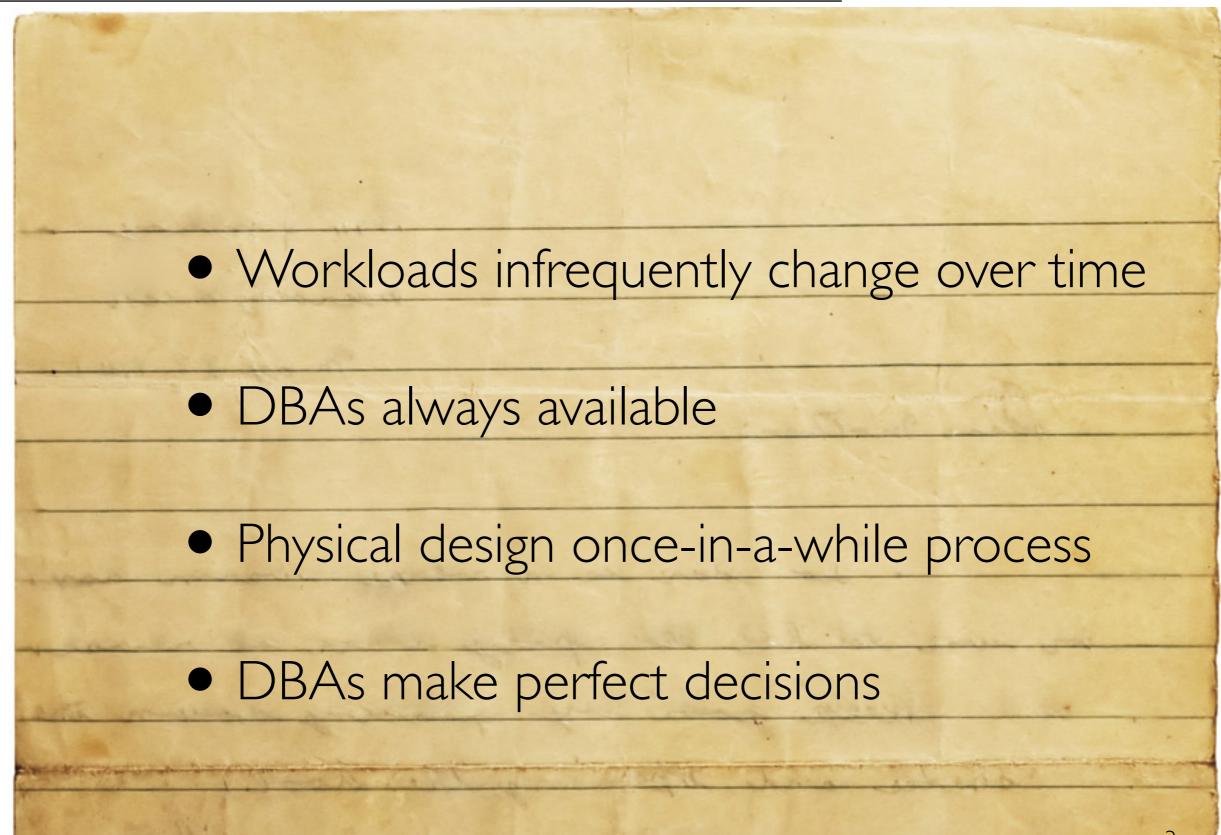
September 2, 2011

Thursday, September 1, 2011

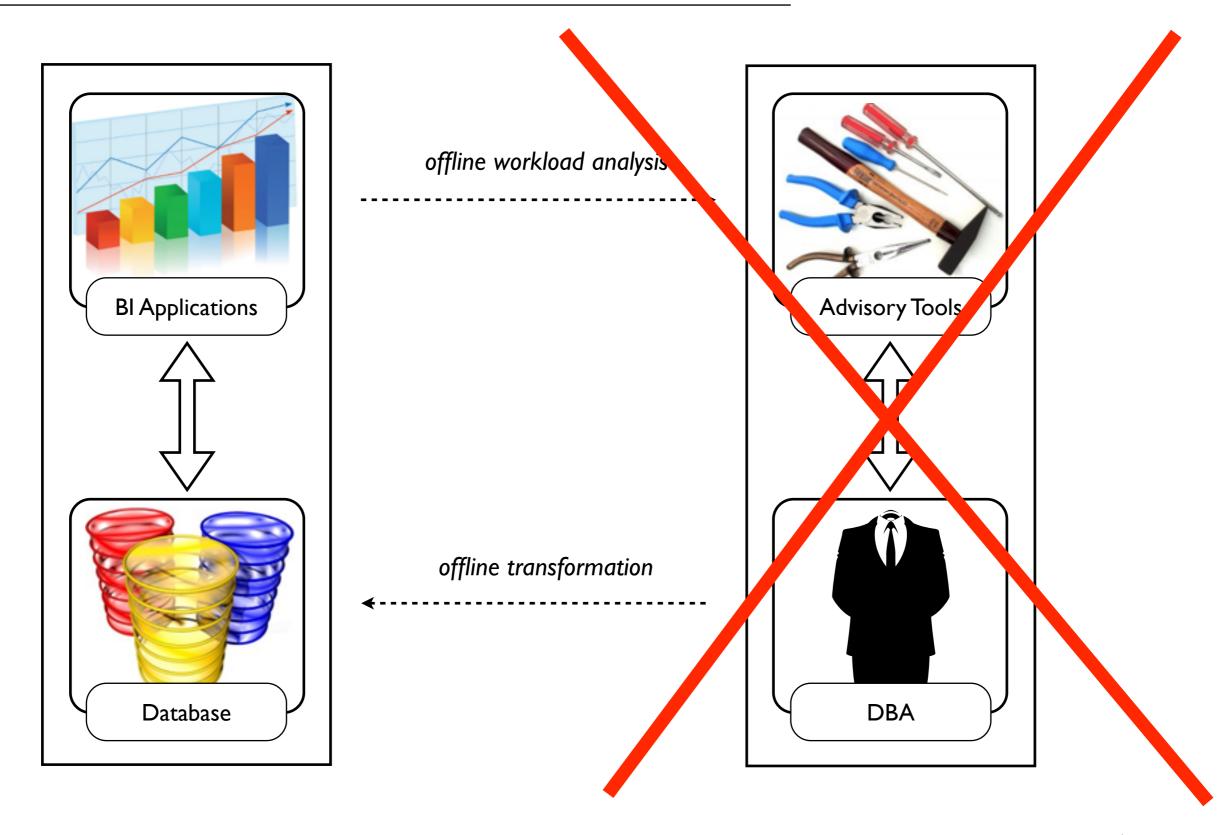
Motivation: Offline Physical Database Design



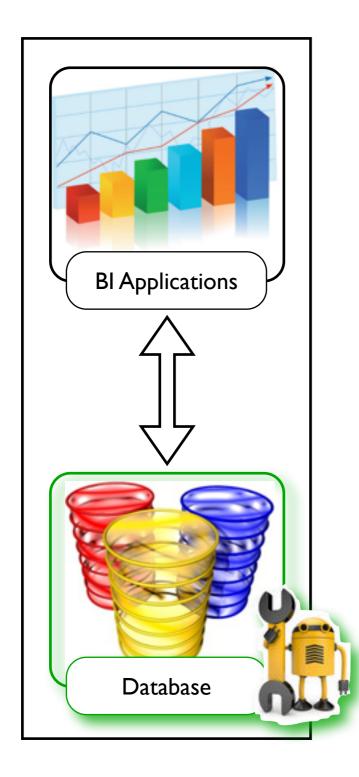
Offline Design Cheats!



Motivation: Offline Physical Database Design



Motivation: Online Physical Database Design



Sub-Problem	Proposed Solution
Indexing	Online Indexing Database Cracking Adaptive Indexing
Materialized Views	Dynamic Materialized Views
Partitioning	WE!

Challenges in Online Partitioning

- Collecting online workload
- Analyzing workload online
- Querying with online workload analysis
- Creating partitions online

Challenges in Online Partitioning

- Collecting online workload
- Analyzing workload online
- Querying with online workload analysis
- Creating partitions online

- offline approach: take the last query log as workload (static)
- online approach: collect incoming queries in a window and slide it when more queries come (dynamic)

- offline approach: take the last query log as workload (static)
- online approach: collect incoming queries in a window and slide it when more queries come (dynamic)

Window Size = 5

$$Q_1 Q_2 Q_3 Q_4 Q_5$$

- offline approach: take the last query log as workload (static)
- online approach: collect incoming queries in a window and slide it when more queries come (dynamic)

Window Size = 5

$$Q_1 Q_2 Q_3 Q_4 Q_5 Q_6$$

- offline approach: take the last query log as workload (static)
- online approach: collect incoming queries in a window and slide it when more queries come (dynamic)

Window Size = 5

$$Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \quad Q_7$$

- offline approach: take the last query log as workload (static)
- online approach: collect incoming queries in a window and slide it when more queries come (dynamic)

$$Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \quad Q_7 \quad \dots$$

How to Express the Partitioning Problem?

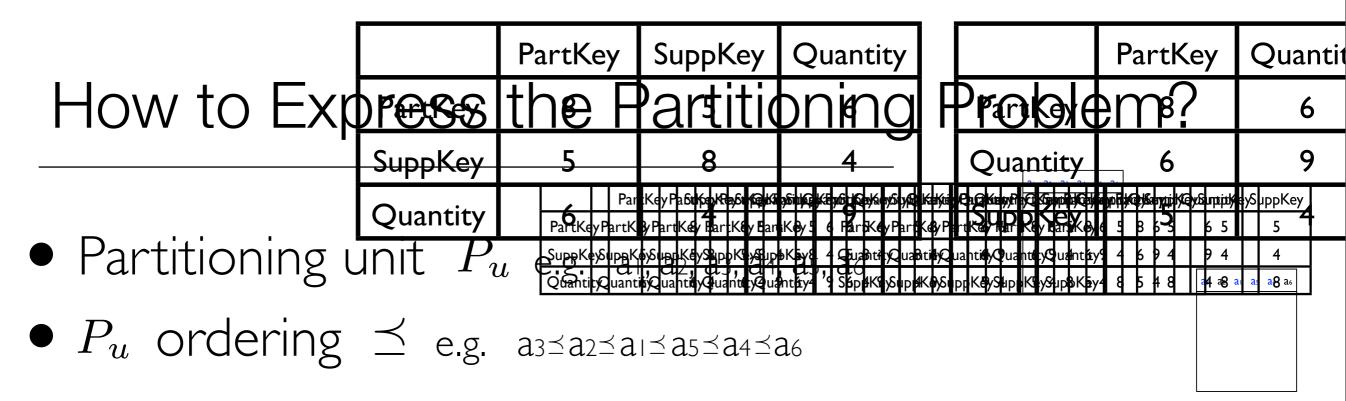
• Partitioning unit P_u e.g. a1, a2, a3, a4, a5, a6

2	l ı	a 2	a 3	a 4	a 5	a 6

		PartKey	SuppKey	Quantity			PartKey	Quantit
How to Exp	୲୲୶ୠୄୄ	th€ F	Partitic	bnina	F	Prole	em?	6
• • • • • • • • • • • • • • • • • • •	SuppKey	5	8	4		Quantity	6	9
	Quantity	Par PartKeyPartK	:Key Paɓ u ƙa Ke afa Y PartiK éy B artKéy Ban	϶ϐͷͷϸϾϼϳϒ϶ϐ϶ϧϐϳϲϥϬͼϙϧϐͼϗϧϐ ϗϹϐϒͺϨͺͺͺͺϐͺͺΡϐϫϲ·ϳϹϹͼϗϒΡ϶ϲ·	KanKµa KaβγP		anna ar an	еубиррКеу 5
 Partitioning L 	unit P_{u}		δ δ ν Quanti y Quanti v V V V V V V V V V V V V V	bKays 4 Quanti Quab http://www.supticarticarter.com	ti liy Qu K a γSu	յanti s,⊄ uantityՉuantit ppKeySupnKsySupbKa	/⁴ 8 5 4 8 4 8	4 8

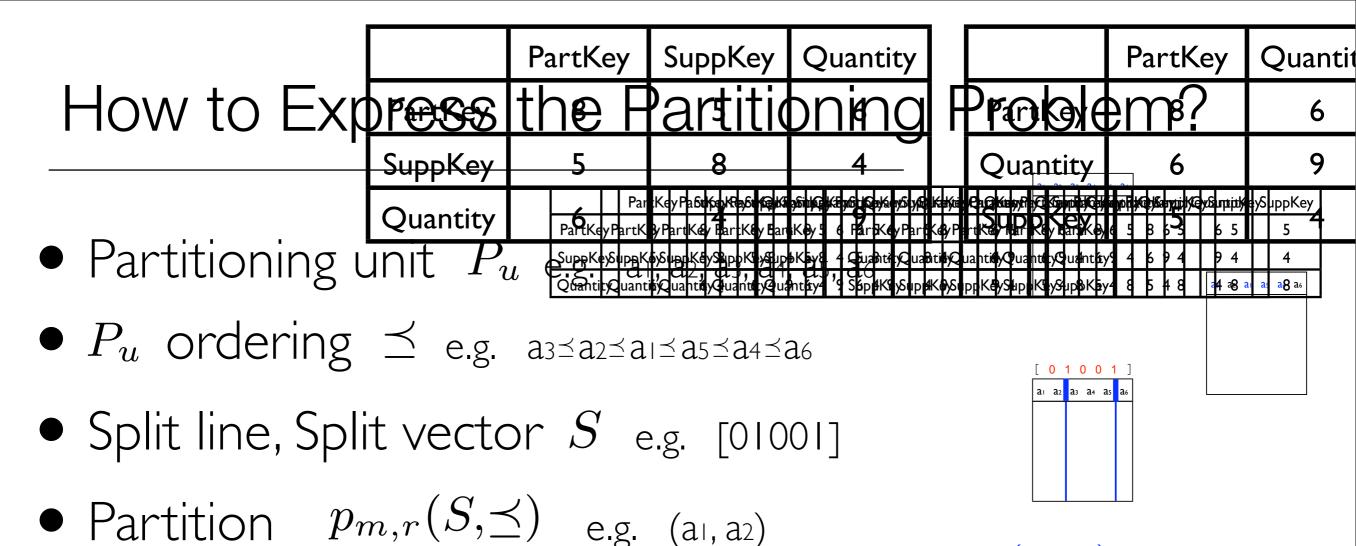
• P_u ordering \preceq e.g. $a_3 \preceq a_2 \preceq a_1 \preceq a_5 \preceq a_4 \preceq a_6$

a 3	a 2	aı	a 5	a 4	a 6

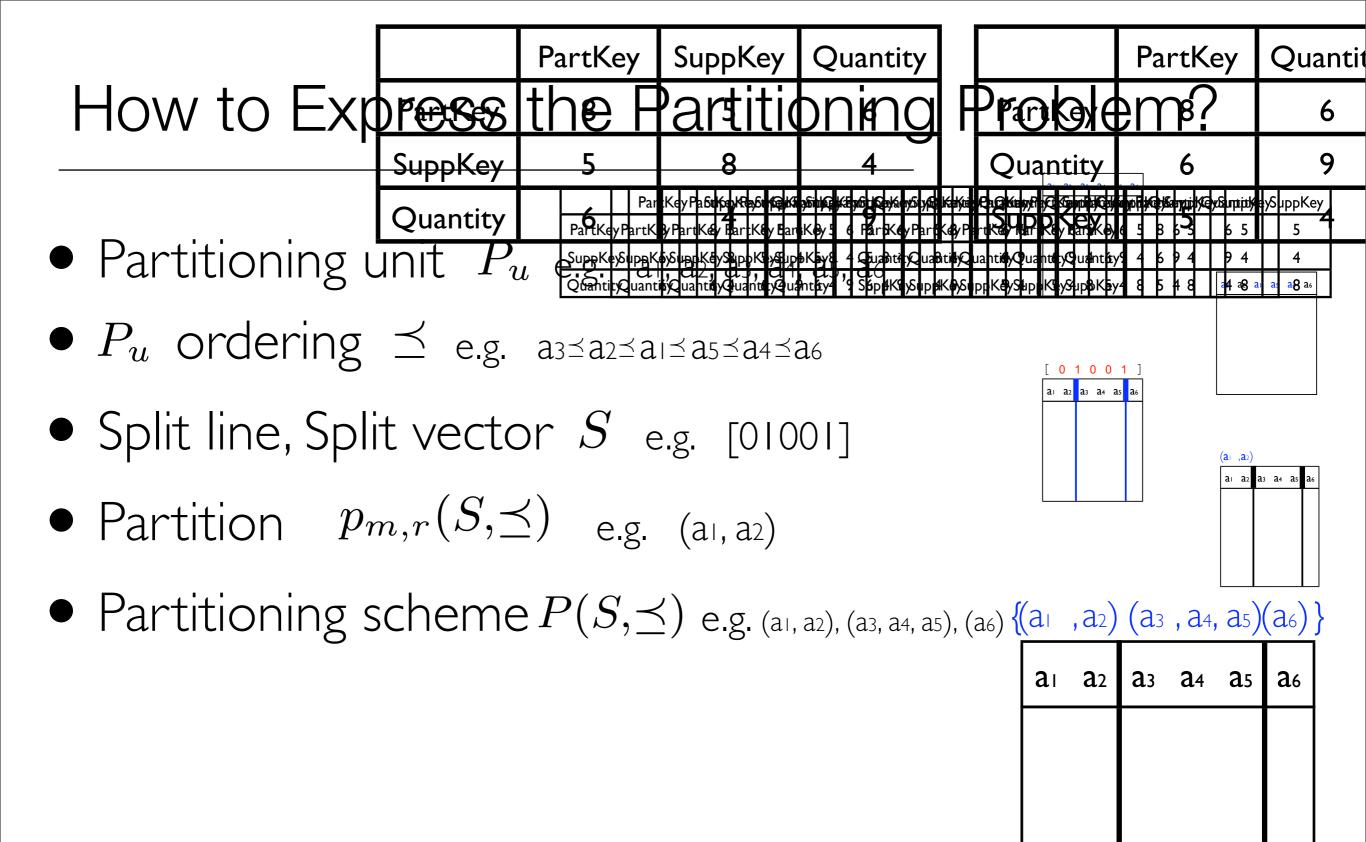


• Split line, Split vector S e.g. [01001]

[() 1	() () ′	1
a	a 2	a3	a 4	a 5	a6



(a)	,a 2)				
aı	a 2	a3	a 4	a 5	a 6



What about Horizontal Partitioning?

- Just rotate the table by 90 degrees
- P_u abstraction allows us to solve both problems
- P_u can be attributes, row-ranges, or any other table slice

aı	a 2	a3	a 4	a 5	a 6

	r4	r 2	rı

		PartKey	SuppKey	Quantity			PartKey	Quantit
Partitioning	Partio C	lem:	What	to A	n	A VEC	? 8	6
	SuppKey	5	8	4		Quantity	6	9
	Quantity	Par PartKeyPartK	:KeyPa6 thorke 96 t foeks YPartK éy B artK éy B ar	жылы каубу Баула уж Кеу! (Пат Кеу Раг	KatKui Ka8∕P		engelinikaengenikkaengenikkaengenikkaengenikkaengenikkaengenikkaengenikkaengenikkaengenikkaengenikkaengenikkae eksterise	aySuppKey 5
 Partitioning L 	unit P_{i}		eyδuppKeySuppKeySup , Ct2, Ct2, Ct1 ioQuanteyGuanteyGua		- / 1	uanti dy Q uantity Quaint íty Dir Kay Suda Kay Suda Kay		4

- P_u ordering \preceq e.g. $a_3 \preceq a_2 \preceq a_1 \preceq a_5 \preceq a_4 \preceq a_6$
- Split line, Split vector $S_{\rm e.g.}$ [01001]
- Partition $p_{m,r}(S, \preceq)$ e.g. (a1, a2) PartKey SuppKey Quantity PartKey Quantity SuppKey Par net e.g (a, $\frac{22}{PartKey}$, $\frac{24}{4}$, $\frac{15}{6}$), (a) (\mathbf{P}, \mathbf{T}) 5 6 $W_{t_k}^{\mathbf{5}}$ 8 9 Quantity 4 4 6 9 SuppKey 5 8 Quantity 4 4
- Problem statement Find \preceq , S' such that: $S' = \underset{S}{\operatorname{argmin}} C_{\operatorname{est.}} \left(W_{t_k}, P(S, \preceq) \right)$

How to Analyze the Workload?

Step 1: Finding Partitioning Unit Ordering

- offline approach: create affinity matrix and cluster it once, as proposed by Navathe et. al.
- online approach: leverage the affinity idea, but dynamically update and cluster the affinity matrix

Offline Partitioning Unit Ordering

• Create affinity matrix having attributes co-occurrences

				PartKey	Su	ррКеу	Quant	ity			
	PartKey	Sup	PartKey Key	Quantity		5	6	_ P	artKey	Quantity	SuppKey
PartKey	8		SuppKey	6		8 Part	Key 4		8	6	5
SuppKey	5	Į	Quantry B	4		₄ Qua	9 ntity		6	9	4
Quantity	6		4	9		Supp	oKey		5	4	8

Custer affini PartKey SuppKey Quantity
PartKey Quantity
PartKey Quantity
PartKey Quantity
PartKey Quantity
Quantity PartKey Quantity SuppKey artKey SuppKe Quantity PartKey Quantity SuppKey -H2 artK **K**rtKe 6 8 4 $M(\preceq) = \sum_{i=1}^{\text{SuppKey}} \sum_{j=1}^{5} \frac{8}{A} \frac{4}{9} \frac{\text{Quantity}}{\text{SuppKey}} \frac{6}{5} \frac{9}{4} \frac{\text{SuppKey}}{\text{QBantity}} \frac{5}{6} \frac{8}{4} \frac{4}{9} \frac{\text{Quantity}}{\text{SuppKey}} \frac{6}{5} \frac{8}{4} \frac{4}{9} \frac{1}{9} \frac{\text{Quantity}}{\text{SuppKey}} \frac{6}{5} \frac{8}{4} \frac{4}{9} \frac{1}{9} \frac$ Quantity 9 SuppKey 5 4 8

$M(\preceq) = 404$								
	PartKey SuppKey Quantity							
PartKey	8	5	6					
SuppKey	5	8	4					
Quantity	6	4	9					

 $M(\preceq) = 440$

(=)								
	PartKey	Quantity	SuppKey					
PartKey	8	6	5					
Quantity	6	9	4					
SuppKey	5	4	8					

Online Partitioning Unit Ordering

• Update only the referenced P_u in affinity matrix

	PartKey	Quantity	SuppKey
PartKey	8	6	5
Quantity	6	9	4
SuppKey	5	4	8

(PartKey, SuppKey)
5

()		PartKey	Quantity	SuppKey
<i>Y</i>)	PartKey	9	6	6
	Quantity	6	9	4
	SuppKey	6	4	9

• Re-cluster only the referenced P_u in affinity matrix

	+48 —	0	
	PartKey	Quantity	SuppKey
PartKey	9	6	6
Quantity	6	9	4
SuppKey	6	4	9

	SuppKey	PartKey	Quantity
SuppKey	6	9	6
PartKey	4	6	9
Quantity	9	6	4

How to Analyze the Workload?

Step 2: Enumerating Split Vectors

 offline approach: consider all possible split vectors (brute force)

aı	a 2	a3	••	•••	an

Complexity:
$$2^{n-1}$$

How to Analyze the Workload?

Step 2: Enumerating Split Vectors

- offline approach: consider all possible split vectors (brute force)
- online approach: One-dimensional Online Partitioning (O₂P) Algorithm

<u>Technique I</u>: prune non-referenced partitioning units <u>Technique 2</u>: consider split vectors greedily <u>Technique 3</u>: save previous best split vectors using dynamic programming

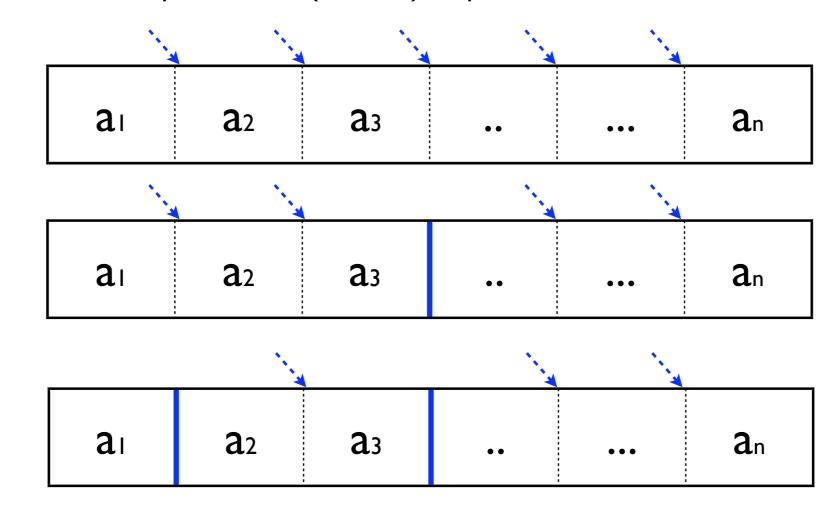
<u>Idea</u>: Prune the unused (non-referenced) P_u in at most two separate partitions

a 1 a 2	a3	••	•••	an
-----------------------	----	----	-----	----

Complexity: For p leading and q trailing unused P_u $2^{n-p-q-1}$

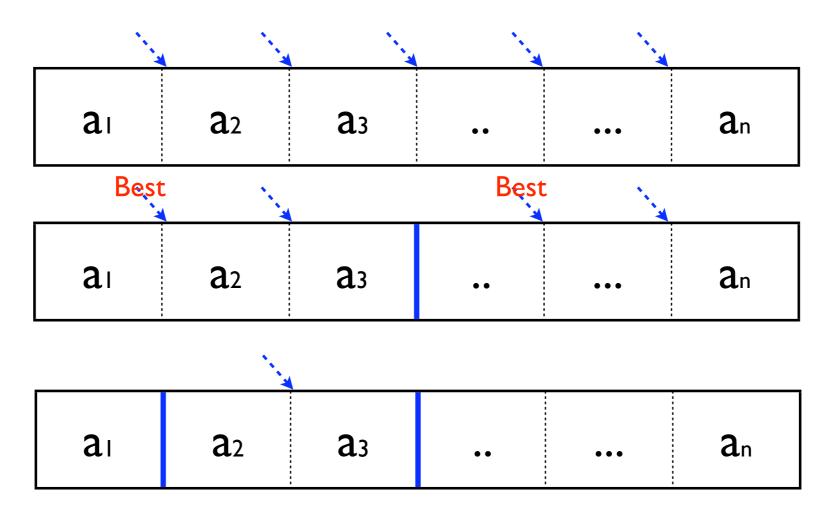
Greedy Split Vector Enumeration

Idea: Mark only one (best) split vector at a time



<u>Complexity</u>: worst case n²

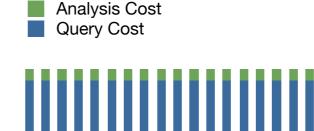
Idea: save best split vectors in un-split partitions



How to Amortize Partitioning Analysis?

offline approach: querying after computing and creating partitions
 Analysis Cost Query Cost

online approach:
 option1: interleave queries with partitioning analysis
 option2: queries in a separate thread



Goals of the Experiments

- Does greedy partitioning hurt Quality?
- How much is O₂P faster?
- Can such a system adapt to changing workload ?
- Will our approach work on real systems?

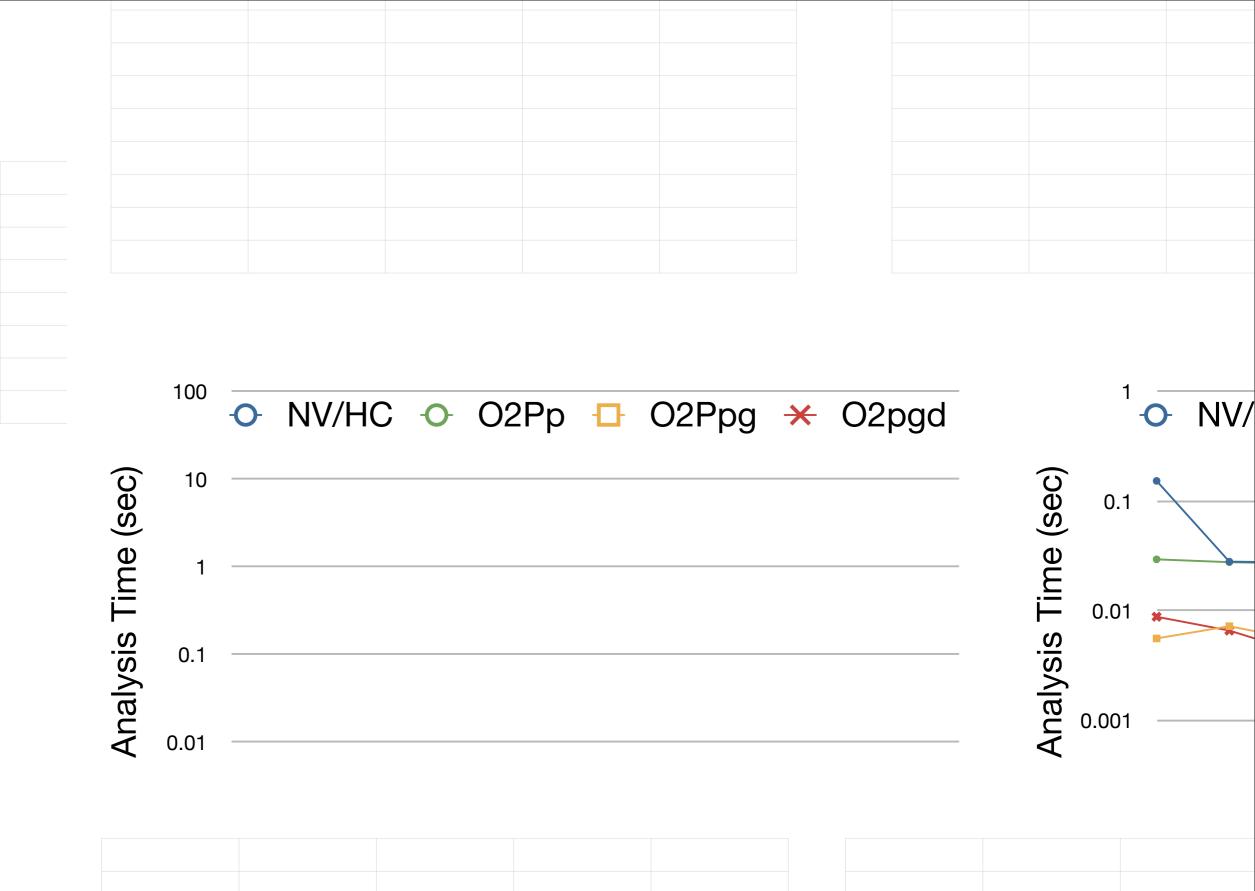
Dynamic Workload

- Mix of OLTP and OLAP style queries
- OLTP: 1% selectivity and 75-100% attributes
- OLAP: 10% selectivity and 1-25% attributes
- Vary the fraction of OLTP-OLAP over time

Does Greedy Partitioning Hurt Quality?

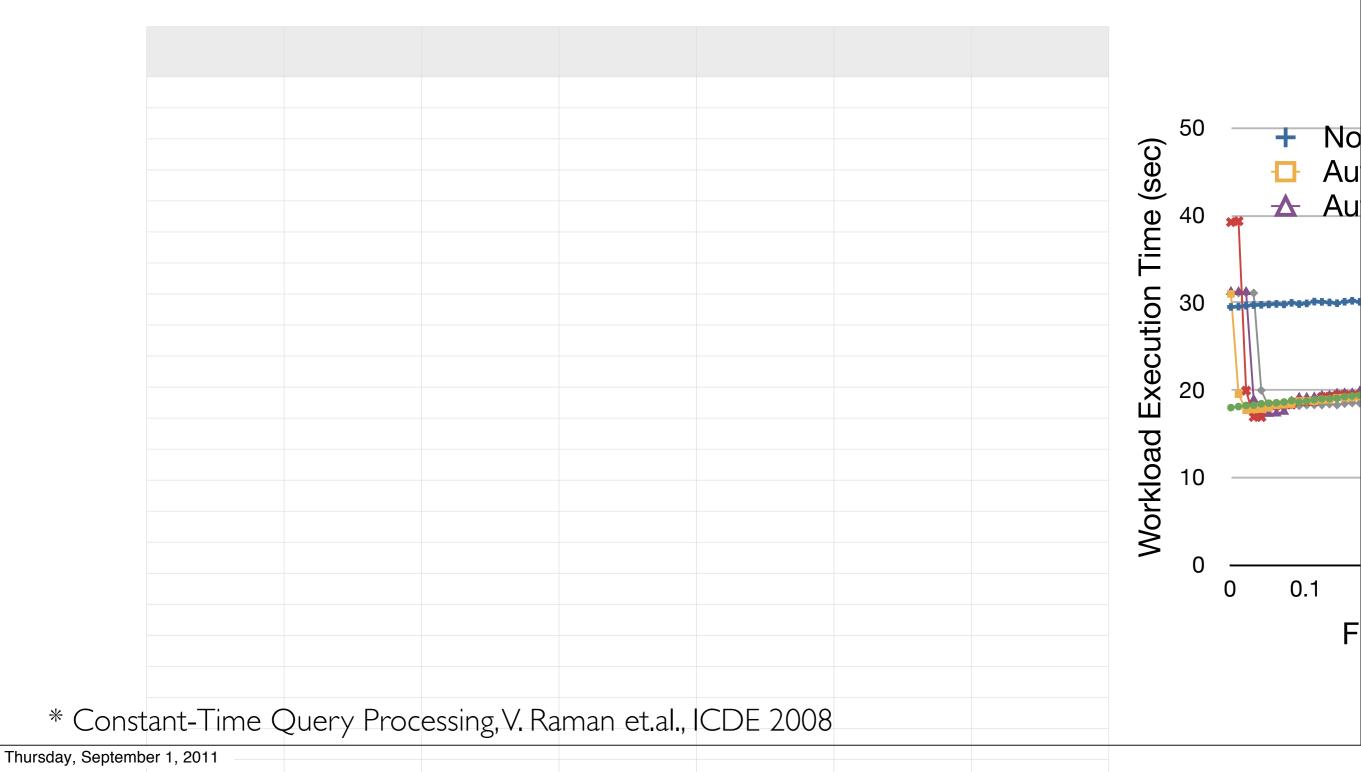
<u>Quality</u>: Ratio of expected query costs of optimal partitioning and the partitioning produced by the algorithm

		Customer			Lineitem	
	Optimal	Navathe	O2P	Optimal	Navathe	O2P
Quality	100%	99.29%	92.76%	100%	97.45%	95.80%
Iterations	100%	14.60%	2.28%	100%	2.42%	0.14%



Can such a System Adapt to Changing Workload?

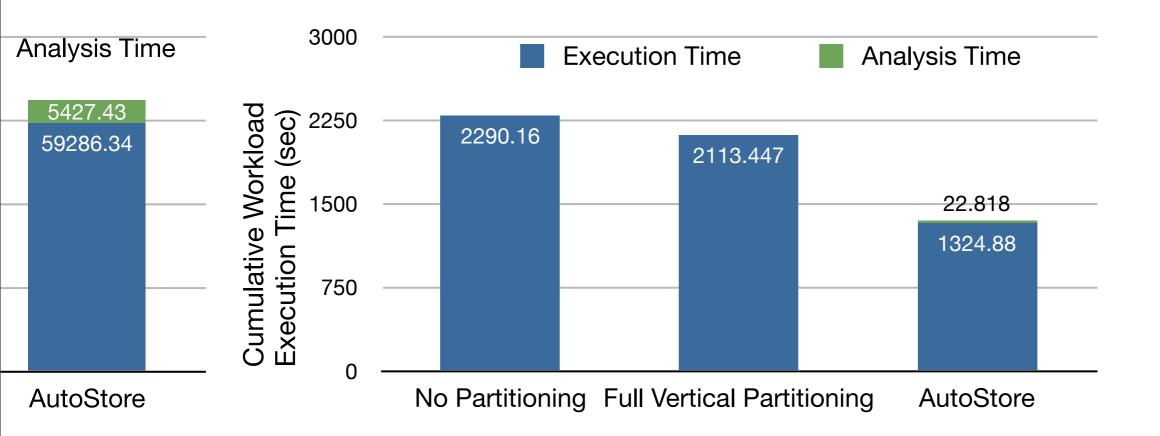
Setup: Universal relation de-normalized from TPC-H schema *. SF 1



*

Will our Approach Work on Real System?

Setup:TPC-H Customer table, SF 1, BerkeleyDB

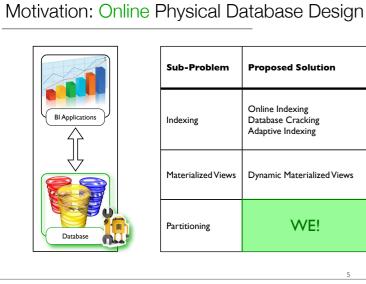


So Whats the Point Again?

Workloads infrequently change over time DDA a aluna available DDI 15 always available Dhucical docian anco in a while rigsical design once in a write 2 MACACC DDA a pool a portant a DDI 13 MARC PERCEL DECISIONS



Summary



Partitioning Problem: What to Analyze?

- Partitioning unit P_u e.g. a1, a2, a3, a4, a5, a6
- P_u ordering \preceq e.g. $a_3 \preceq a_2 \preceq a_1 \preceq a_5 \preceq a_4 \preceq a_6$
- Split line, Split vector $S_{\text{e.g.}}$ [01001]
- Partition $p_{m,r}(S, \preceq)$ e.g. (a1, a2)
- Partitioning scheme $P(S, \preceq)$ e.g. (a1, a2), (a3, a4), (a5, a6)
- Workload W_{t_k}
- Problem statement Find \preceq , S' such that: $S' = \underset{c}{\operatorname{argmin}} C_{\operatorname{est.}} (W_{t_k}, P(S, \preceq))$

Online Partitioning Unit Ordering

• Update *only* the referenced P_u in affinity matrix

	PartKey	Quantity	SuppKey	(PartKey, SuppKey)		PartKey	Quantity	SuppKey
PartKey	8	6	5		PartKey	9	6	6
Quantit	6	9	4	52	Quantity	6	9	4
SuppKe	5	4	8		SuppKey	6	4	9

• Re-cluster only the referenced P_u in affinity matrix

	+48 -	_0=					
	PartKey	Quantity	SuppKey		SuppKey	PartKey	Quantity
PartKey	9	6	6	SuppKey	6	9	6
Quantity	6	9	4	PartKey	4	6	9
SuppKey	6	4	9	Quantity	9	6	4

How much is O₂P Faster? How to Analyze the Workload? Can such a System Adapt to Changing Workload ? Step 2: Enumerating Split Vectors Setup: TPC-H Lineitem table, 10,000 queries in total Setup: Universal relation de-normalized from TPC-H schema, 11 attributes, SF 1 • offline approach: consider all possible split vectors - • NV/HC - • • O2Pp - □ • O2Ppg- × O2pgd • Full Vertical Partitioning (brute force) + No Partitioning □ AutoStore (O2Ppgd) ★ AutoStore (O2Ppgdm Analysis Time (sec) AutoStore (O2Ppgda) • online approach: One-dimensional Online Partitioning (O₂P) Algorithm - prune non-referenced partitioning units - consider split vectors greedily 0.001 - save previous best split vectors using dynamic programming 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 0.2 0.3 0.4 0.5 0.6 0.7 01 Fraction of OLAP against OLTP queries Fraction of OLAP against OLTP queries

0.8