Motivations		Conclusions

L-Store: Lineage-based Storage Architectures ECS165A: Winter 2022

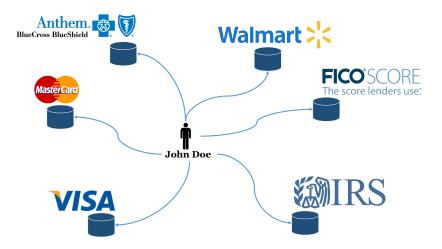
Slides are adopted from Sadoghi, et al. L-Store: A Real-time OLTP and OLAP System, EDBT'18





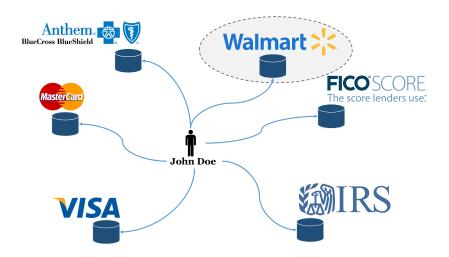




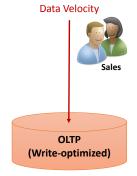




Data Management at Macroscale: The Four V's of Big Data



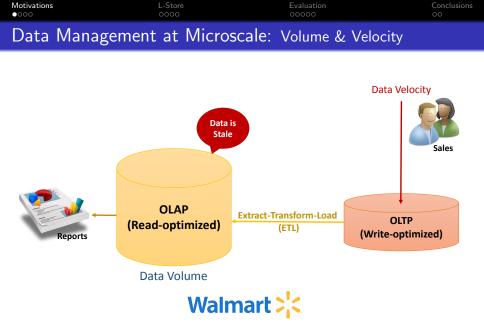
Motivations	L-Store	Evaluation	Conclusions
●○○○	0000	00000	
Data Managemen	t at Microscale:	Volume & Velocity	



イロト イヨト イヨト イヨト

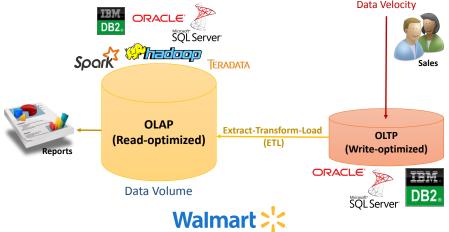


æ



< ロト < 同ト < ヨト < ヨト





< ロト < 同ト < ヨト < ヨト

Motivations	L-Store	Evaluation	Conclusions
0000			
One Size Doe	s not Fit All As	of 2012	

Ad/Media Apps Vertical Apps Business Analytics and Intelligence Visualization Bloomreach. MYRRIX rocketfuel 🔆 + obleov OPalantin ORACLE | Hyperion Ablem Recorded Future OPERA metalawer SAP Business Objects Digned dataspora Log Data Apps METAMARKETS **Di licolin** Microsoft Business Intelligence TERADATA ASTER splunk> loggly sumplogic DataX0 SSAS MTIBCO CARMASHERE IBM COGNOS *birst panopticon Data As A Service kaqqle Autonomy -Datameer bime 📕 factual. кпоета OlikView platfora GoodData alteryx svisually AVATA Analytics Operational Infrastructure As Structured Infrastructure Infrastructure A Service Databases Hortonworks Соиснваяе 10gen 🔙 amazon ORACLE MySOL TERADATA Read and the second sec SQL Server HADAPT EMC O GREENPLUM infochimps IBM DB2. '- TERRACOTTA VoltDB SYBASE N NETEZZA (kognitio MarkLogic INFORMATICA Google BigQuery Ö, DATASTAX MASOL Technologies Aredooo Anzidaiaja HBASE I Cassandra dave@vcdave.com blogs.forbes.com/davefeinleib Copyright © 2012 Dave Feinleib

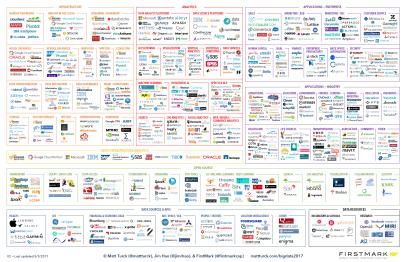
Big Data Landscape

э

イロト イポト イヨト イヨト

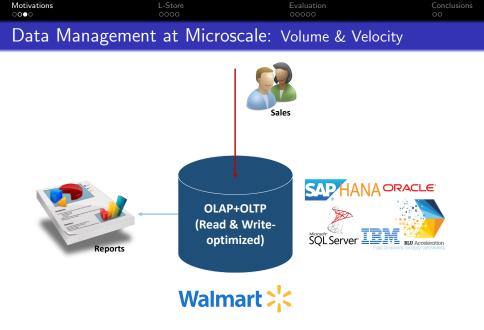
Motivations	L-Store	Evaluation	Conclusions
○●○○	0000	00000	
One Size Does no	t Fit All As of 20)17	

BIG DATA LANDSCAPE 2017

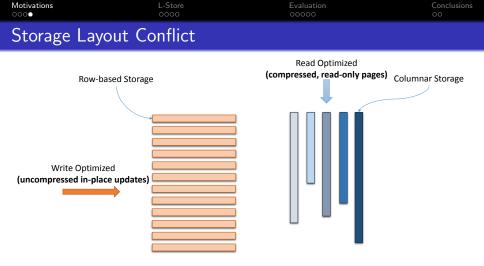


ECS165A

æ



(日) (四) (日) (日) (日)



Write-optimized (i.e., uncompressed & row-based) vs. read-optimized (i.e., compressed & column-based) layouts

Mohammad Sadoghi

Reducing Inde	ex maintenance: V	elocity Dimension	
00000000	0000000	00000	
Indirection	2VCC	Vision	References

In the absence of in-place updates in operational multi-version databases, the cost of index maintenance becomes a major obstacle to cope with data velocity.

Reducing Index maintenance:	Velocity Dimension	
Indirection 2VCC ○○●○○○○○○○ ○○○○○○○○○	Vision 00000	References O

In the absence of in-place updates in operational multi-version databases, the cost of index maintenance becomes a major obstacle to cope with data velocity.

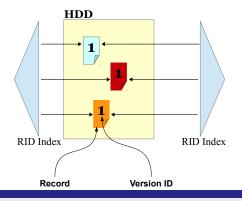
Extending storage hierarchy (using fast non-volatile memory) with *an extra level of indirection* in order to

Reducing	Index maintenance.	Velocity Dimension	
00000000	0000000	00000	
Indirection	2VCC	Vision	References

In the absence of in-place updates in operational multi-version databases, the cost of index maintenance becomes a major obstacle to cope with data velocity.

Extending storage hierarchy (using fast non-volatile memory) with an extra level of indirection in order to Decouple Logical and Physical Locations of Records to Reduce Index Maintenance





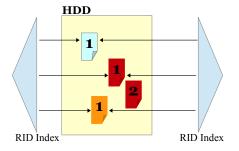
Updating random leaf pages

Mohammad Sadoghi (UC Davis)

ECS165a - 2022

The distance NA D	the second s		
000000	0000000	00000	
Indirection	2VCC	Vision	References

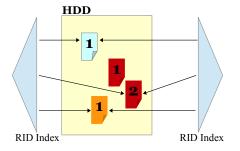






The Present	M http://www.texa.locale.to		
0000000	0000000	00000	
Indirection	2VCC	Vision	References

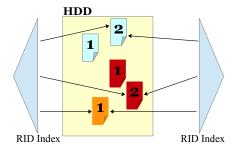






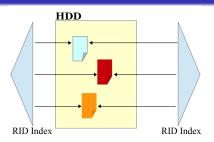
T 11.1 I NA 1.1	1		
000000			
Indirection	2VCC	Vision	References







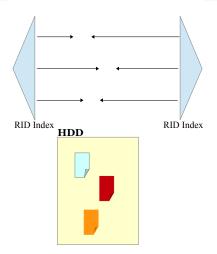
Indirection	Indexing: Updating Records	S	
	2000	Vision 00000	O O



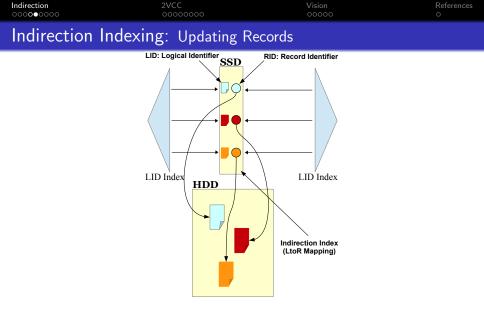
イロト イ部ト イヨト イヨト 二日

Indirection	2VCC	Vision	References
00000000			

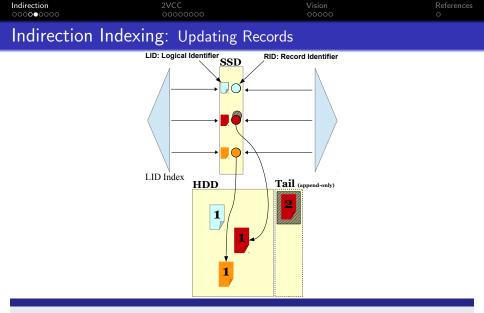
Indirection Indexing: Updating Records



イロト イポト イヨト イヨト



イロト イヨト イヨト イヨト

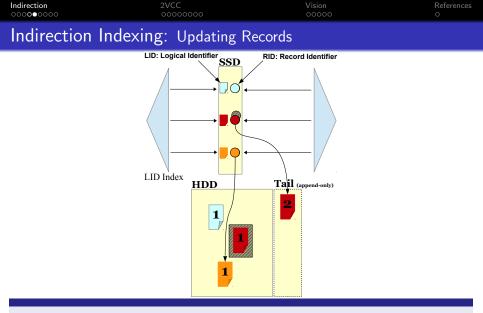


Eliminating random leaf-page updates

Mohammad Sadoghi (UC Davis)

ECS165a - 2022

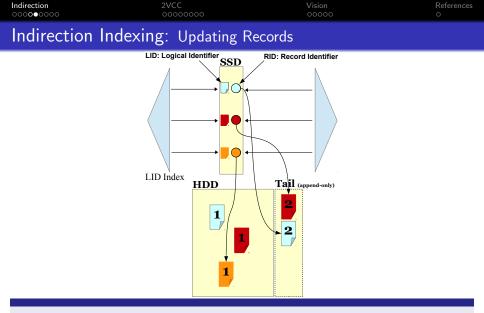
< □ > < □ > < □ > < □ >



Eliminating random leaf-page updates

Mohammad Sadoghi (UC Davis)

ECS165a - 2022



Eliminating random leaf-page updates

Mohammad Sadoghi (UC Davis)

ECS165a - 2022

< □ > < □ > < □ > < □ >

Unifying OLTP	and OLAP.	Velocity & Volume Dim	ensions
	0000		
Motivations	L-Store	Evaluation	Conclusions

In operational databases, there is a pressing need to close the gap between the write-optimized layout for OLTP (i.e., row-wise) and the read-optimized layout for OLAP (i.e., column-wise).

Unifying OLTP	and OLAP.	Velocity & Volume Dim	ensions
	0000		
Motivations	L-Store	Evaluation	Conclusions

In operational databases, there is a pressing need to close the gap between the write-optimized layout for OLTP (i.e., row-wise) and the read-optimized layout for OLAP (i.e., column-wise).

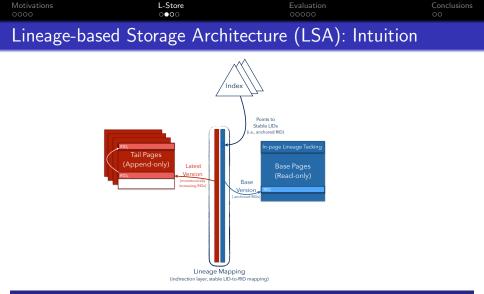
Introducing a *lineage-based storage architecture*, a contention-free update mechanism over a native columnar storage in order to

Unifying OLTP	and OLAP.	Velocity & Volume Dim	ensions
	0000		
Motivations	L-Store	Evaluation	Conclusions

In operational databases, there is a pressing need to close the gap between the write-optimized layout for OLTP (i.e., row-wise) and the read-optimized layout for OLAP (i.e., column-wise).

Introducing a *lineage-based storage architecture*, a contention-free update mechanism over a native columnar storage in order to

lazily and independently stage stable data from a write-optimized layout (i.e., OLTP) into a read-optimized layout (i.e., OLAP)



Physical Update Independence: De-coupling data & its updates (reconstruction via in-page lineage tracking and lineage mapping)

Mohammad Sadoghi

イロト 不得 トイヨト イヨト



Base

Version

Base Pages

(Read-only)

(reconstruction via in-page lineage tracking and lineage mapping)

(Append-only)

Latest

Version

increasing RIDs)

Append-only Updates

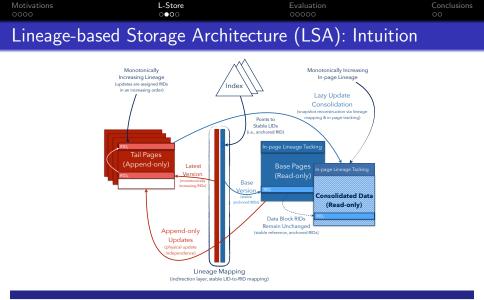
Mohammad Sadoghi

Physical Update Independence: De-coupling data & its updates

Lineage Mapping (indirection layer, stable LID-to-RID mapping)

7/16

イロト 不得 トイヨト イヨト

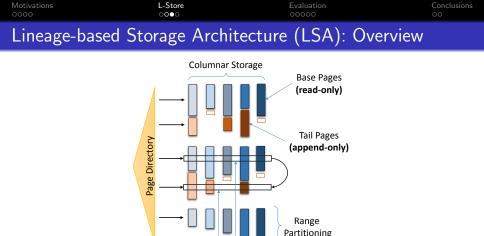


Physical Update Independence: De-coupling data & its updates (reconstruction via in-page lineage tracking and lineage mapping)

Mohammad Sadoghi

7/16

イロト イヨト イヨト

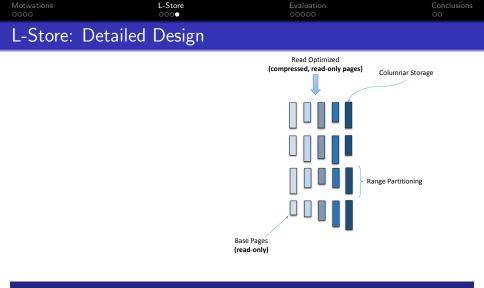


Overview of the lineage-based storage architecture (base pages and tail pages are handled identically at the storage layer)

Record (spanning over a set of aligned columns)

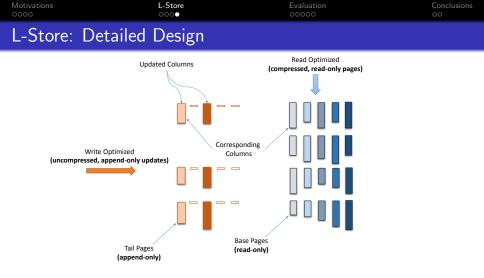
Mohammad Sadoghi

< □ > < 同 > < 回 > < 回 >



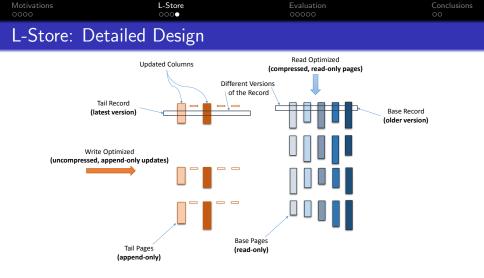
Records are range-partitioned and compressed into a set of ready-only **base pages** (accelerating analytical queries)

	4	ロト・西ト・モト・モト	Ξ.	୬୯୯
Mohammad Sadoghi	ECS165A			9 / 16



Recent updates for a range of records are clustered in their **tails pages** (transforming costly point updates into an amortized analytical-like query)

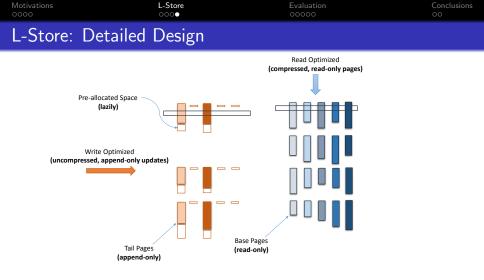
(日) (四) (日) (日) (日)



Recent updates for a range of records are clustered in their **tails pages** (transforming costly point updates into an amortized analytical-like query)

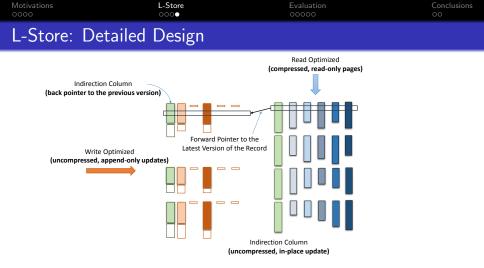
Ma	hammad	Sad	loghi
1010	nannnau	Jau	ogin

(日) (四) (日) (日) (日)



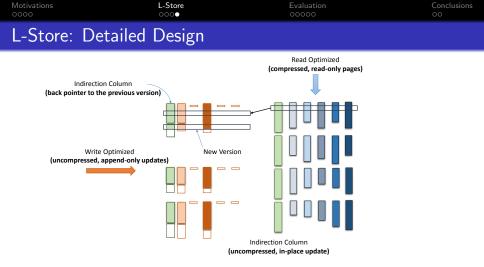
Recent updates are strictly appended, uncompressed in the pre-allocated space (eliminating the read/write contention)

	<.	< ⊡ >	- < ≣ >	★ ≣ ≯	- 2	500
Mohammad Sadoghi	ECS165A					9 / 16



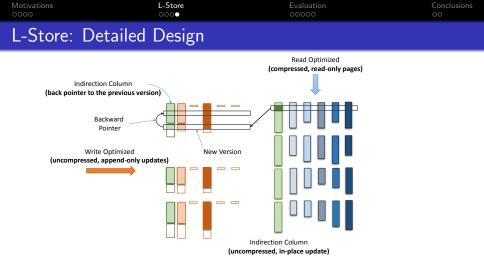
Achieving (at most) 2-hop access to the latest version of any record (avoiding read performance deterioration for point queries)

イロト イポト イヨト イヨト

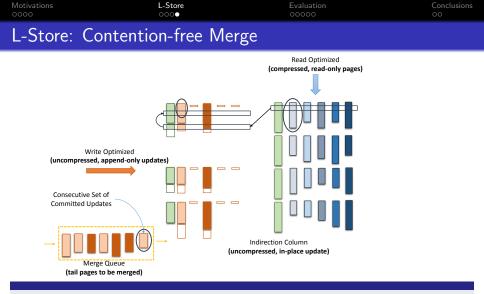


Achieving (at most) 2-hop access to the latest version of any record (avoiding read performance deterioration for point queries)

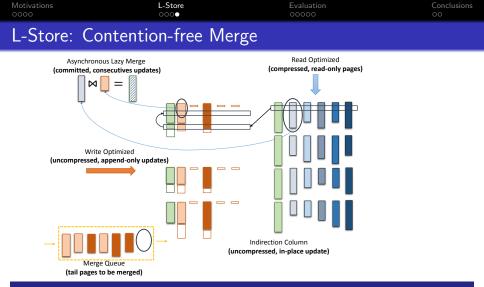
イロト イポト イヨト イヨト



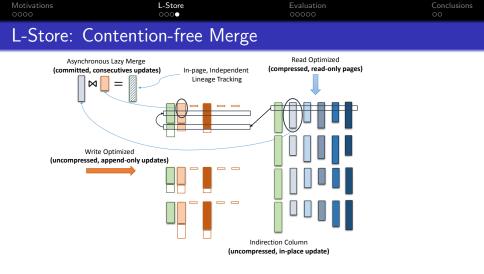
Achieving (at most) 2-hop access to the latest version of any record (avoiding read performance deterioration for point queries)



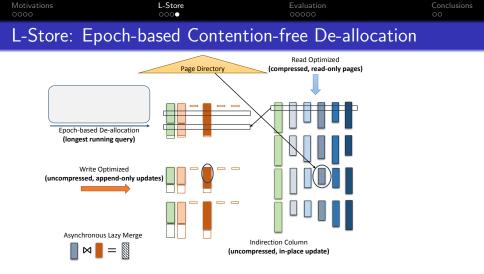
Contention-free merging of only stable data: read-only and committed data (no need to block on-going and new transactions)

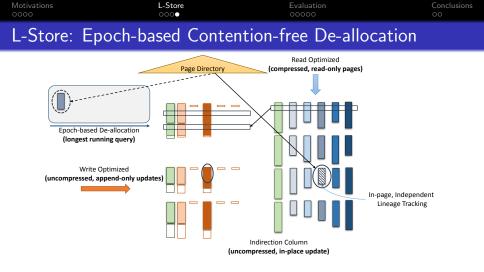


Lazy independent merging of **base pages** with their corresponding **tail pages** (resembling a local left outer-join of the base and tail pages)

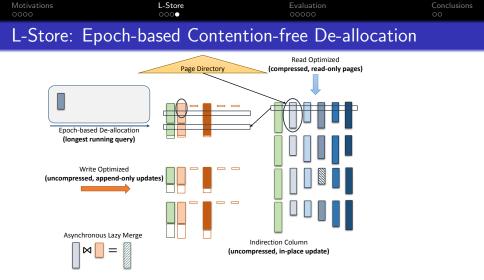


Independently tracking the lineage information within every page (no need to coordinate merges among different columns of the same records)



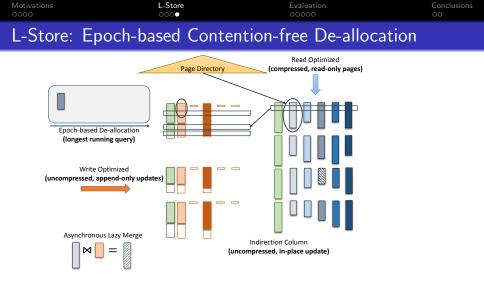


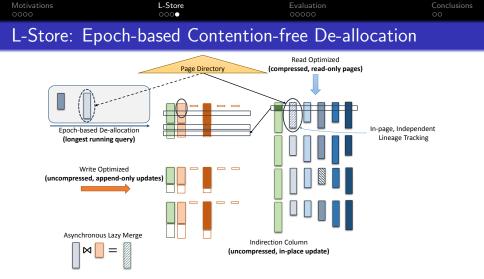
Mohammad Sadoghi



Mohammad Sadoghi

9/16





Mohammad Sadoghi

Motivations	Evaluation	Conclusions
	00000	

Experimental Analysis

æ

3 1 4 3 1

Motivations	L-Store	Evaluation	Conclusions
0000	0000	○●○○○	00
Experimental	Settings		

Hardware:

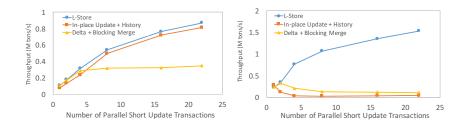
• 2 \times 6-core Intel(R) Xeon(R) CPU E5-2430 @ 2.20GHz, 64GB, 15 MB L3 cache

Workload: Extended Microsoft Hekaton Benchmark

- Comparison with In-place Update + History and Delta + Blocking Merge
- Effect of varying contention levels
- Effect of varying the read/write ratio of short update transactions
- Effect of merge frequency on scan
- Effect of varying the number of short update vs. long read-only transactions
- Effect of varying L-Store data layouts (row vs. columnar)
- Effect of varying the percentage of columns read in point queries
- Comparison with log-structured storage architecture (*LevelDB*)

Motivations	L-Store	Evaluation	Conclusions
0000	0000		00

Effect of Varying Contention Levels



Achieving up to $40 \times$ as increasing the update contention

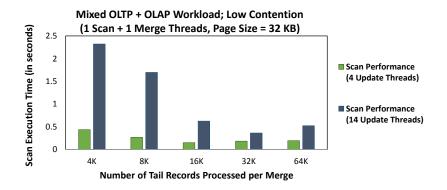
Mohammad Sadoghi

12/16

э

Motivations	L-Store	Evaluation	Conclusions
0000	0000	○○○●○	00
= ~ ~ ~ ~	_		

Effect of Merge Frequency on Scan Performance



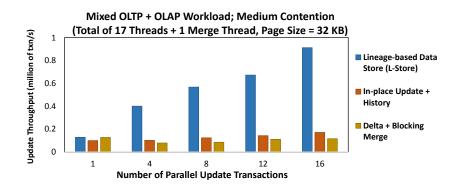
Merge process is essential in maintaining efficient scan performance

Mohammad Sadoghi

ECS165A

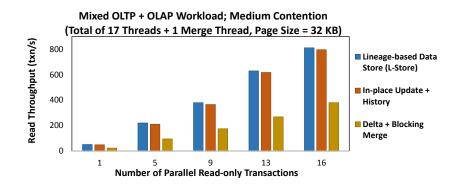
< ロト < 同ト < ヨト < ヨト

Motivations	L-Store	Evaluation	Conclusions
0000	0000	○○○○●	00
Effect of Mixed Work	loads: Update Perfo	rmance	



Eliminating latching & locking results in a substantial performance improvement

Motivations	L-Store	Evaluation	Conclusions
0000	0000	○○○○●	00
Effect of Mixed Work	loads: Read Perform	iance	



Coping with tens of update threads with a single merge thread

Mohammad Sadoghi

ECS165A

Motivations	L-Store	Evaluation	Conclusions
0000	0000	00000	●0
L-Store Kev C	ontributions		

- Unifying OLAP & OLTP by introducing lineage-based storage architecture (LSA)
- LSA is a native multi-version, columnar storage model that lazily & independently stages data from a write-optimized layout into a read-optimized one
- Contention-free merging of only stable data without blocking ongoing or incoming transactions
- Contention-free page de-allocation without draining ongoing transactions
- L-Store outperforms in-place update & delta approaches by factor of up to **8**× on mixed OLTP/OLAP workloads and up to **40**× on update-intensive workloads

Motivations		Conclusions
		00

Questions? Thank you!

Exploratory Systems Lab (ExpoLab) Website: https://expolab.org/



