Motivations	L-Store	Evaluation	Conclusions

L-Store: A Real-time OLTP and OLAP System

Mohammad Sadoghi[†] Souvik Bhattacharjee[‡], Bishwaranjan Bhattacharjee[#], Mustafa Canim[#]

[†]Exploratory Systems Lab
 [†] University of California, Davis
 [‡] University of Maryland, College Park
 [#] IBM T.J. Watson
 EDBT'18 (March 27, 2018)





IBM Research

















Image: Image:



< A

∃ ▶ ∢





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

Big Data Landscape Ad/Media Apps Vertical Apps Business Analytics and PREDICTIVE POLICING Visualization Intelligence rocketfuel Bloomreach. MYRRIX 🔆 + obleov 🛛 📿 Palantir ORACLE | Hyperion Abluefin Accorded Future OPERA metaLoyer SAP Business Objects Business dataspora Log Data Apps TH METAMARKETS (J) LUCKYSORT Microsoft Business Intelligence centrifupe TERADATA ASTER Media Science TURN Data XU splunk> loggly sumologic SSAS TIBCO' IBM COGNOS *birst panopticon Data As A Service Autonomy Datameer factual bime 📕 ĸnoema™ QlikView platfora CIRRO GNIP MASTI "Window Aster INRIX @ LexisNexis" Static LOGATE GoodData alteryx svisual.ly AVATA Chart.io Analytics Operational Infrastructure As Structured Infrastructure Infrastructure A Service Databases Hortonworks ORACLE MySQL Соцснвазе 10gen m amazon INFOBRIGHT cloudera PARAccel. SQL Server HADAPT Mindows Azure EMC @ GREENPLUM. infochimps IBM. DB2. TERRACOTTA VoltDE N NETEZZA 6 kognitio INFORMATICA MarkLogic 🔘 mem 💅 Google BigQuery

SYBASE **Technologies** An=dooo hedoop HBASE 🥨 Cassandra dave@vcdave.com blogs.forbes.com/davefeinleib Copyright @ 2012 Dave Feinleib

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

BIG DATA LANDSCAPE 2017



Mohammad Sadoghi (UC Davis)

イロト イヨト イヨト イヨト



< ∃ > <



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

Mohammad Sadoghi (UC Davis)

Unifying OLTE	and OLAP.	Volocity & Volumo Din	oncions
0000	0000	0000	00
Motivations	L-Store	Evaluation	Conclusions

Observed Trends

ъ

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 OLTE	and OLAP.	Volocity & Volumo Din	oncions
0000	0000	0000	00
Motivations	L-Store	Evaluation	Conclusions

Observed Trends

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 OLTE	and OLAP.	Volocity & Volumo Din	oncions
0000	0000	0000	00
Motivations	L-Store	Evaluation	Conclusions

Observed Trends

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)





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

< 口 > < 同





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





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

Mohammad Sadoghi (UC Davis)



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

Mohammad Sadoghi (UC Davis)



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

<ロト </p>



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

Mohammad Sadoghi (UC Davis)

<ロト </p>



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

Mohammad Sadoghi (UC Davis)



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

Mohammad Sadoghi (UC Davis)



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

Mohammad Sadoghi (UC Davis)



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

Mohammad Sadoghi (UC Davis)



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

Mohammad Sadoghi (UC Davis)



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

Mohammad Sadoghi (UC Davis)

イロト イ団ト イヨト イヨト



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

Mohammad Sadoghi (UC Davis)



Mohammad Sadoghi (UC Davis)



Mohammad Sadoghi (UC Davis)



Mohammad Sadoghi (UC Davis)



Mohammad Sadoghi (UC Davis)



Mohammad Sadoghi (UC Davis)

Motivations	L-Store	Evaluation	Conclusions

Experimental Analysis

		0 000	00
Experimental Sett	ings		

Hardware:

■ 2 × 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 0000		L-S	Store			E	Evaluation ⊃●○○	Conclusions 00
		_						

Effect of Varying Contention Levels



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

• • • • • • • • • • • •

-

EDBT'18

12 / 16



Effect of Merge Frequency on Scan Performance



Merge process is essential in maintaining efficient scan performance

EDBT'18

13 / 16





Eliminating latching & locking results in a substantial performance improvement

Mohammad Sadoghi (UC Davis)

EDBT'18 14 / 16





Coping with tens of update threads with a single merge thread

A B > A B >

EDBT'18

14 / 16

0000	0000	0000	0
L-Store Kev Contr	ributions		

- 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
0000	0000	0000	00

Questions? Thank you!

Exploratory Systems Lab (ExpoLab) Website: https://msadoghi.github.io/

