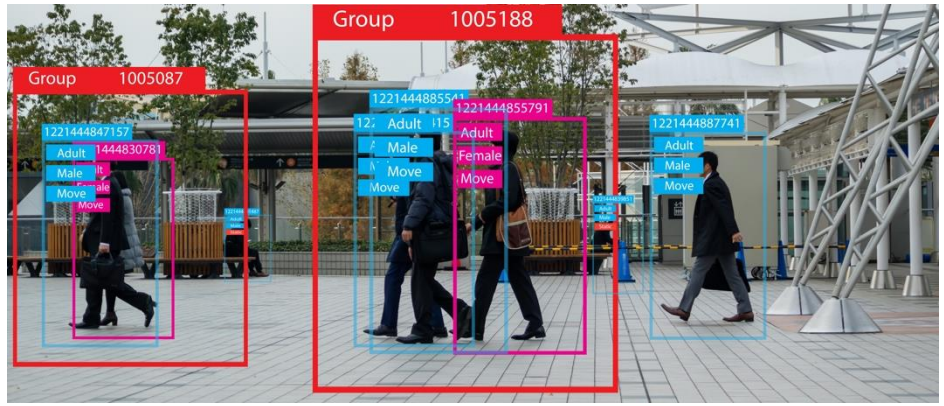
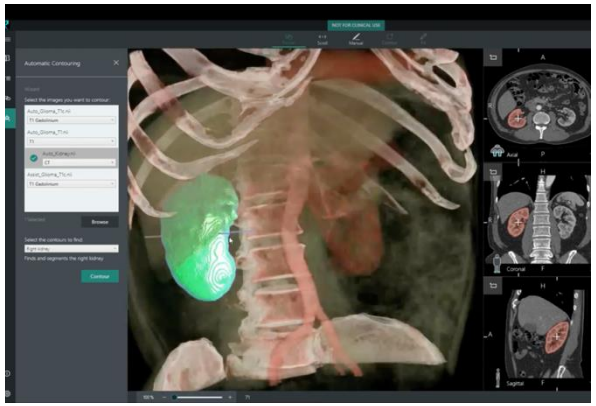
ALEX– An adaptive learned index for dynamic workloads

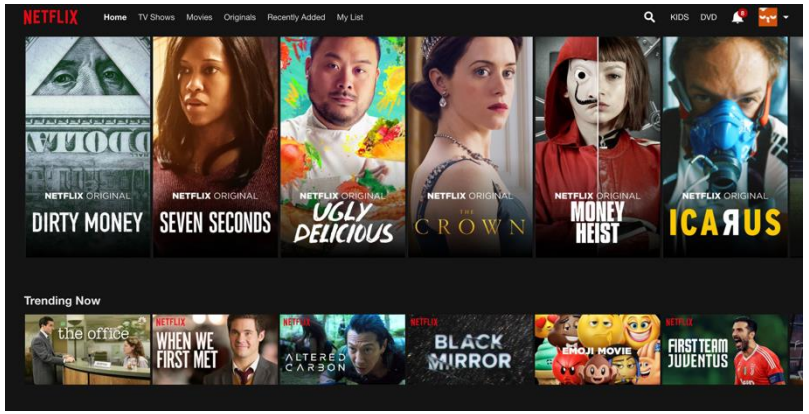
UMAR FAROOQ MINHAS, JIALIN DING, DONALD KOSSMANN, JOHANNES GEHRKE, BADRISH CHANDRAMOULI, YINAN LI, DAVID LOMET, CHI WANG, JAE DO, TIM KRASKA

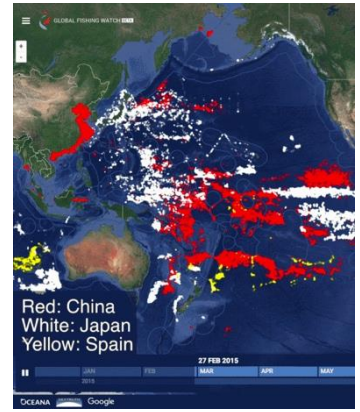
MICROSOFT RESEARCH 2019

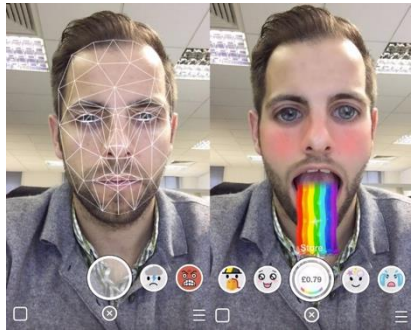
“AI” today: Perceptual and NL tasks

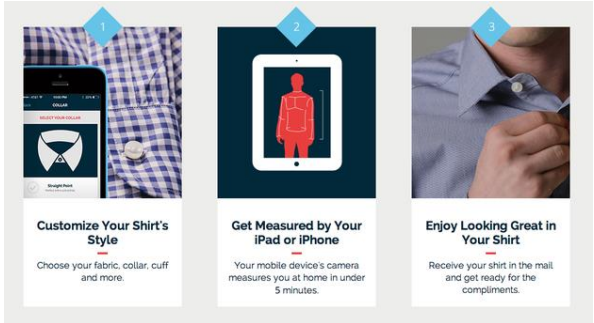




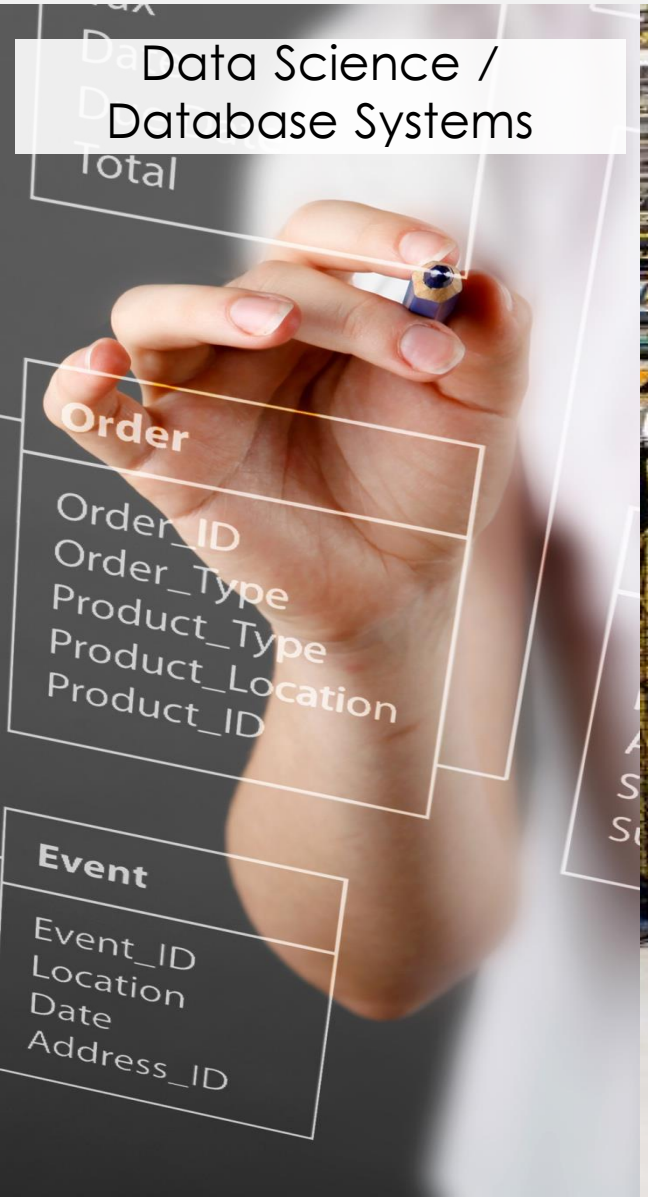








“AI” tomorrow: + Systems

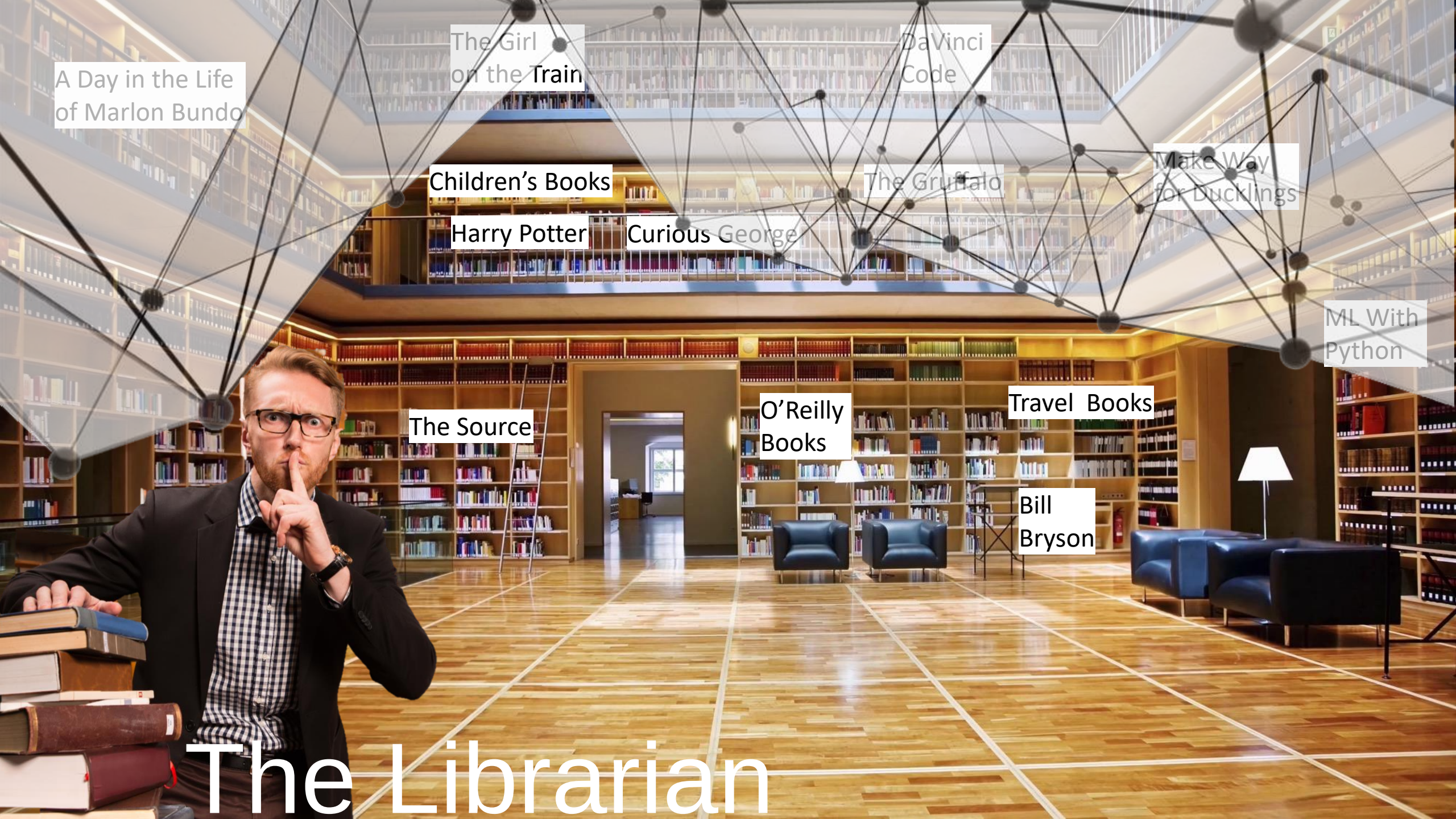


Fundamental building blocks

Fundamental building blocks


B-C
C-G
G-J
K-N
N-R
S-U
U-V
V-X
X-@
Q-S

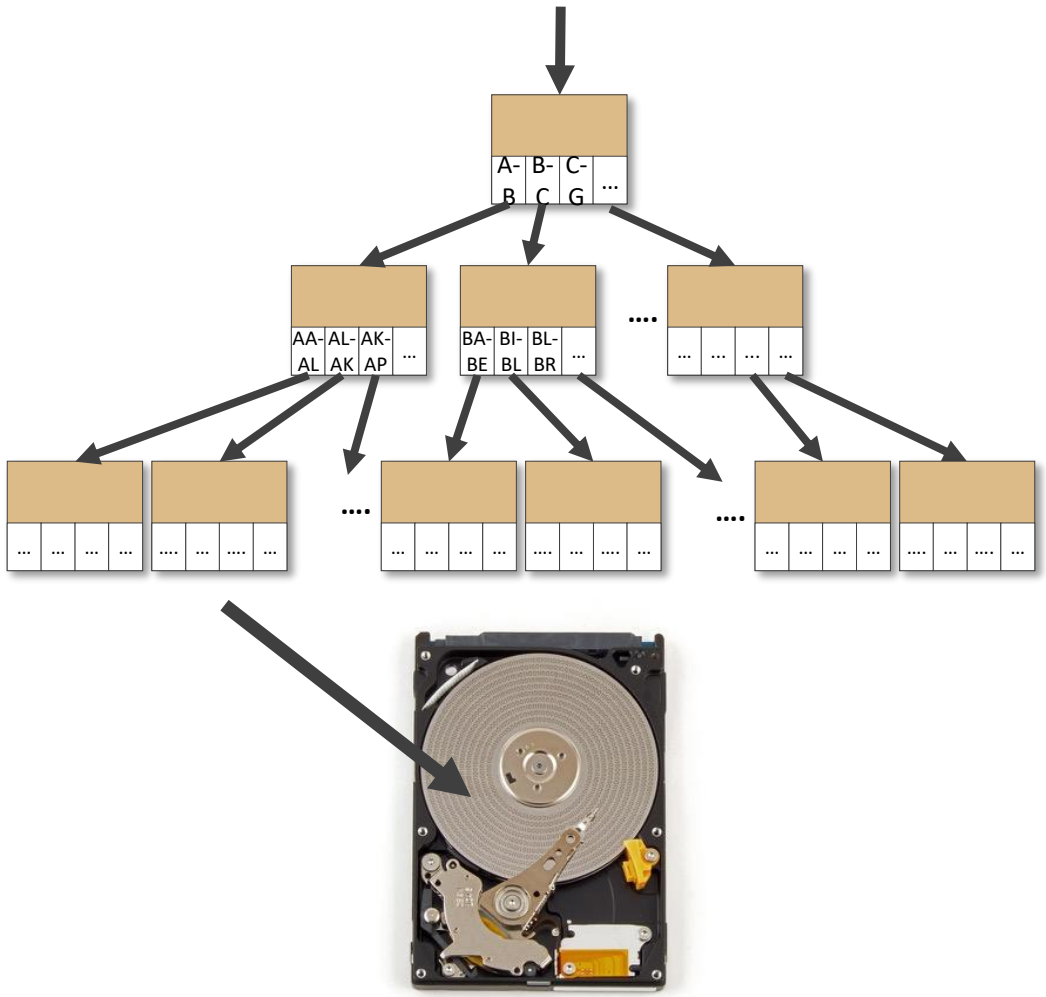

B-C
C-G
G-J
K-N
N-R


A Day in the Life of Marlon Bundo
The Girl on the Train
DaVinci Code
Children's Books
Harry Potter
Curious George
The Gruffalo
Make Way for Ducklings
ML With Python
The Source
O'Reilly Books
Travel Books
Bill Bryson
The Librarian

B-Tree

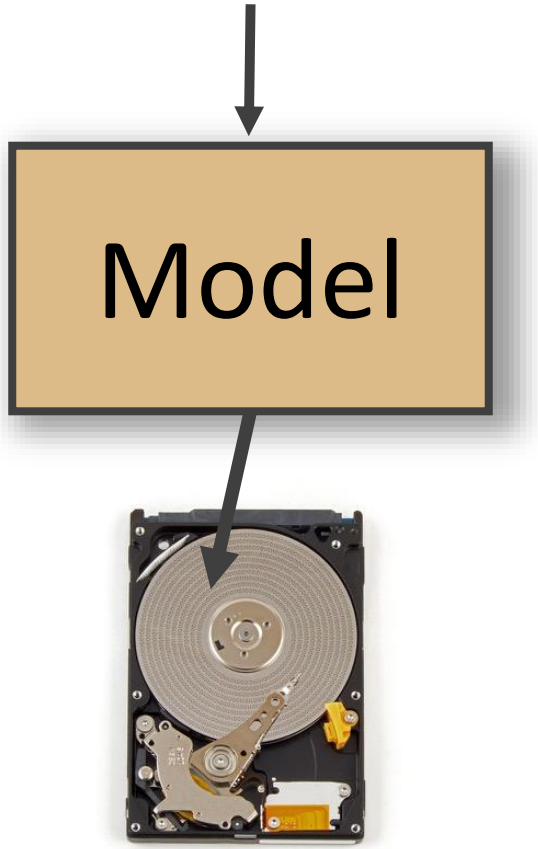
Key

(e.g., book title, author,…)



Learned Index

Key
(e.g., book title, author,…)



Model predicts the location of the data
like the librarian predicts the location of the book

How to build models to predict the location

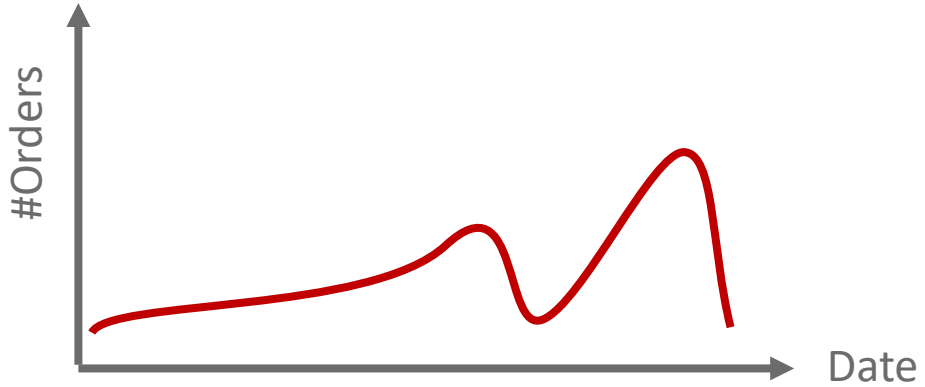
id	date	first_name	last_name	email	address	zip	state	credit_card_nb	amount
1000	2017-01-01	Hobart	Spracklin	hspracklin0@dailymotion.com	20565 High Crossing Plaza	56372	Minnesota	4405-6975-7285-5160	$ 611.00
1001	2017-01-02	Billye	Binnion	bbinnion1@123-reg.co.uk	3698 Upham Point	20260	District of Columbia	3533-7150-7728-9850	$ 244.00
1002	2017-01-02	Johann	Brockley	jbrockley2@bizjournals.com	23844 Artisan Place	98516	Washington	67597-1193-7985-5100	$ 233.00
1003	2017-01-03	Artie	MacMenami	amacmenamin3@hao123.com	6276 Toban Trail	78759	Texas	3537-4829-6134-5000	$ 210.00
1004	2017-01-03	Delilah	O'Currigan	docurrigan4@chron.com	86016 New Castle Avenue	72199	Arkansas	3555-2017-2226-5780	$ 286.00
1005	2017-01-04	Gretta	Will	gwill5@yelp.com	0 Dottie Circle	68524	Nebraska	503844-1984-2085-5000	$ 870.00
1006	2017-01-04	Gordon	Kirsopp	gkirsopp6@utexas.edu	64060 Scott Park	20370	District of Columbia	633332-1895-2414-5000	$ 687.00
1007	2017-01-05	Bendick	Fagg	bfagg7@army.mil	94 Florence Hill	45440	Ohio	3528-9673-1815-8420	$ 733.00
1008	2017-01-05	Dimitry	Boyet	dboyet8@sakura.ne.jp	35886 Golf Plaza	30066	Georgia	3576-6991-4041-3170	$ 382.00
1009	2017-01-06	Ailsun	Beinke	abeinke9@si.edu	1 Badeau Place	46295	Indiana	56022-2011-8072-1400	$ 854.00
1010	2017-01-07	Lou	Hallows	lhallowsa@theguardian.com	1 Twin Pines Junction	91125	California	5602-2364-4079-0250	$ 150.00
1011	2017-01-09	Tiffani	Mathew	tmathewb@seattletimes.com	0456 Meadow Vale Lane	75260	Texas	6387-6943-8910-4580	$ 313.00
1012	2017-01-09	Perl	Bridie	pbridiec@hubpages.com	07 Bluestem Junction	33124	Florida	3539-8662-2397-5880	$ 558.00
1013	2017-01-09	Rosabelle	Blasik	rblasikd@delicious.com	7 Fairfield Pass	79699	Texas	5602-2297-6599-8560	$ 941.00
1014	2017-01-10	Meggi	Belamy	mbelamye@ask.com	0995 Manufacturers Street	10170	New York	3557-5094-7405-8340	$ 875.00
1015	2017-01-10	Tadio	Balderston	tbalderstonf@apache.org	80 Novick Road	75260	Texas	60485-3728-7119-9300	$ 954.00
1016	2017-01-11	Gianina	Oxteby	goxtebyg@google.pl	72674 Fuller Avenue	89505	Nevada	4-0415-9268-2397	$ 239.00
1017	2017-01-12	Brendan	Doody	bdoodyh@craigslist.org	87414 Golden Leaf Street	11480	New York	201-6348-4121-1314	$ 308.00
1018	2017-01-13	Conway	Coombs	ccoombsi@blogger.com	2810 Oakridge Park	32859	Florida	3529-1514-0357-9120	$ 60.00
1019	2017-01-14	Germaine	Bere	gberej@bravesites.com	82802 Oakridge Park	20041	District of Columbia	670961-0240-4054-9000	$ 95.00
1020	2017-01-15	Davide	Tolcharde	dtolchardek@redcross.org	89 Continental Avenue	79165	Texas	5018-7748-4325-9510	$ 137.00
1021	2017-01-16	Nigel	Artharg	narthargl@gizmodo.com	31 Mcbride Point	22301	Virginia	560225-6965-2870-0000	$ 496.00
1022	2017-01-17	Rickard	Trenholm	rtrenholmm@cbslocal.com	93 Hoepker Parkway	70593	Louisiana	3541-5241-5383-9970	$ 760.00
1023	2017-01-18	Juditha	Dwane	jdwanen@vk.com	7914 Eliot Lane	14276	New York	5456-4410-0914-3180	$ 474.00
1024	2017-01-19	Susan	Ilden	sildeno@aol.com	25204 Huxley Road	21684	Maryland	3574-8586-6367-9920	$ 83.00
1025	2017-01-20	Abbey	Triggle	atrigglep@google.com.au	47 Debra Pass	74184	Oklahoma	3538-6047-6315-7710	$ 513.00
1026	2017-01-21	Zsazsa	Dunster	zdunsterq@nature.com	7 Gerald Alley	40576	Kentucky	3562-0325-7709-3490	$ 952.00
1027	2017-01-22	Grantham	Friatt	gfriattr@seattletimes.com	774 Prairieview Circle	29225	South Carolina	3571-1171-9476-8780	$ 942.00
1028	2017-01-22	Ross	Gaudin	rgaudins@samsung.com	3102 Loeprich Trail	68197	Nebraska	5108-7578-4665-2710	$ 572.00
1029	2017-01-22	Aluino	Drover	adrovert@dagondesign.com	2717 Northridge Avenue	72199	Arkansas	670999-3171-8848-0000	$ 318.00
1030	2017-01-23	Shurlock	Braker	sbrakeru@huffingtonpost.com	30783 Jenna Alley	80945	Colorado	6331106-1894-9878-0000	$ 166.00
1031	2017-01-24	Glenda	Goodbody	ggoodbodyv@economist.com	720 Pierstorff Way	7522	New Jersey	36-0593-2719-1684	$ 412.00
1032	2017-01-24	Rollin	Reddie	rreddiew@tinypic.com	09 Gina Park	65810	Missouri	4665-9188-1324-1040	$ 383.00
1033	2017-01-26	Dorry	Jenks	djenksx@virginia.edu	1 Butterfield Road	85210	Arizona	3578-9195-0297-7730	$ 636.00
1034	2017-01-26	Patti	Emby	pembyy@weather.com	26 Hoard Drive	91210	California	3585-8243-7506-2470	$ 957.00

How to build models to predict the location

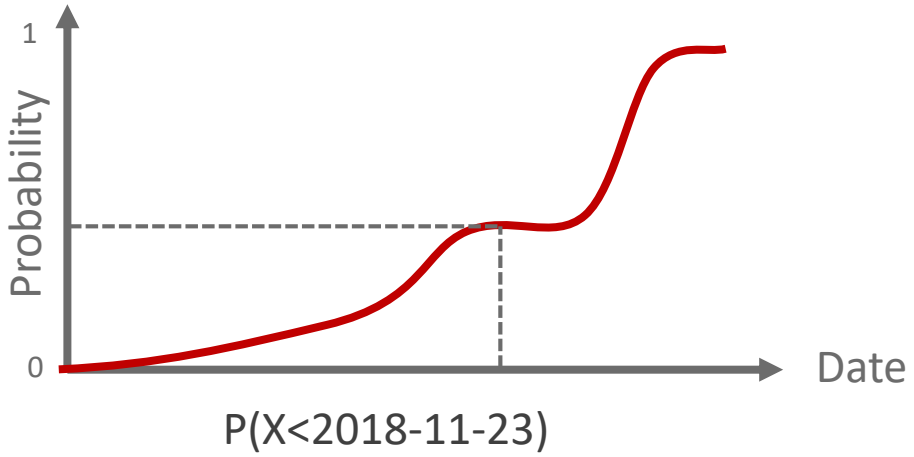
date
2017-01-01
2017-01-02
2017-01-02
2017-01-03
2017-01-03
2017-01-04
2017-01-04
2017-01-05
2017-01-05
2017-01-06
2017-01-07
2017-01-09
2017-01-09
2017-01-09
2017-01-10
2017-01-10
2017-01-11
2017-01-12
2017-01-13
2017-01-14
2017-01-15
2017-01-16
2017-01-17
2017-01-18
2017-01-19
2017-01-20
2017-01-21
2017-01-22
2017-01-22
2017-01-22
2017-01-23
2017-01-24
2017-01-24
2017-01-26
2017-01-26

How to build models to predict the location

Frequency Distribution



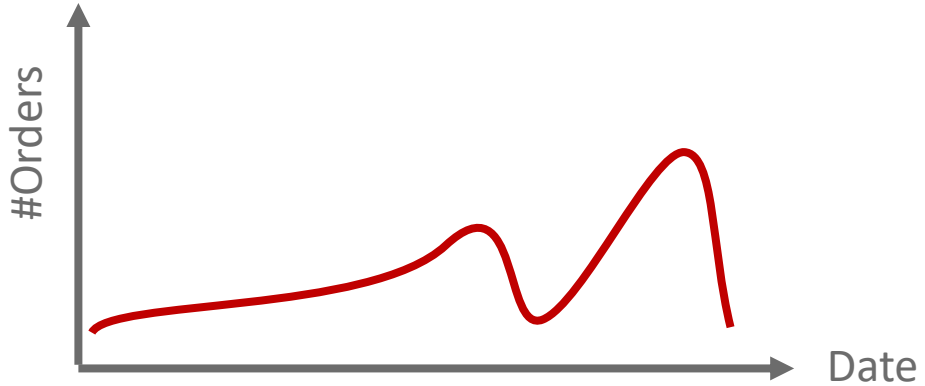
Cumulative Distribution Function (CDF)



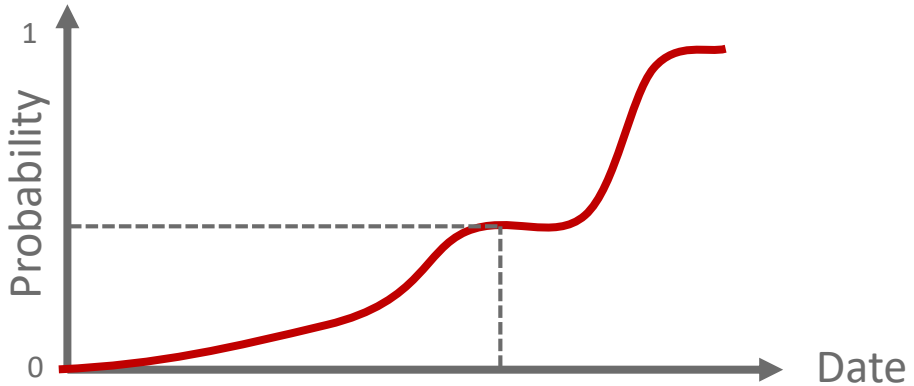
date
2017-01-01
2017-01-02
2017-01-02
2017-01-03
2017-01-03
2017-01-04
2017-01-04
2017-01-05
2017-01-05
2017-01-06
2017-01-07
2017-01-09
2017-01-09
2017-01-09
2017-01-10
2017-01-10
2017-01-11
2017-01-12
2017-01-13
2017-01-14
2017-01-15
2017-01-16
2017-01-17
2017-01-18
2017-01-19
2017-01-20
2017-01-21
2017-01-22
2017-01-22
2017-01-22
2017-01-23
2017-01-24
2017-01-24
2017-01-26
2017-01-26
2017-01-26
2017-01-28
2017-01-29
2017-01-30
2017-01-30
2017-01-30
2017-01-31
2017-01-31
2017-02-01
2017-02-01
2017-02-02
2017-02-04
2017-02-05
2017-02-05
2017-02-06
2017-02-06
2017-02-06
2017-02-07
2017-02-07
2017-02-08
2017-02-08
2017-02-08
2017-02-09
2017-02-10
2017-02-10
2017-02-11
2017-02-12
2017-02-13
2017-02-13
2017-02-14
2017-02-14
2017-02-15

How to build models to predict the location

Frequency Distribution



Cumulative Distribution Function (CDF)



P(X<2018-11-23) * N

date
2017-01-01
2017-01-02
2017-01-02
2017-01-03
2017-01-03
2017-01-04
2017-01-04
2017-01-05
2017-01-05
2017-01-06
2017-01-07
2017-01-09
2017-01-09
2017-01-09
2017-01-10
2017-01-10
2017-01-11
2017-01-12
2017-01-13
2017-01-14
2017-01-15
2017-01-16
2017-01-17
2017-01-18
2017-01-19
2017-01-20
2017-01-21
2017-01-22
2017-01-22
2017-01-22
2017-01-23
2017-01-24
2017-01-24
2017-01-26
2017-01-26
2017-01-26
2017-01-28
2017-01-29
2017-01-30
2017-01-30
2017-01-30
2017-01-31
2017-01-31
2017-02-01
2017-02-01
2017-02-02
2017-02-04
2017-02-05
2017-02-05
2017-02-06
2017-02-06
2017-02-06
2017-02-07
2017-02-07
2017-02-08
2017-02-08
2017-02-08
2017-02-09
...
2017-11-27
2017-11-27
2017-11-27
2017-11-28
2017-11-28
...

Why are learned indexes so good?

B-tree is designed for the *average* case

Learned index adapts to the *specific* instance

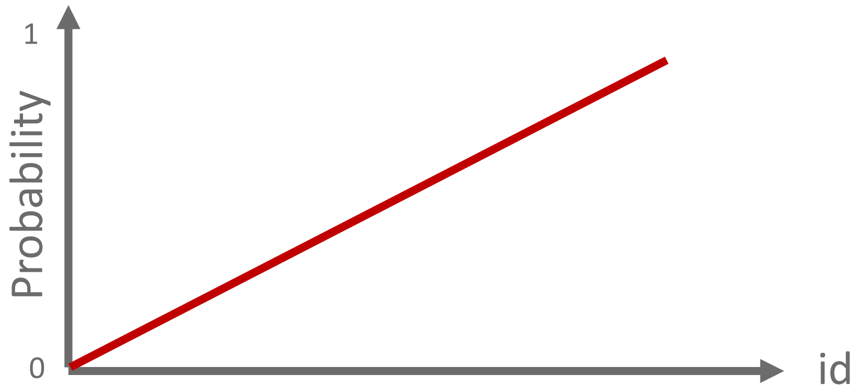
Instance-specific optimizaton

id	date	first_name	last_name	email	address	zip	state	credit_card_nb	amount
1000	2017-01-01	Hobart	Spracklin	hspracklin0@dailymotion.com	20565 High Crossing Plaza	56372	Minnesota	4405-6975-7285-5160	$ 611.00
1001	2017-01-02	Billye	Binnion	bbinnion1@123-reg.co.uk	3698 Upham Point	20260	District of Columbia	3533-7150-7728-9850	$ 244.00
1002	2017-01-02	Johann	Brockley	jbrockley2@bizjournals.com	23844 Artisan Place	98516	Washington	67597-1193-7985-5100	$ 233.00
1003	2017-01-03	Artie	MacMenami	amacmenamin3@hao123.com	6276 Toban Trail	78759	Texas	3537-4829-6134-5000	$ 210.00
1004	2017-01-03	Delilah	O'Currigan	docurrigan4@chron.com	86016 New Castle Avenue	72199	Arkansas	3555-2017-2226-5780	$ 286.00
1005	2017-01-04	Gretta	Will	gwill5@yelp.com	0 Dottie Circle	68524	Nebraska	503844-1984-2085-5000	$ 870.00
1006	2017-01-04	Gordon	Kirsopp	gkirsopp6@utexas.edu	64060 Scott Park	20370	District of Columbia	633332-1895-2414-5000	$ 687.00
1007	2017-01-05	Bendick	Fagg	bfagg7@army.mil	94 Florence Hill	45440	Ohio	3528-9673-1815-8420	$ 733.00
1008	2017-01-05	Dimitry	Boyet	dboyet8@sakura.ne.jp	35886 Golf Plaza	30066	Georgia	3576-6991-4041-3170	$ 382.00
1009	2017-01-06	Ailsun	Beinke	abeinke9@si.edu	1 Badeau Place	46295	Indiana	56022-2011-8072-1400	$ 854.00
1010	2017-01-07	Lou	Hallows	lhallowsa@theguardian.com	1 Twin Pines Junction	91125	California	5602-2364-4079-0250	$ 150.00
1011	2017-01-09	Tiffani	Mathew	tmathewb@seattletimes.com	0456 Meadow Vale Lane	75260	Texas	6387-6943-8910-4580	$ 313.00
1012	2017-01-09	Perl	Bridie	pbridiec@hubpages.com	07 Bluestem Junction	33124	Florida	3539-8662-2397-5880	$ 558.00
1013	2017-01-09	Rosabelle	Blasik	rblasikd@delicious.com	7 Fairfield Pass	79699	Texas	5602-2297-6599-8560	$ 941.00
1014	2017-01-10	Meggi	Belamy	mbelamye@ask.com	0995 Manufacturers Street	10170	New York	3557-5094-7405-8340	$ 875.00
1015	2017-01-10	Tadio	Balderston	tbalderstonf@apache.org	80 Novick Road	75260	Texas	60485-3728-7119-9300	$ 954.00
1016	2017-01-11	Gianina	Oxteby	goxtebyg@google.pl	72674 Fuller Avenue	89505	Nevada	4-0415-9268-2397	$ 239.00
1017	2017-01-12	Brendan	Doody	bdoodyh@craigslist.org	87414 Golden Leaf Street	11480	New York	201-6348-4121-1314	$ 308.00
1018	2017-01-13	Conway	Coombs	ccoombsi@blogger.com	2810 Oakridge Park	32859	Florida	3529-1514-0357-9120	$ 60.00
1019	2017-01-14	Germaine	Bere	gberej@bravesites.com	82802 Oakridge Park	20041	District of Columbia	670961-0240-4054-9000	$ 95.00
1020	2017-01-15	Davide	Tolcharde	dtolchardek@redcross.org	89 Continental Avenue	79165	Texas	5018-7748-4325-9510	$ 137.00
1021	2017-01-16	Nigel	Artharg	narthargl@gizmodo.com	31 Mcbride Point	22301	Virginia	560225-6965-2870-0000	$ 496.00
1022	2017-01-17	Rickard	Trenholm	rtrenholmm@cbslocal.com	93 Hoepker Parkway	70593	Louisiana	3541-5241-5383-9970	$ 760.00
1023	2017-01-18	Juditha	Dwane	jdwanen@vk.com	7914 Eliot Lane	14276	New York	5456-4410-0914-3180	$ 474.00
1024	2017-01-19	Susan	Ilden	sildeno@aol.com	25204 Huxley Road	21684	Maryland	3574-8586-6367-9920	$ 83.00
1025	2017-01-20	Abbey	Triggle	atrigglep@google.com.au	47 Debra Pass	74184	Oklahoma	3538-6047-6315-7710	$ 513.00
1026	2017-01-21	Zsazsa	Dunster	zdunsterq@nature.com	7 Gerald Alley	40576	Kentucky	3562-0325-7709-3490	$ 952.00
1027	2017-01-22	Grantham	Friatt	gfriattr@seattletimes.com	774 Prairieview Circle	29225	South Carolina	3571-1171-9476-8780	$ 942.00
1028	2017-01-22	Ross	Gaudin	rgaudins@samsung.com	3102 Loeprich Trail	68197	Nebraska	5108-7578-4665-2710	$ 572.00
1029	2017-01-22	Aluino	Drover	adrovert@dagondesign.com	2717 Northridge Avenue	72199	Arkansas	670999-3171-8848-0000	$ 318.00
1030	2017-01-23	Shurlock	Braker	sbrakeru@huffingtonpost.com	30783 Jenna Alley	80945	Colorado	6331106-1894-9878-0000	$ 166.00
1031	2017-01-24	Glenda	Goodbody	ggoodbodyv@economist.com	720 Pierstorff Way	7522	New Jersey	36-0593-2719-1684	$ 412.00
1032	2017-01-24	Rollin	Reddie	rreddiew@tinypic.com	09 Gina Park	65810	Missouri	4665-9188-1324-1040	$ 383.00
1033	2017-01-26	Dorry	Jenks	djenksx@virginia.edu	1 Butterfield Road	85210	Arizona	3578-9195-0297-7730	$ 636.00

Instance-specific optimization

Frequency Distribution

Cumulative Distribution Function (CDF)



Learned Index:
- Perf: 1x addition
- Storage: 16 bytes

B-Tree:
- Perf: ~1 I/O
- Storage: O(n)

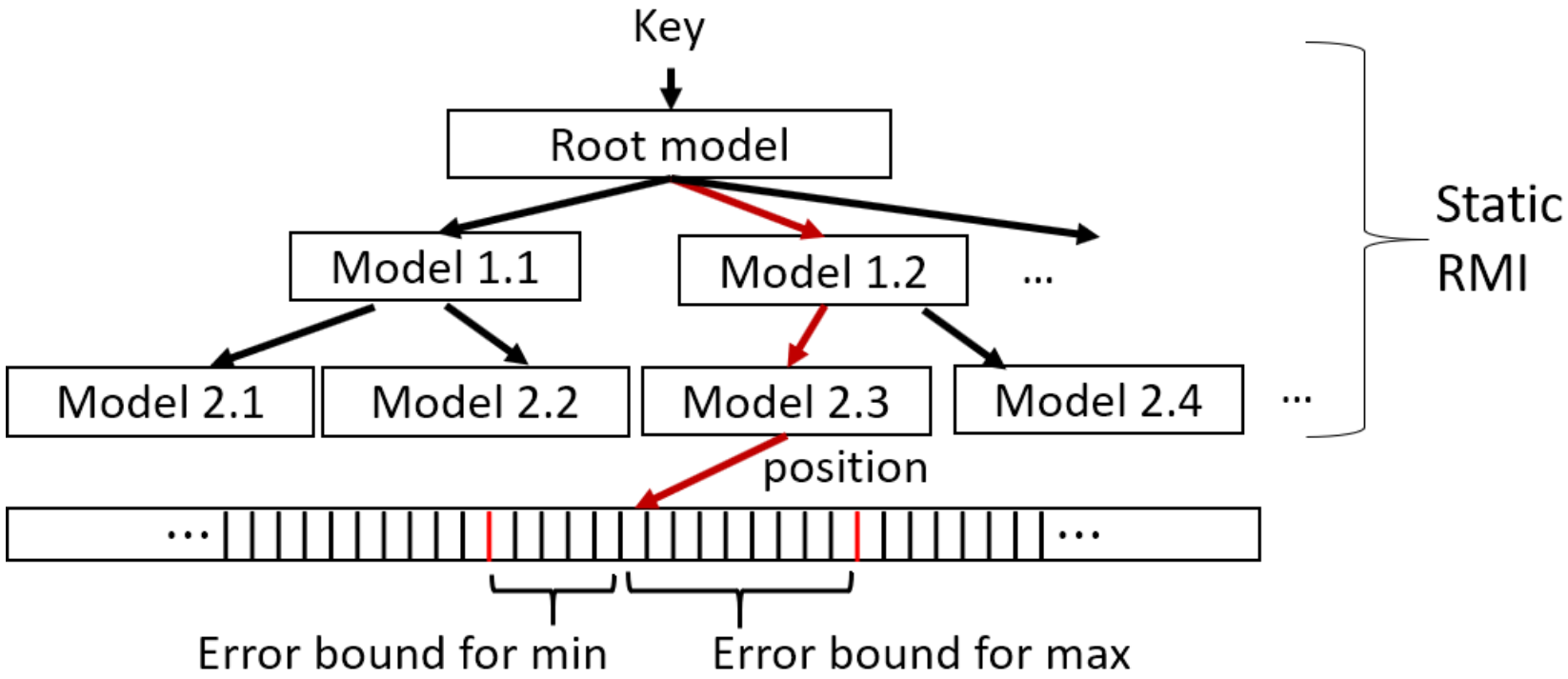
```
data_array[id - 1000]
```

id	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066

Learned Index*

~3x faster lookups,
~10x smaller size vs.
B+Tree

Works for static,
read-only workloads



*Kraska et al., "The Case for Learned Index Structures", in SIGMOD 2018.

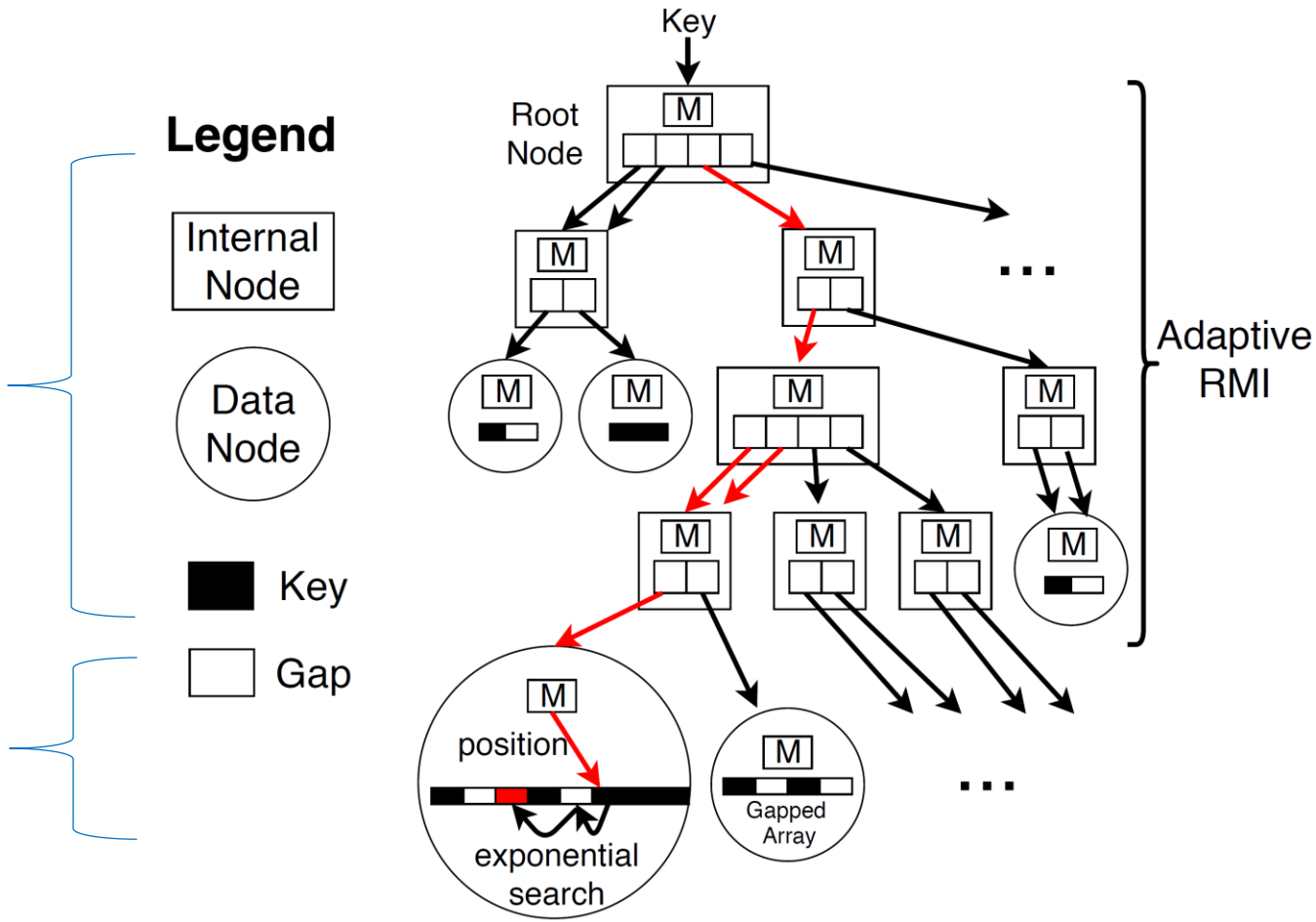
ALEX design goals

	B-Tree	Learned Index	ALEX
Lookup time	Slow	Fast	Faster
Insert time	Fast	Not Supported	Fast
Space usage	High	Low	Low

ALEX design overview

- Recursive Model Index (RMI)
  - Hierarchy of models
  - Higher level models pick model at the next level
  - Leaf level models predict the position

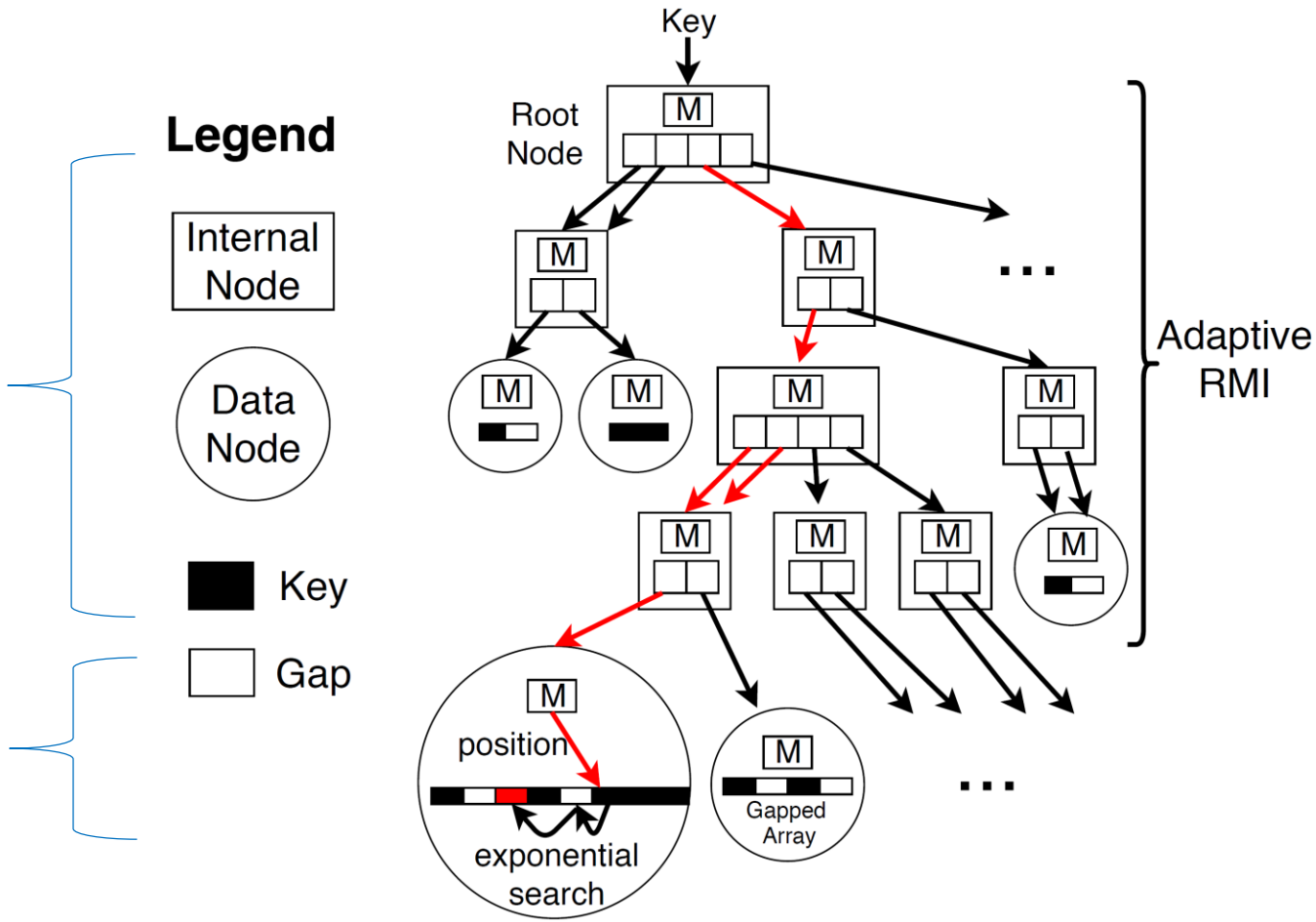
- At leaf nodes, leave gaps (free spaces)
  - Efficiently absorb inserts
  - Use model-based inserts which lead to faster lookups



Lookups in ALEX

Use the RMI to predict the location of the key in a leaf data node

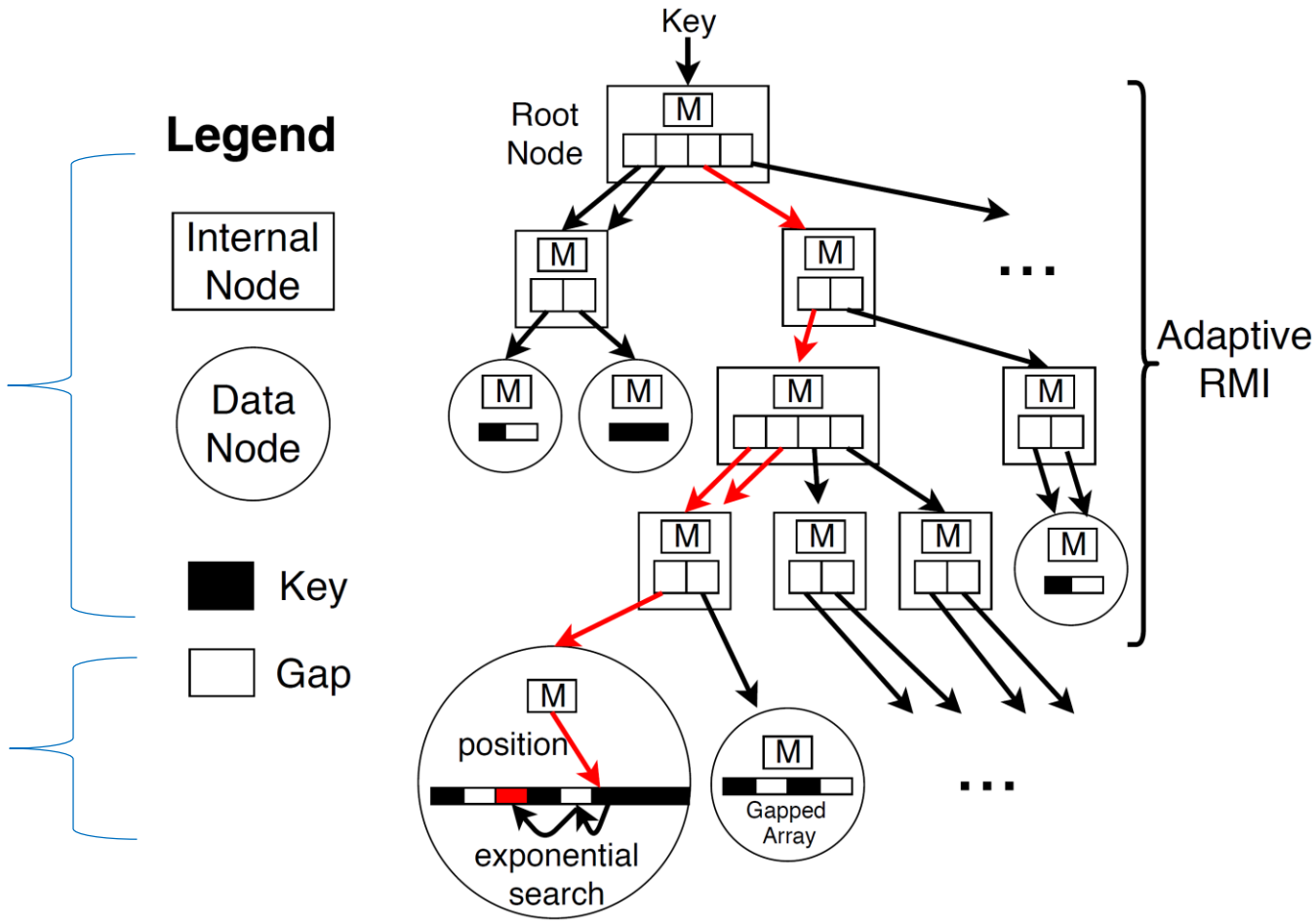
Use exponential search within the leaf node, starting at the predicted location



Inserts in ALEX

- Use the RMI to predict the location where the key ***should be*** inserted in the leaf data node

- If the predicted location is empty, insert the key
- Else shift the existing keys towards the closest gap



Mechanisms to dynamically grow ALEX

Node Expansions

- If a node becomes full, allocate a new larger node by an expansion factor 'c'
- Scale or retrain the model

Node Splits

- If a node becomes full and has reached a max node size, split it into two nodes
  - B-Tree like splits
  - Downward splits

Insertion Algorithm

- Combines these "mechanisms" with appropriate "policies"
  - All decisions are driven by simple "cost models" that are learned from the workload

Dataset characteristics

Table 1: Dataset Characteristics

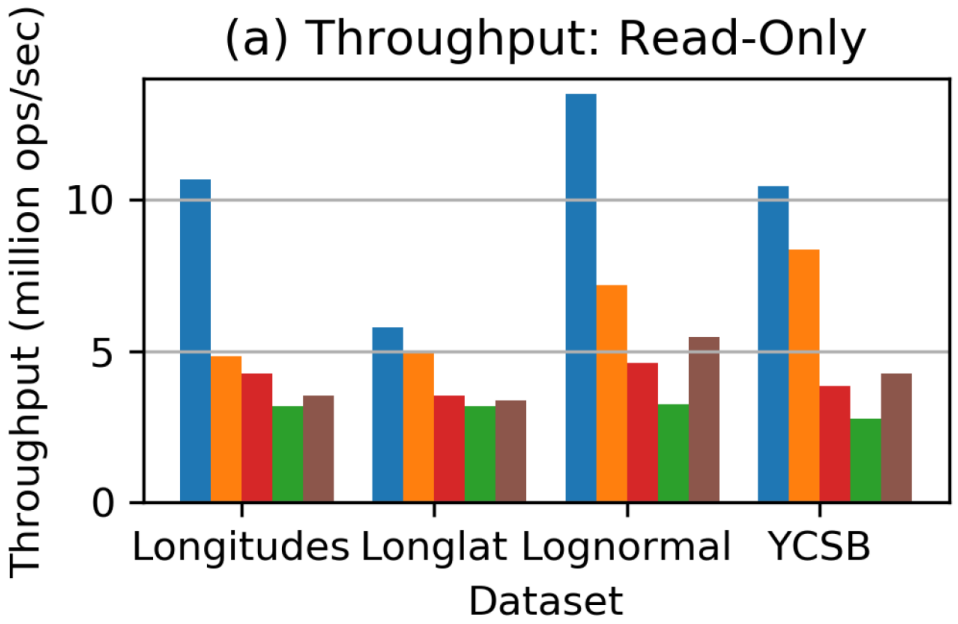
	longitudes	longlat	lognormal	YCSB
Num keys	1B	200M	190M	200M
Key type	double	double	64-bit int	64-bit int
Payload size	8B	8B	8B	80B
Total size	16GB	3.2GB	3.04GB	17.6GB

Performance summary

- Read-only workloads:
  - ALEX has 2.2x faster search speed vs. Learned Index, with 15x smaller index
  - ALEX has 4.1x faster search speed vs. B-Tree, with 3 orders of magnitude smaller index

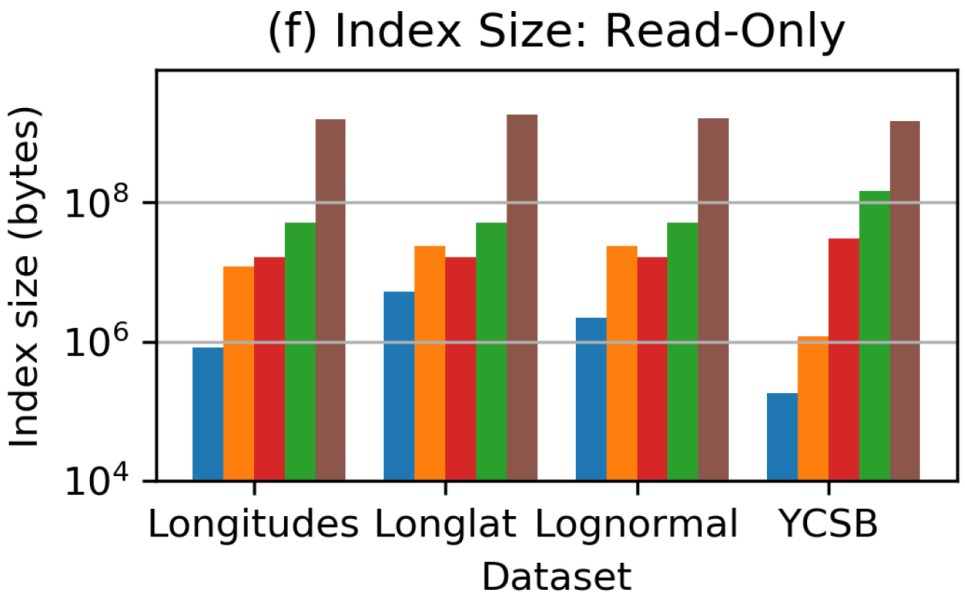
- Read-write workloads:
  - ALEX has 2-4x faster search speed vs. B-Tree

Read-only Workload



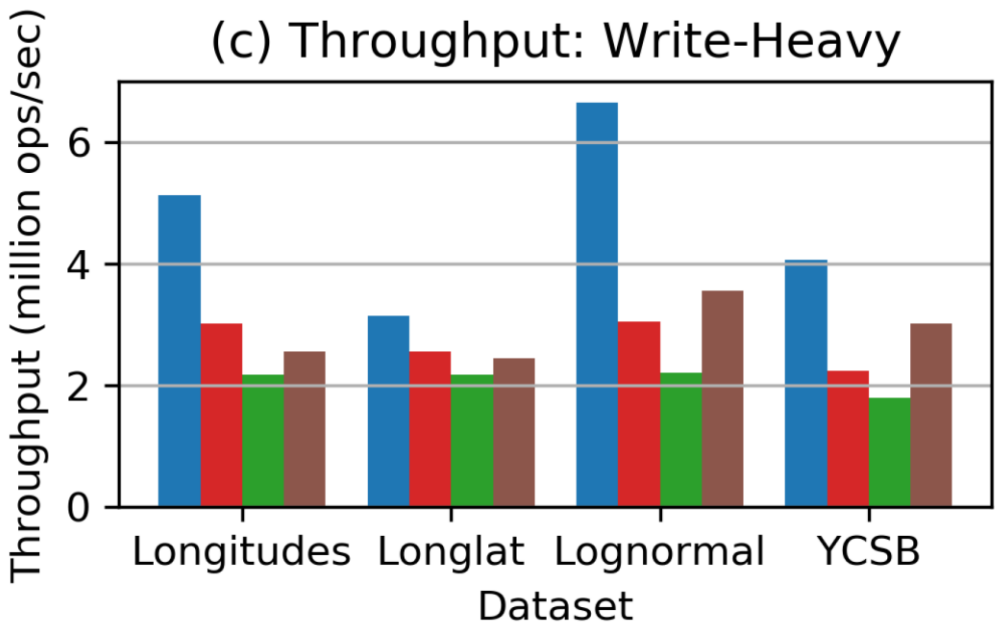
~4x faster than B+Tree

~2x faster than Learned Index

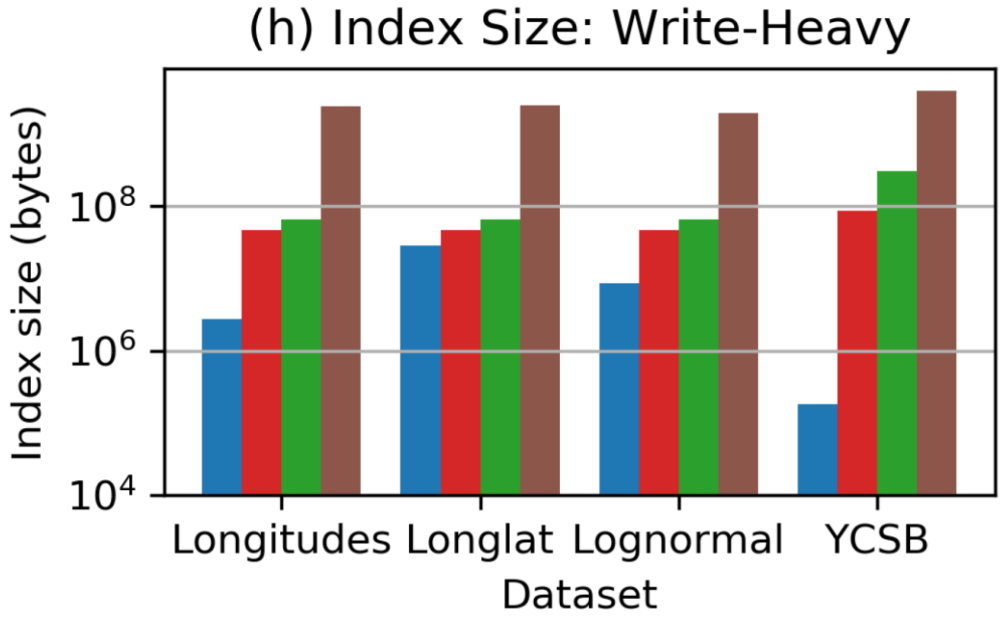


~3 orders of magnitude less space for index

Write-heavy Workload

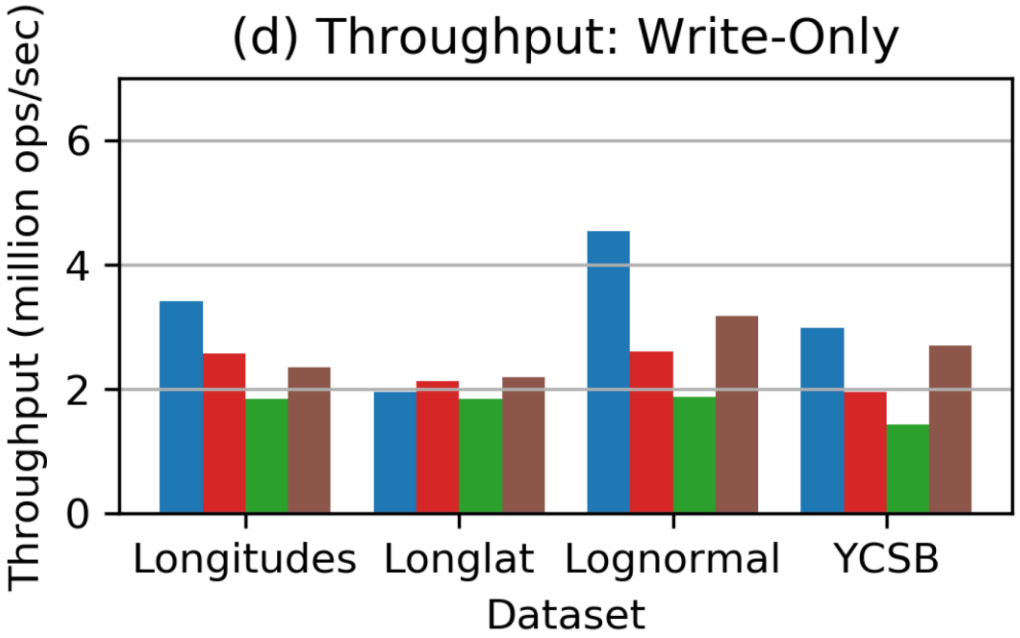


~2-3x faster than B+Tree

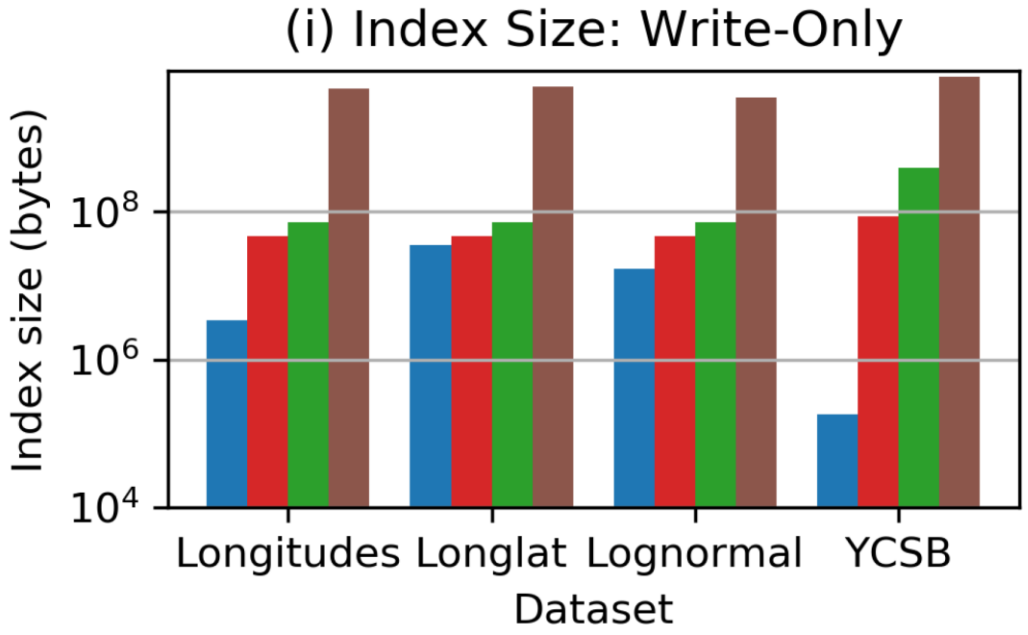


~3 orders of magnitude less space for index

Write-only Workload

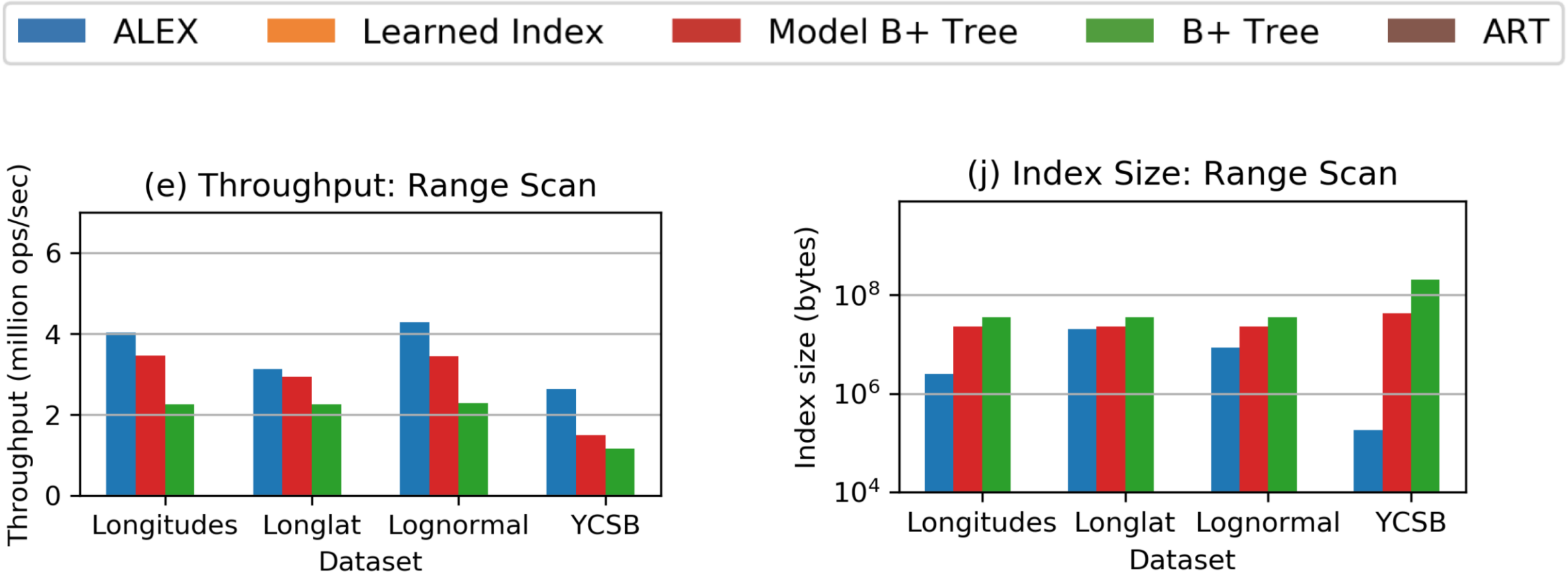


~2x faster than B+Tree



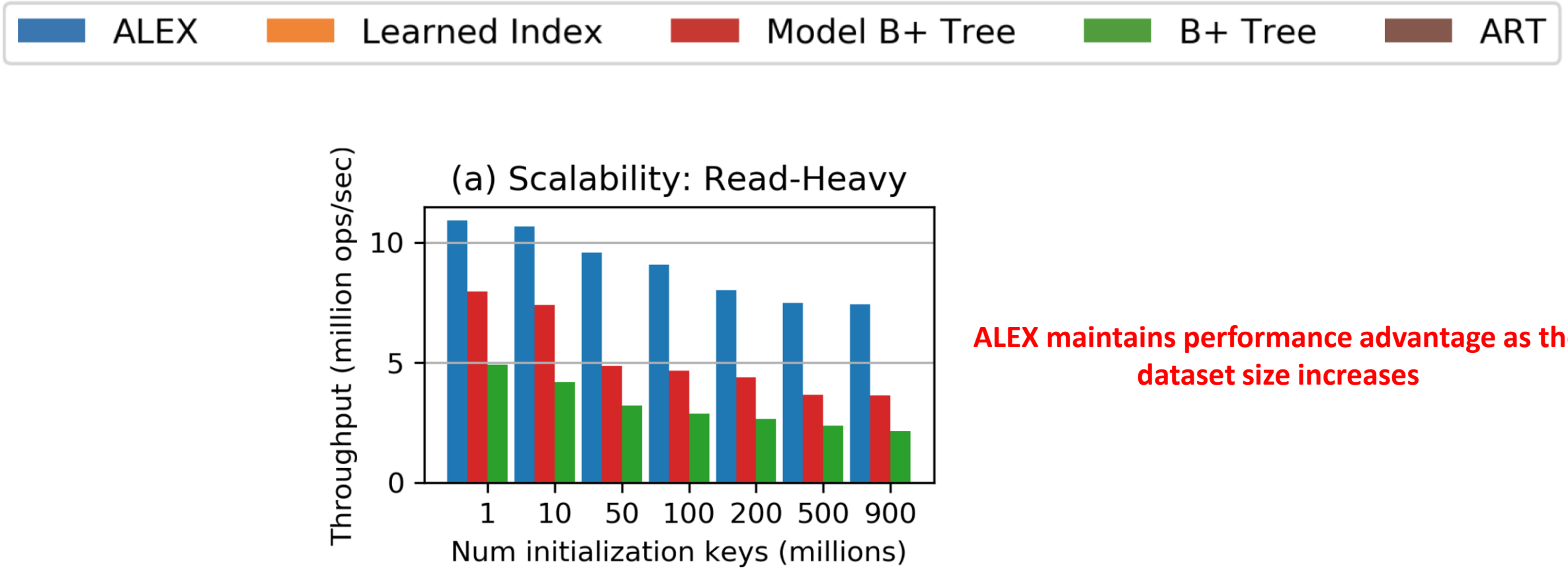
~3 orders of magnitude less space for index

Range-scan Workload



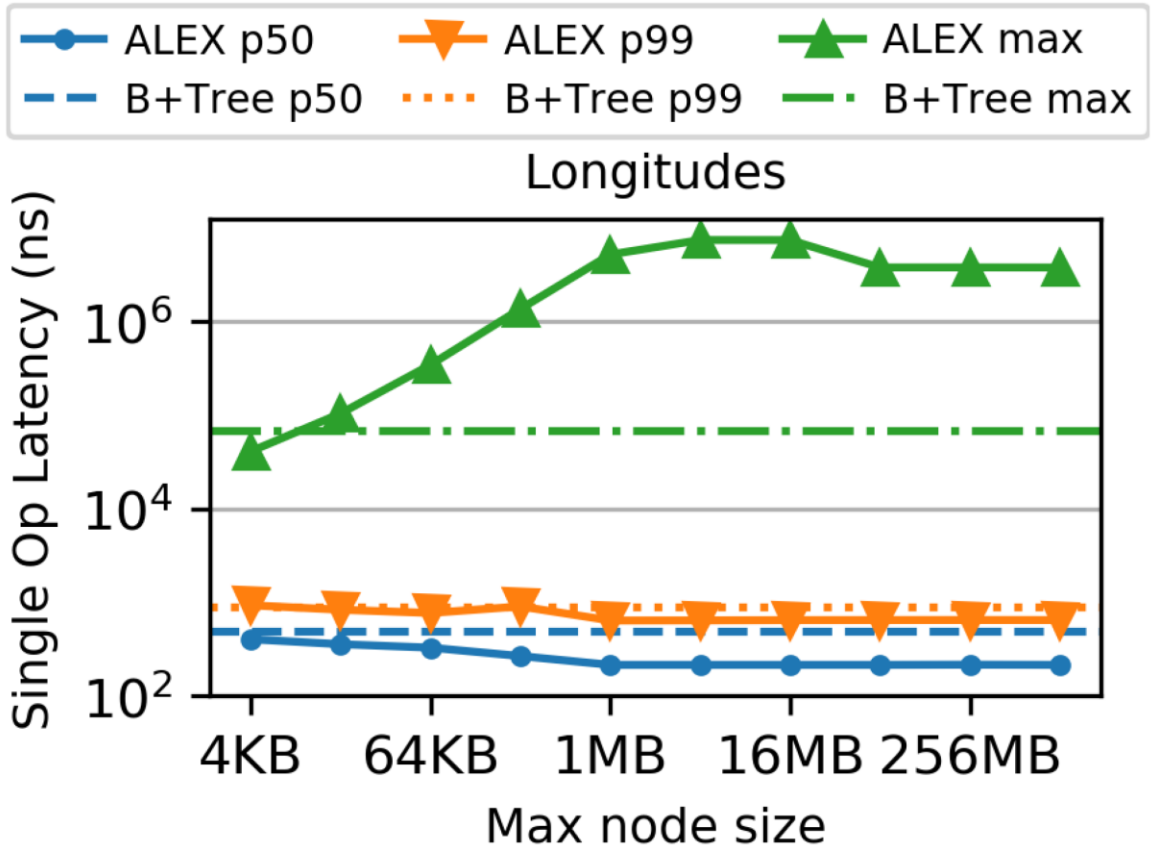
ALEX is still faster but relative performance decreases as scan cost dominates

Scalability



ALEX maintains performance advantage as the dataset size increases

Insert Latency



Increasing the max node size decreases median and P99 latency

… but tail latency is increased

Summary

- ALEX is an adaptive learned index for dynamic workloads
  - effectively combines the core insights from the Learned Index with proven storage & indexing techniques used in B+trees
- ALEX beats all baselines on all datasets
  - while consuming up to **three orders of magnitude** less storage space for indexing
- Next steps:
  - persistence on SSDs or Persistent Memory
  - concurrency control
  - open source

Related Research

- High performance, latch-free indexing for NVM
  - "BzTree: A high-performance latch-free range index for non-volatile memory", joint work with J. Arulraj, J. Levandoski, and P.A. Larson, in VLDB 2018

- Hiding memory stall latencies for in-memory indexes
  - "Exploiting coroutines to attack the killer nanoseconds", joint work with C. Jonathan, J. Hunter, J. Levandoski, and G. Nishanov, in VLDB 2018

Database Group @ MSR Redmond

Arvind Arasu
Researcher

Badrish Chandramouli
Senior Principal Researcher

Jonathan Goldstein
Senior Principal Researcher

Raghav Kaushik
Senior Researcher

Donald Kossmann
Distinguished Scientist and Director of Redmond Lab

Yinan Li
Researcher

David Lomet
Principal Researcher and Research Manager

Umar Farooq Minhas
Researcher

Ravi Ramamurthy
Researcher

New in 2019!

Jae Young Do
&
Jose Faleiro

Database Group @ MSR Redmond

- Always Encrypted – data processing on encrypted data
- AMBROSIA – a programming language independent approach for authoring and deploying highly robust distributed applications (https://github.com/microsoft/ambrosia)
- Cipherbase – a comprehensive database system that provides strong end-to-end data confidentiality through encryption
- CRA – a software layer (library) that makes it easy to create and deploy distributed dataflow-style applications (https://github.com/microsoft/cra)
- Deuteronomy – a database architecture that provides a clean separation of transaction functionality from data management functionality
- FASTER – a new key-value store for point operations, that combines a highly cache-optimized concurrent hash index with a novel self-tuning data organization (https://github.com/microsoft/faster)
- FISHSTORE – a new ingestion and storage layer for flexible- and fixed-schema datasets (https://github.com/microsoft/fishstore)
- MISON – fast parser for data analytics
- Socrates – The New SQL Server in the Cloud (https://aka.ms/sqlsocrates)
- TRILL – fast stream processing library: a trillion tuples a day (https://github.com/microsoft/trill)

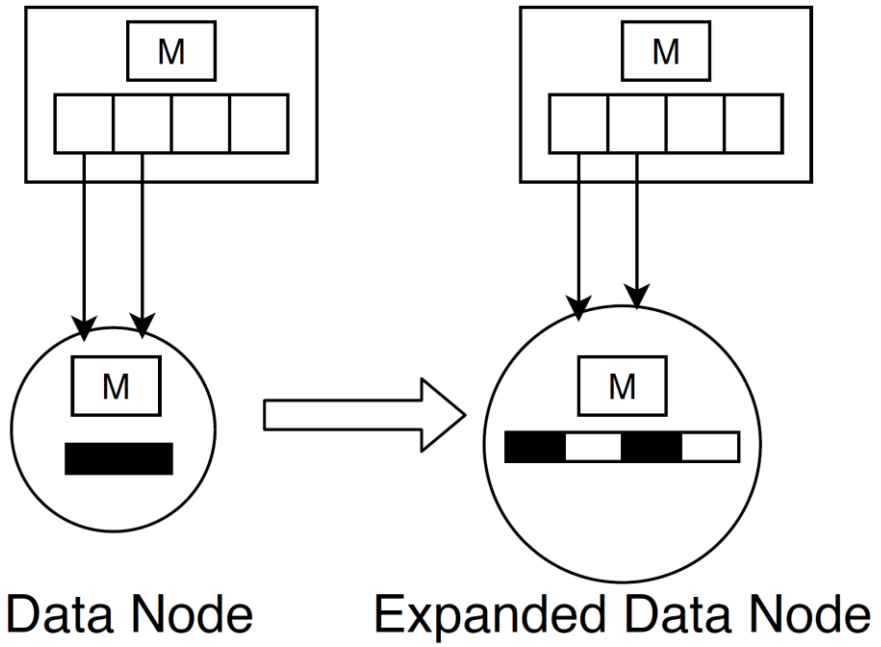
Database Group @ MSR Redmond

Now accepting, FY 2020 applications

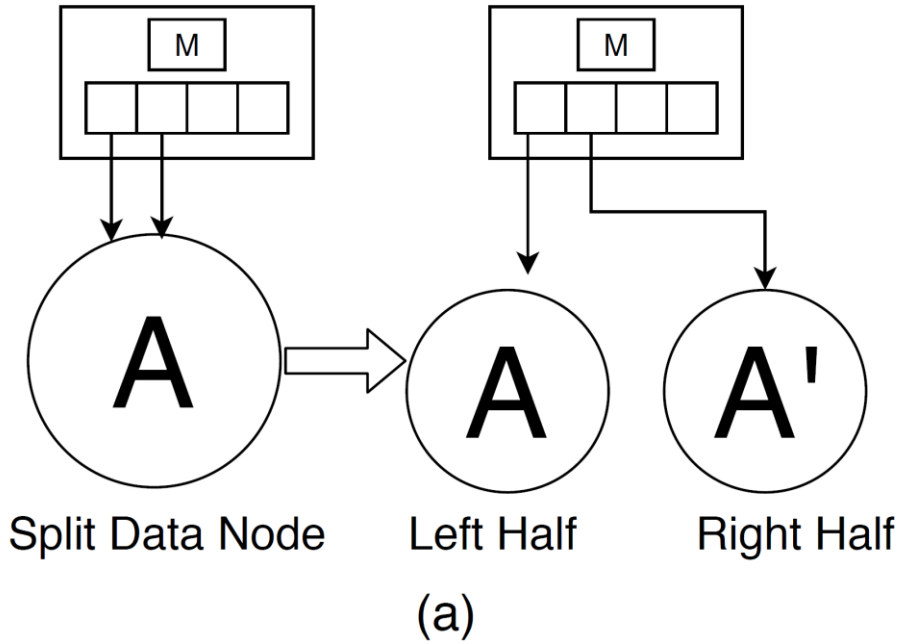
Internships:
https://careers.microsoft.com/us/en/job/722104/Research-Intern-Database-Group

Full-time:
Please contact ufminhas@microsoft.com for details. ☺

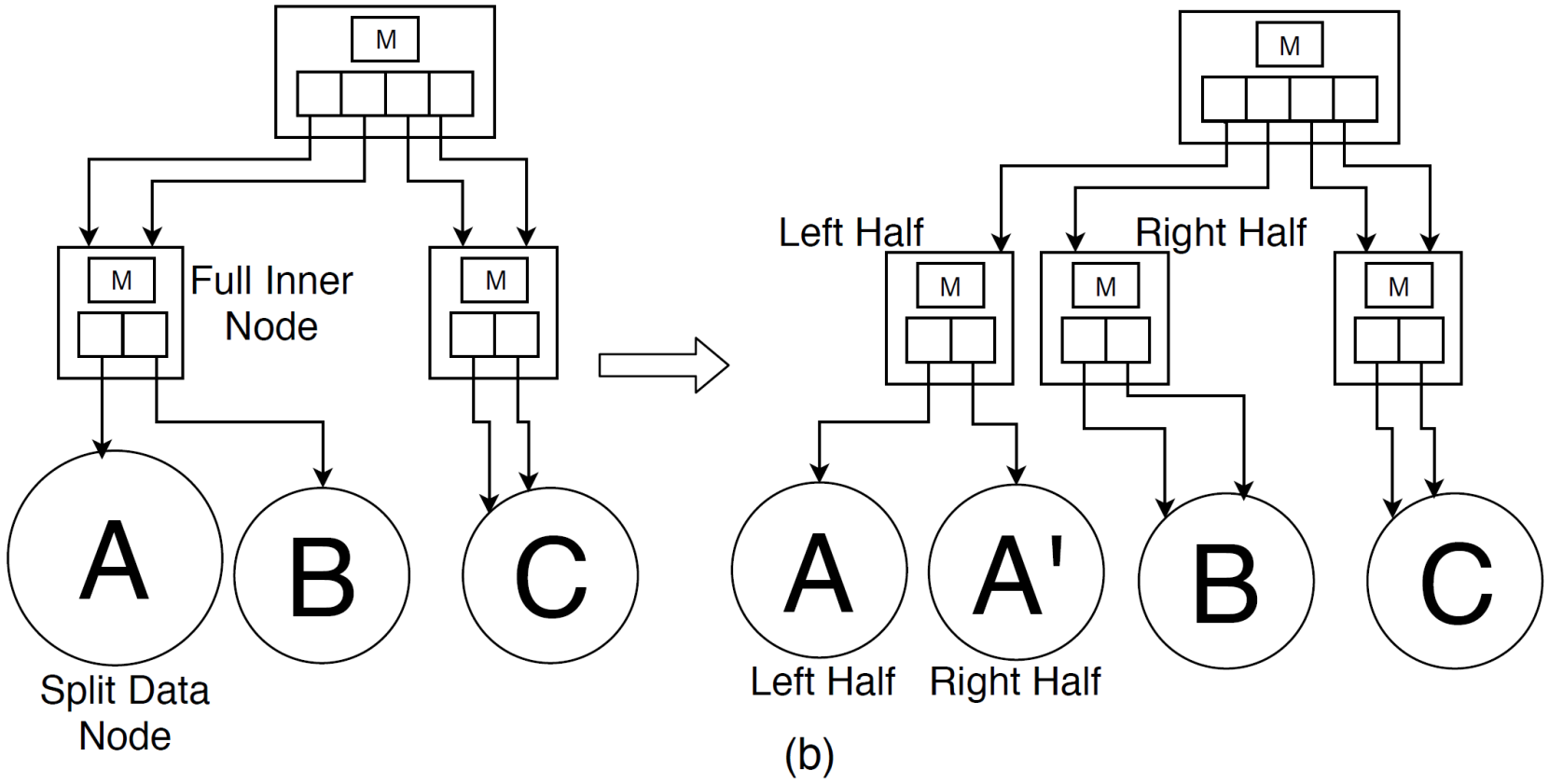
Node Expansion



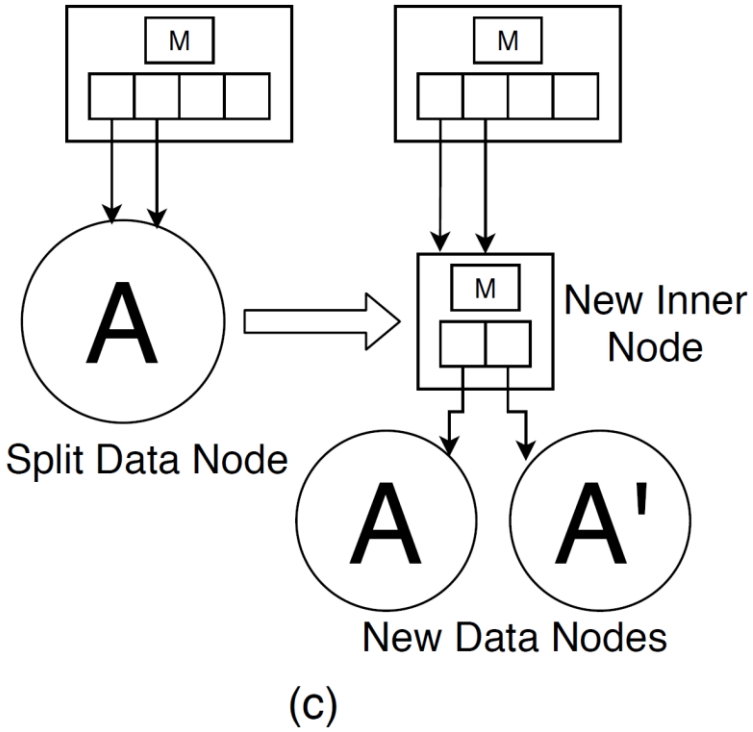
B-Tree Like Split – Case 1



B-Tree Like Split – Case 2

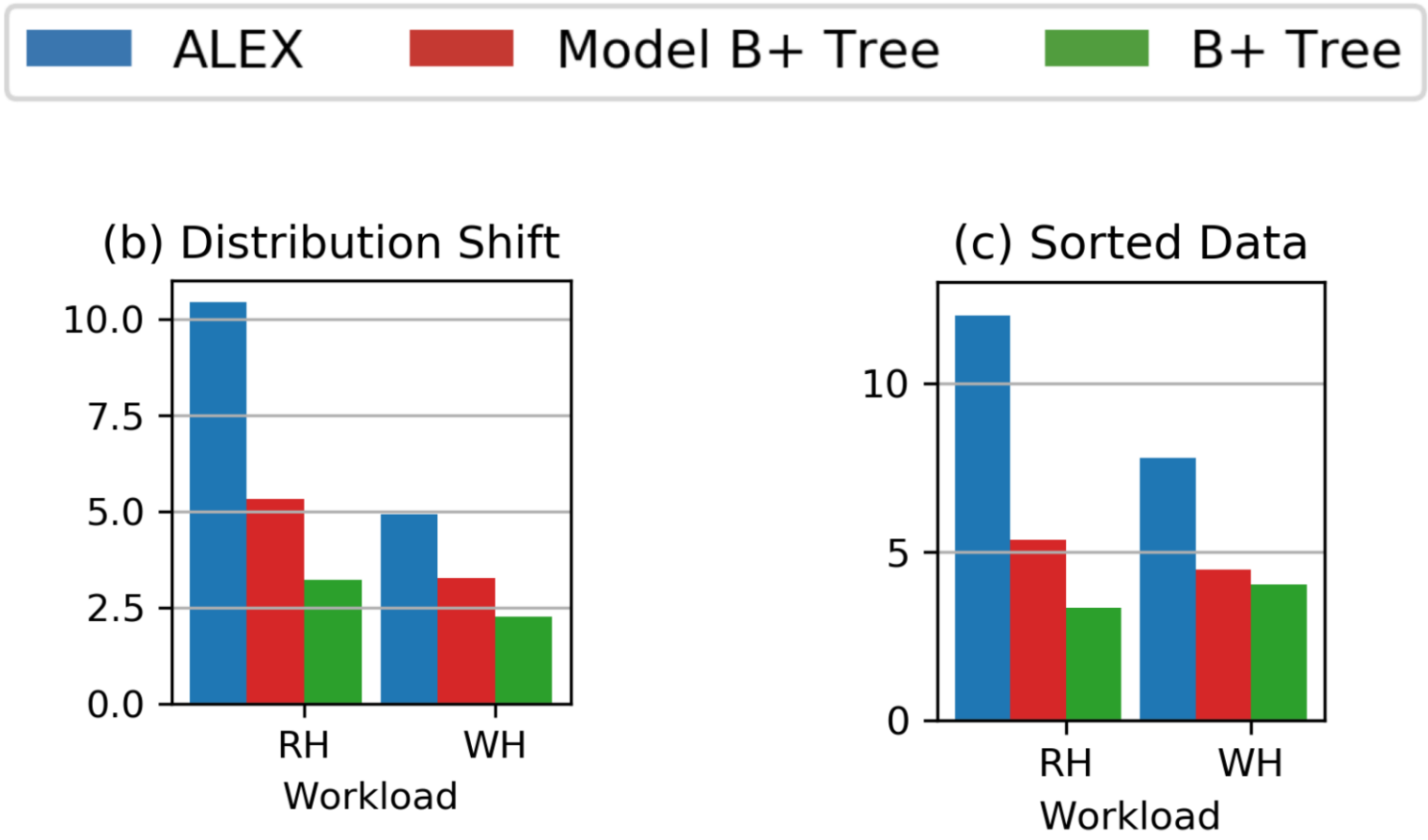


Split Down

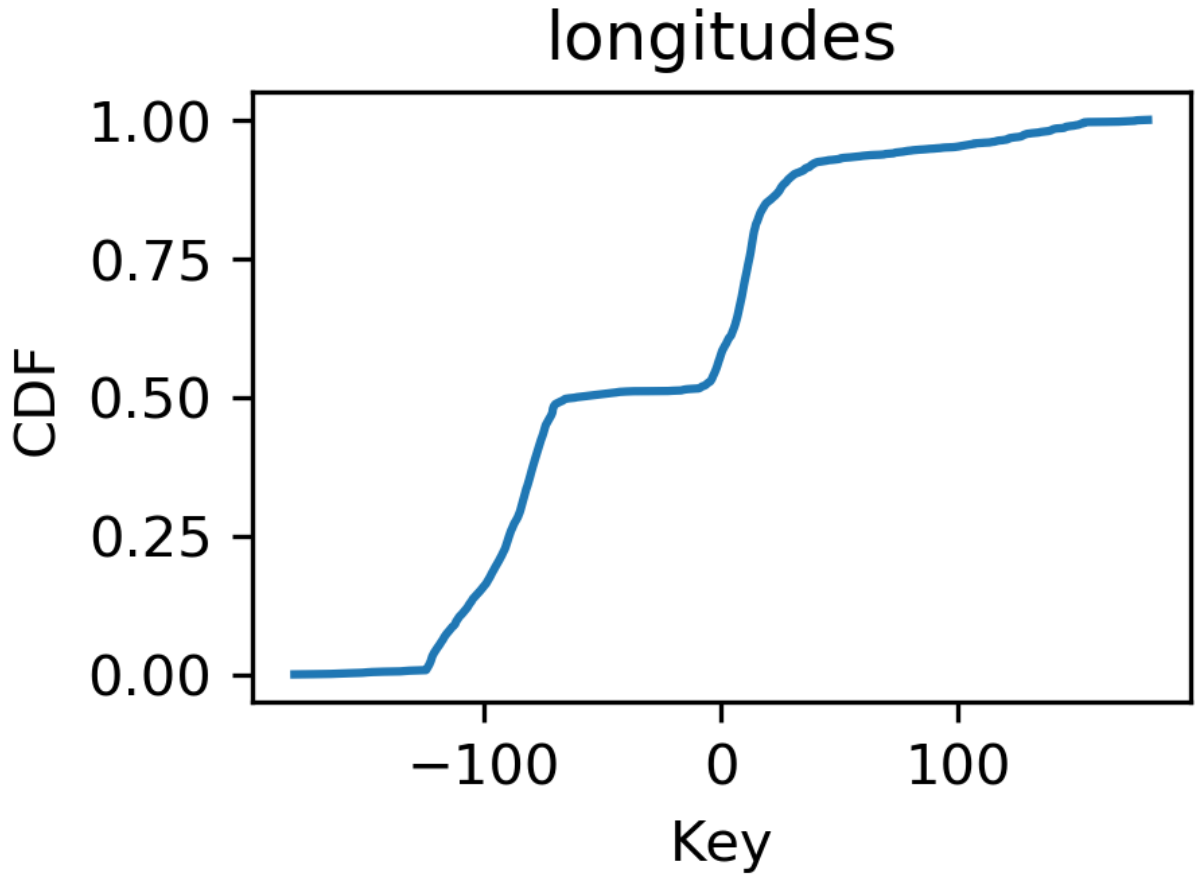


(c)

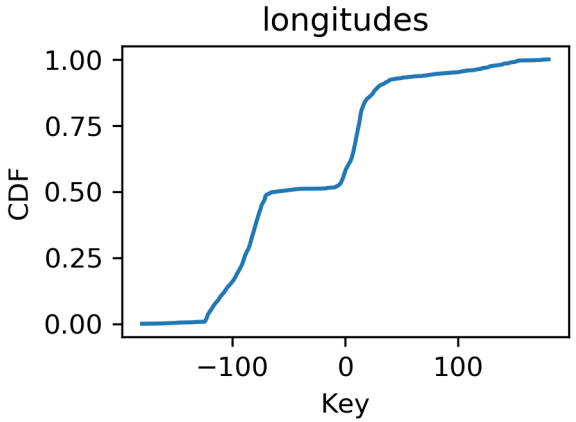
Distribution Shift

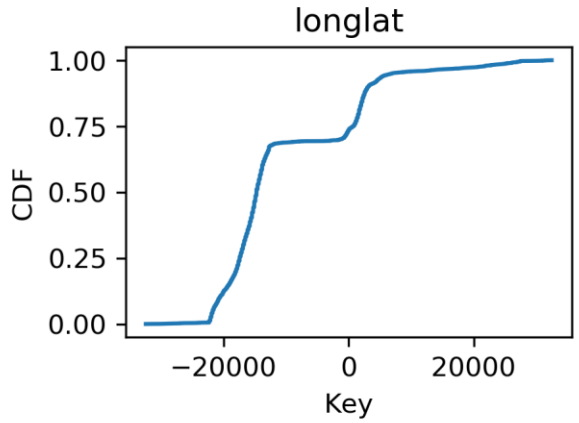


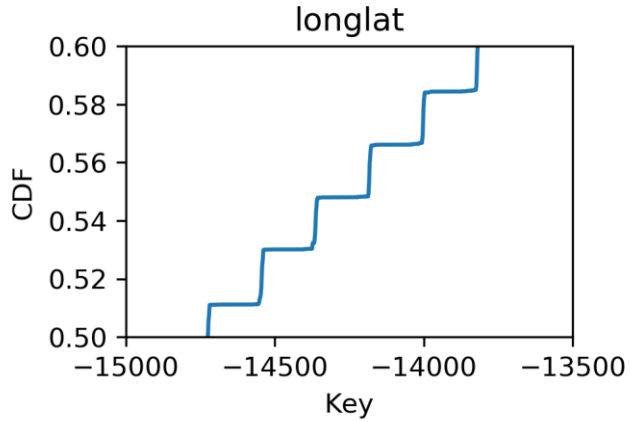
Longitudes dataset CDF

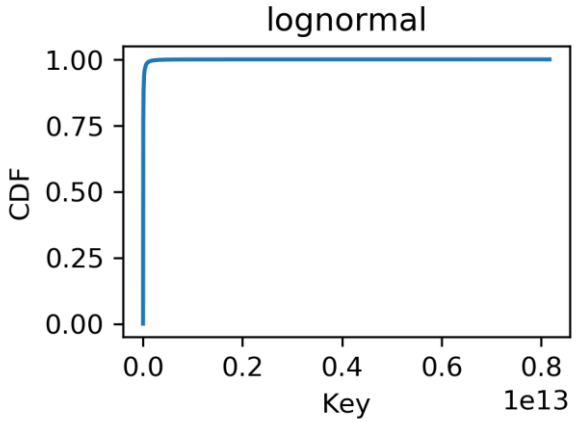


Dataset CDFs

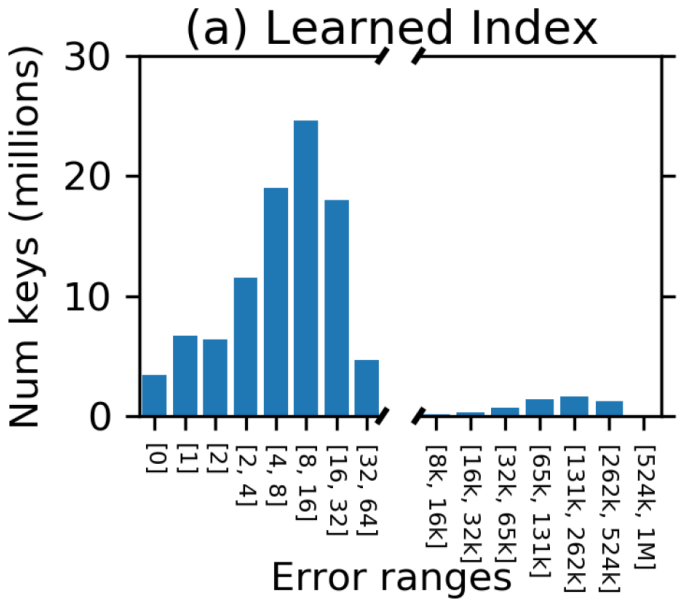


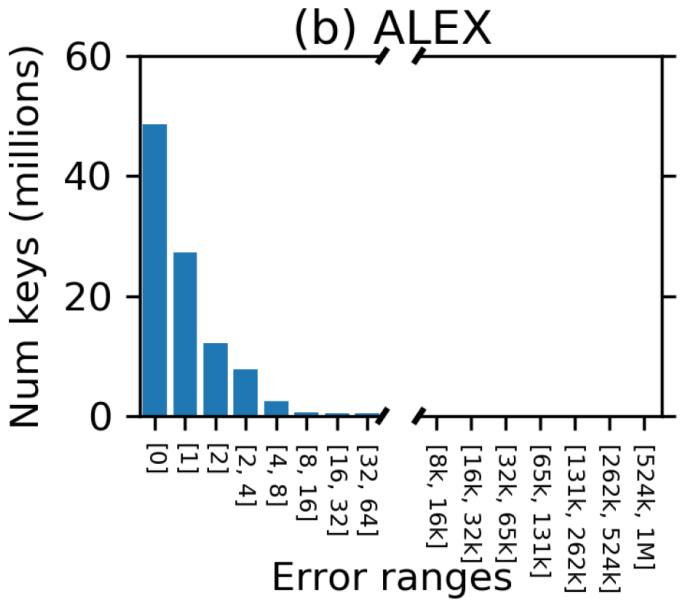


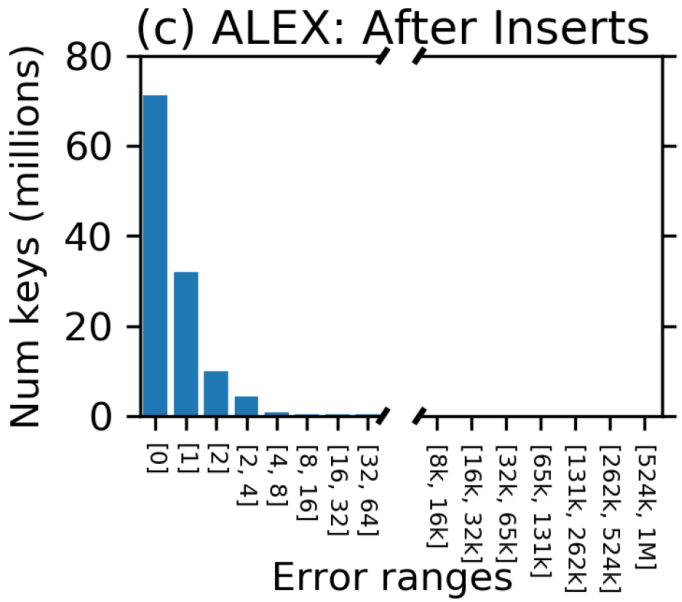




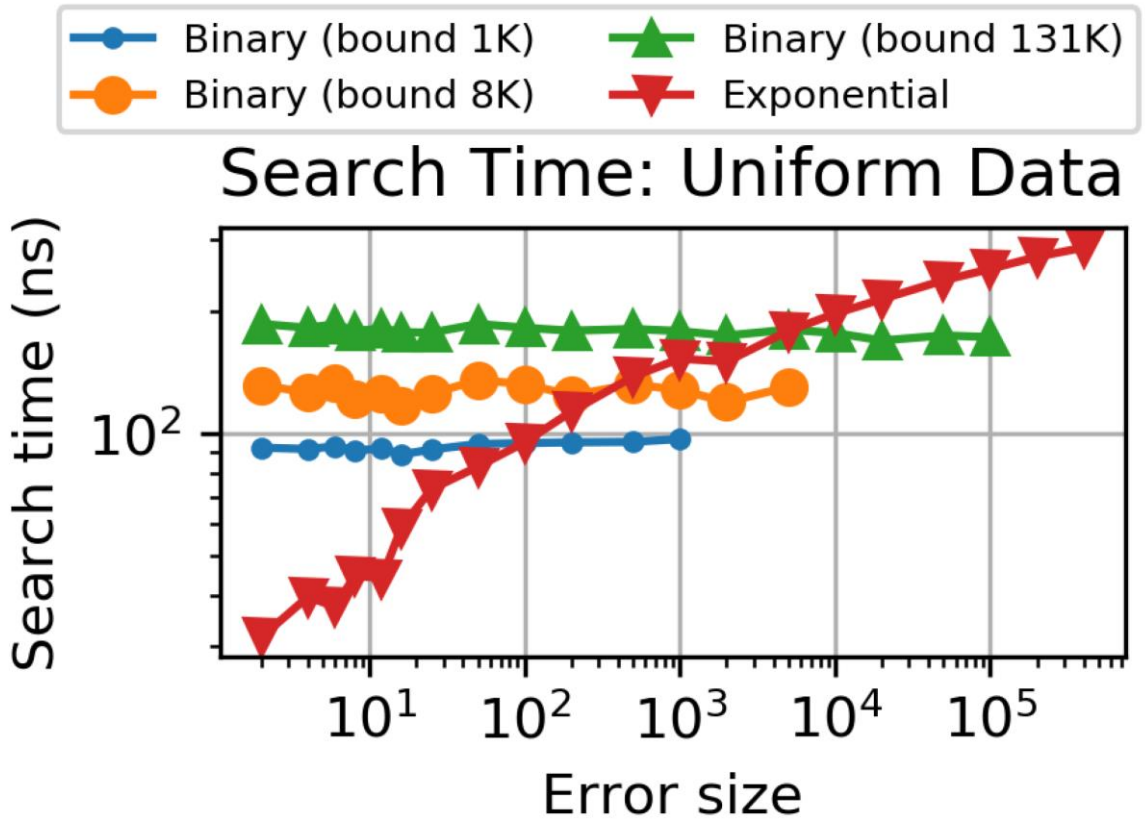
Errors distribution







Exponential vs. Binary Search



Why a learned index?

- Instance-specific optimizations
  - Use statistical or machine learning models to learn the specific distribution of the data at initialization
  - Adapt the structure of the index dynamically with more data

- More efficient execution on modern CPUs
  - Traverse-to-leaf in a learned index is "computation" based (vs. comparison based) – hence avoids branch mispredictions
  - The index structure is much more compact – hence CPU caches can be more effective