Database Forensic Analysis Through Internal Structure Carving

By

James Wagner, Alexander Rasin and Jonathan Grier

Presented At

The Digital Forensic Research Conference

DFRWS 2015 USA Philadelphia, PA (Aug 9th - 13th)

Database Forensic Analysis through Internal Structure Carving

James Wagner
Dr. Alexander Rasin
Jonathan Grier

Roadmap

- Motivation
  - File carving (for databases)
  - Stochastic analysis
- Results
- Future work

Motivation

- A lot of data lives in databases
- Recovery
  - Rely on “safe” storage (or backups)
  - Inconsistent / commercial
- Monitoring
  - Logs and profiling
  - Inspect DB connections
    - e.g., IBM Guardium

Database Storage

Table Rows

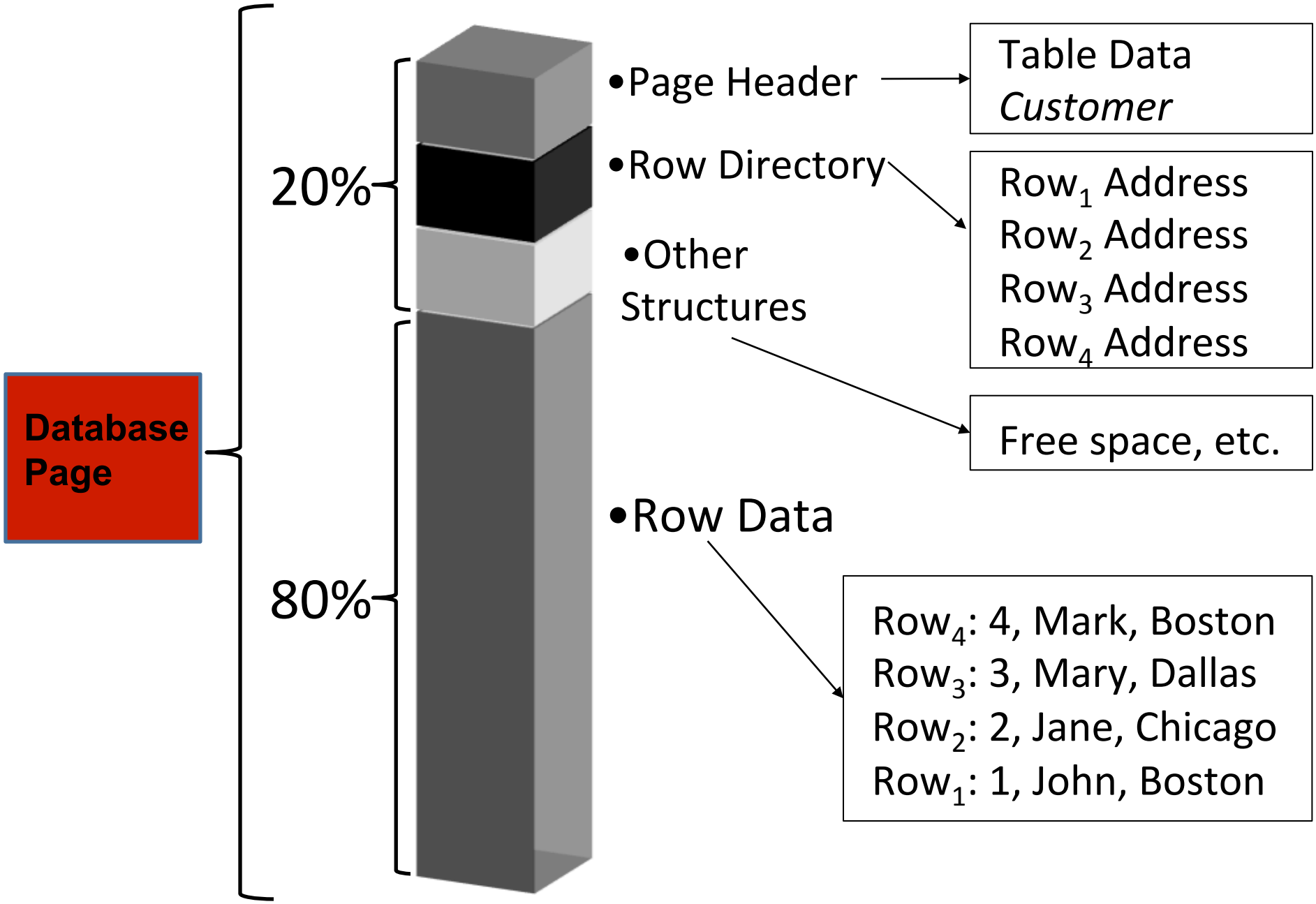
RID	Name	Position	FavoriteTown
1	Stan	Professor	Westwood
2	Ugur	Professor	Providence
3	Tom	Professor	Providence
4	Alex	Student	Providence
5	Stan	Professor	Cambridge
6	Andy	Student	Providence

Logical

Physical

Disk Pages

#1	1	Stan	Professor	Westwood
	2	Ugur	Professor	Providence
#2	3	Tom	Professor	Providence
	4	Alex	Student	Providence
#3	5	Stan	Professor	Cambridge
	6	Andy	Student	Providence


Database Page
20%
80%
•Page Header
•Row Directory
•Other Structures
•Row Data
Table Data
Customer
Row_1 Address
Row_2 Address
Row_3 Address
Row_4 Address
Free space, etc.
Row_4: 4, Mark, Boston
Row_3: 3, Mary, Dallas
Row_2: 2, Jane, Chicago
Row_1: 1, John, Boston

Indexes and other Structures

Index on Favorite Town

Cambridge	5
Providence	2
Providence	3
Providence	4
Providence	6
Westwood	1

RID	Name	Position	FavoriteTown
1	Stan	Professor	Westwood
2	Ugur	Professor	Providence
3	Tom	Professor	Providence
4	Alex	Student	Providence
5	Stan	Professor	Cambridge
6	Andy	Student	Providence

MV

Logical

Physical

#1	Cambridge	5	Providence
	2	Providence	3

#2	Providence	4	Providence
	6	Westwood	1

#1	1	Stan	Professor	Westwood
	2	Ugur	Professor	Providence

#2	3	Tom	Professor	Providence
	4	Alex	Student	Providence

#3	5	Stan	Professor	Cambridge
	6	Andy	Student	Providence

#1

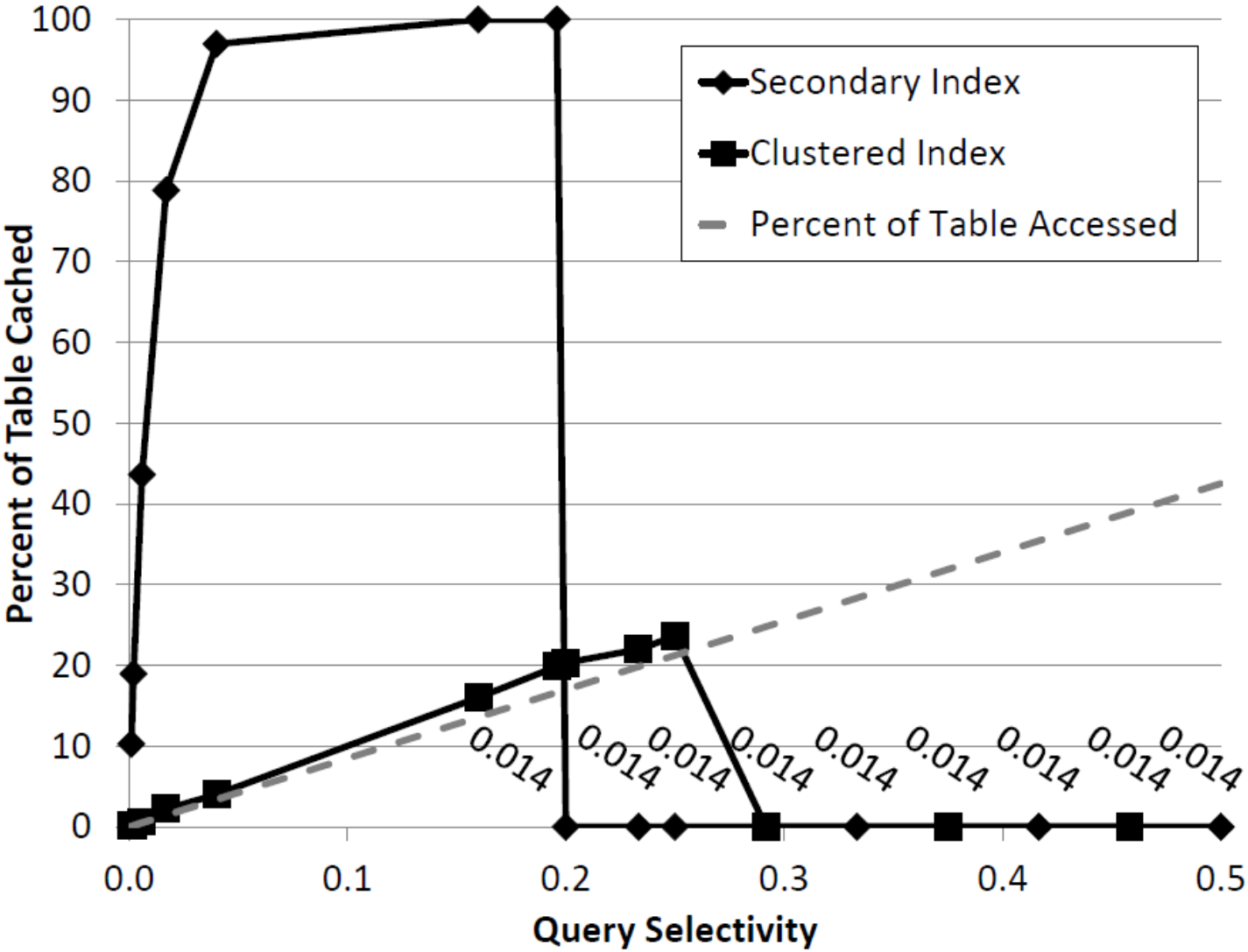
#2

#3

Roadmap

- Motivation
  - File carving (for databases)
  - Stochastic analysis
- Results
- Future work

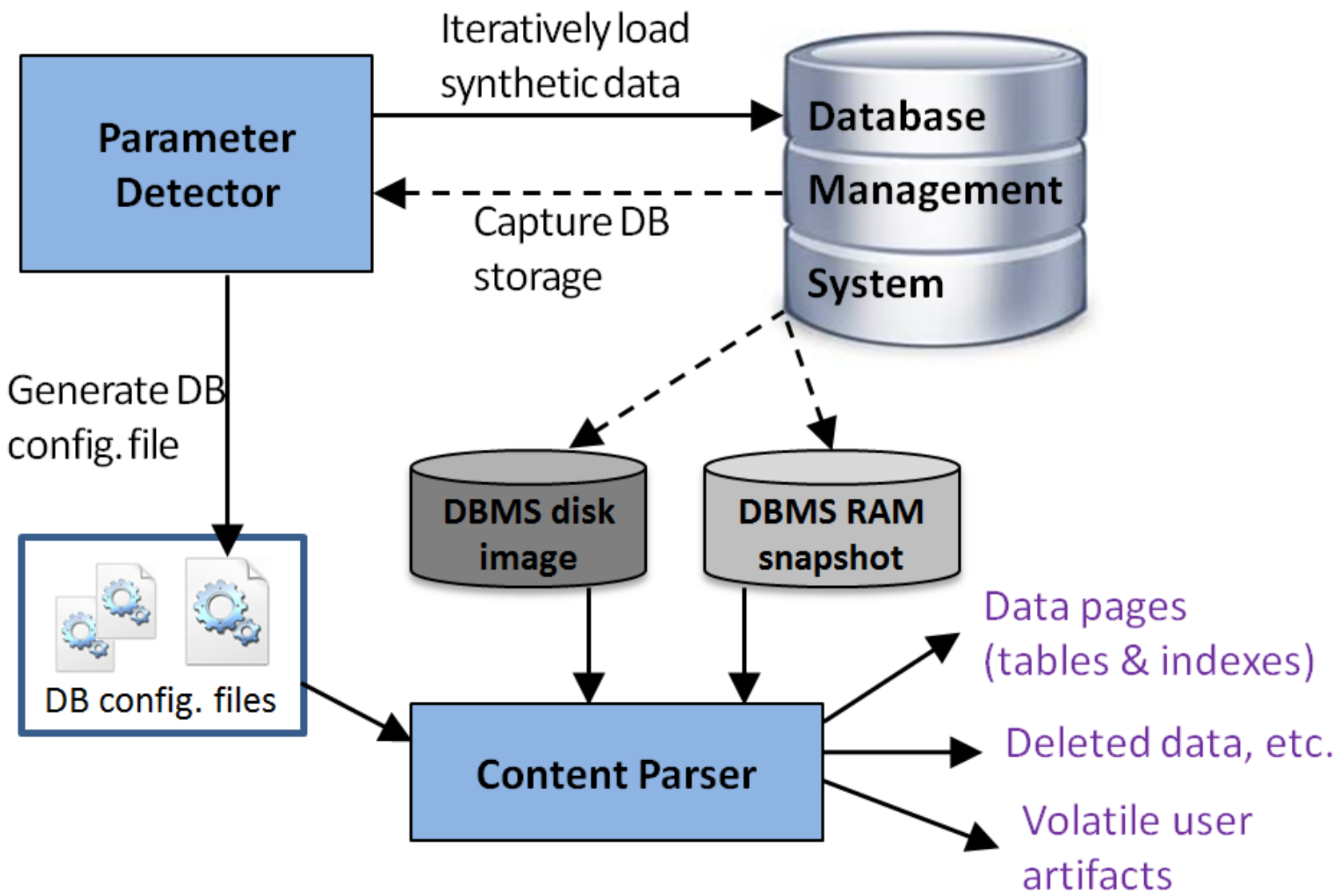
Database Caching Policies



Roadmap

- Motivation
  - File carving (for databases)
  - Stochastic analysis
- Results
- Future work

Architecture



<table>
<tr><th></th><th>Oracle</th><th>PostgreSQL</th><th>SQLite</th><th>Firebird</th><th>DB2</th><th>SQLServer</th><th>MySQL</th><th>Apache Derby</th></tr>
<tr><td>Structure Identifier</td><td>Yes</td><td>No</td><td colspan="5">Yes</td><td>No</td></tr>
<tr><td>Unique Page ID</td><td colspan="7">Yes</td><td>No</td></tr>
<tr><td>Row Dir. Sequence</td><td colspan="4">Top-to-bottom insertion</td><td colspan="4">Bottom-to-top insertion</td></tr>
<tr><td>Row Identifier</td><td>No</td><td colspan="2">Yes</td><td colspan="3">No</td><td colspan="2">Yes</td></tr>
<tr><td>Column Count</td><td colspan="3">Yes</td><td colspan="2">No</td><td>Yes</td><td>No</td><td>Yes</td></tr>
<tr><td>Column Sizes</td><td colspan="4">Yes</td><td colspan="2">No</td><td colspan="2">Yes</td></tr>
<tr><td>Column Directory</td><td colspan="4">No</td><td colspan="2">Yes</td><td colspan="2">No</td></tr>
<tr><td>Numbers w/Strings</td><td colspan="4">Yes</td><td colspan="2">No</td><td colspan="2">Yes</td></tr>
</table>

DBMS Versions

Hard to get some older DB versions

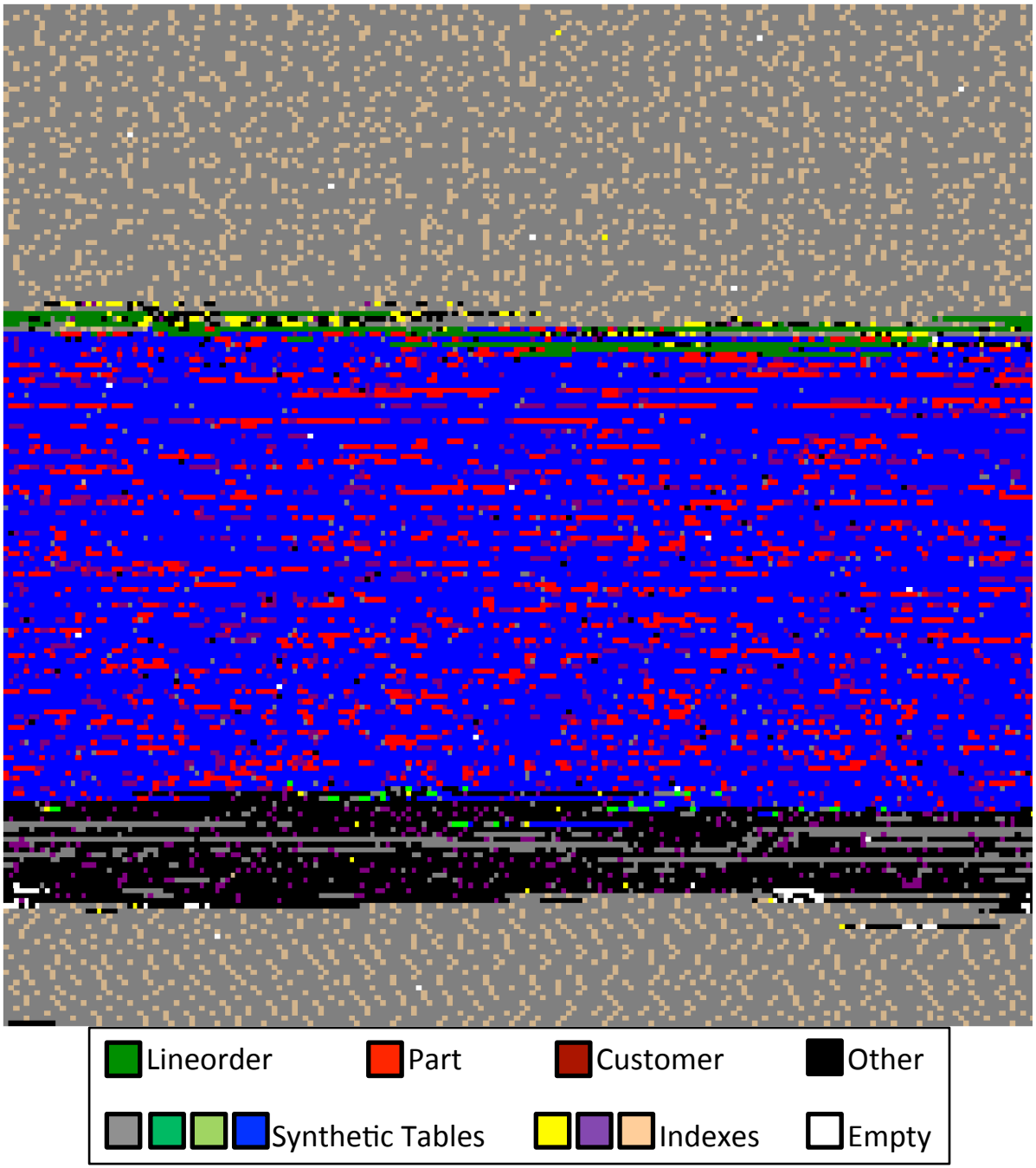
Different parameters

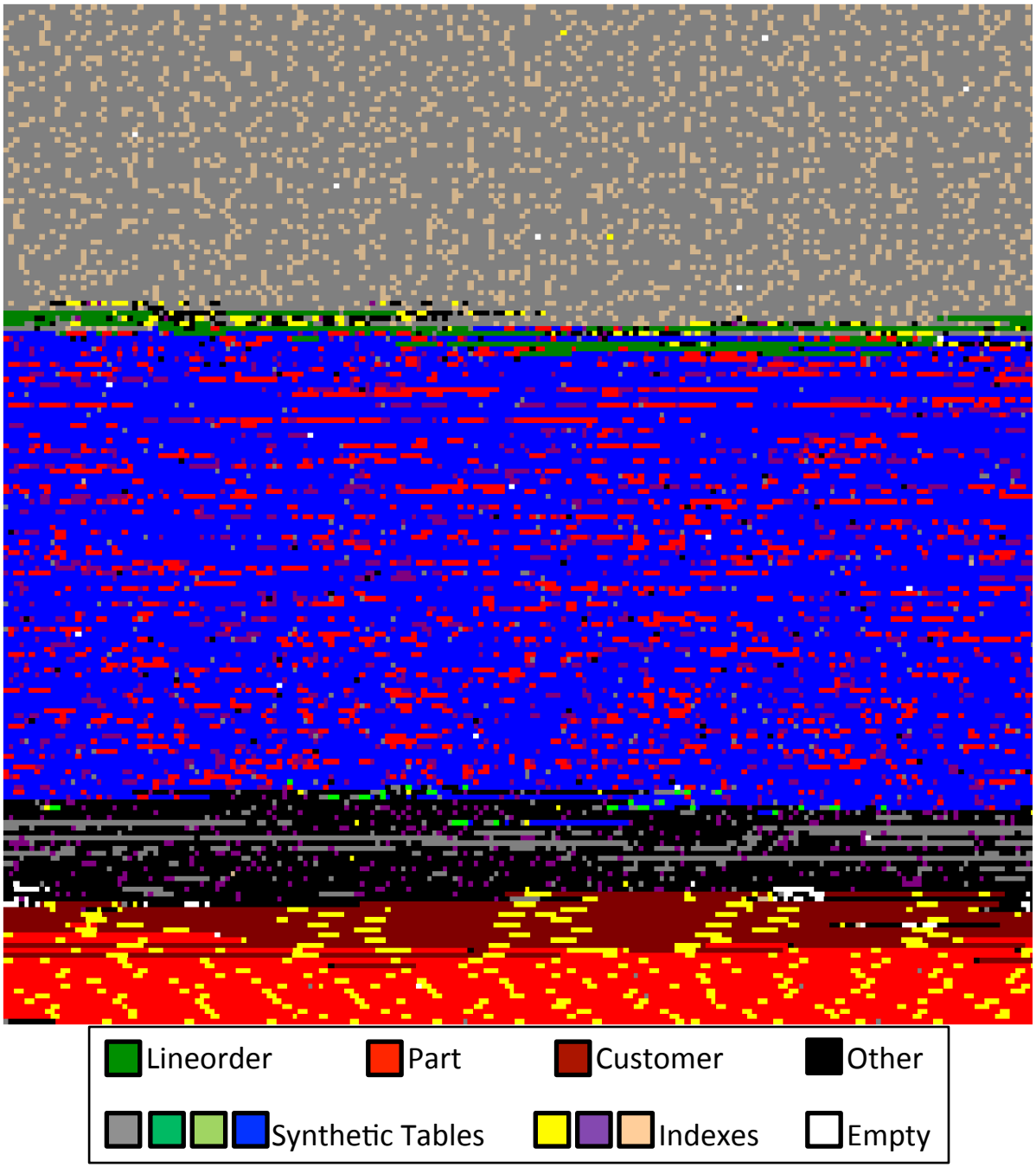
DBMS Version	Testing OS	Buffer Size(MB)	Page Size(KB)
Apache Derby 10.10	Linux	400	4
Apache Derby 10.5	Linux	400	4
DB2 Express-C 10.5	Linux	400	4
Firebird 2.5.1	Linux	400	8
Firebird 2.1.7	Windows	400	8
MySQL Server 5.1.73	Linux	800	16
MySQL Server 5.6.1	Windows	800	16
Oracle 11g R2	Windows	800	8
Oracle 12c R1	Windows	1200	8
PostgreSQL 7.3	Linux	400	8
PostgreSQL 8.4	Linux	400	8
PostgreSQL 9.3	Windows	800	8
SQLite 3.8.6	Linux	2	1
SQLite 3.8.7	Windows	2	1
SQLServer 2008 Enterprise	Windows (~~Linux~~)	800	8

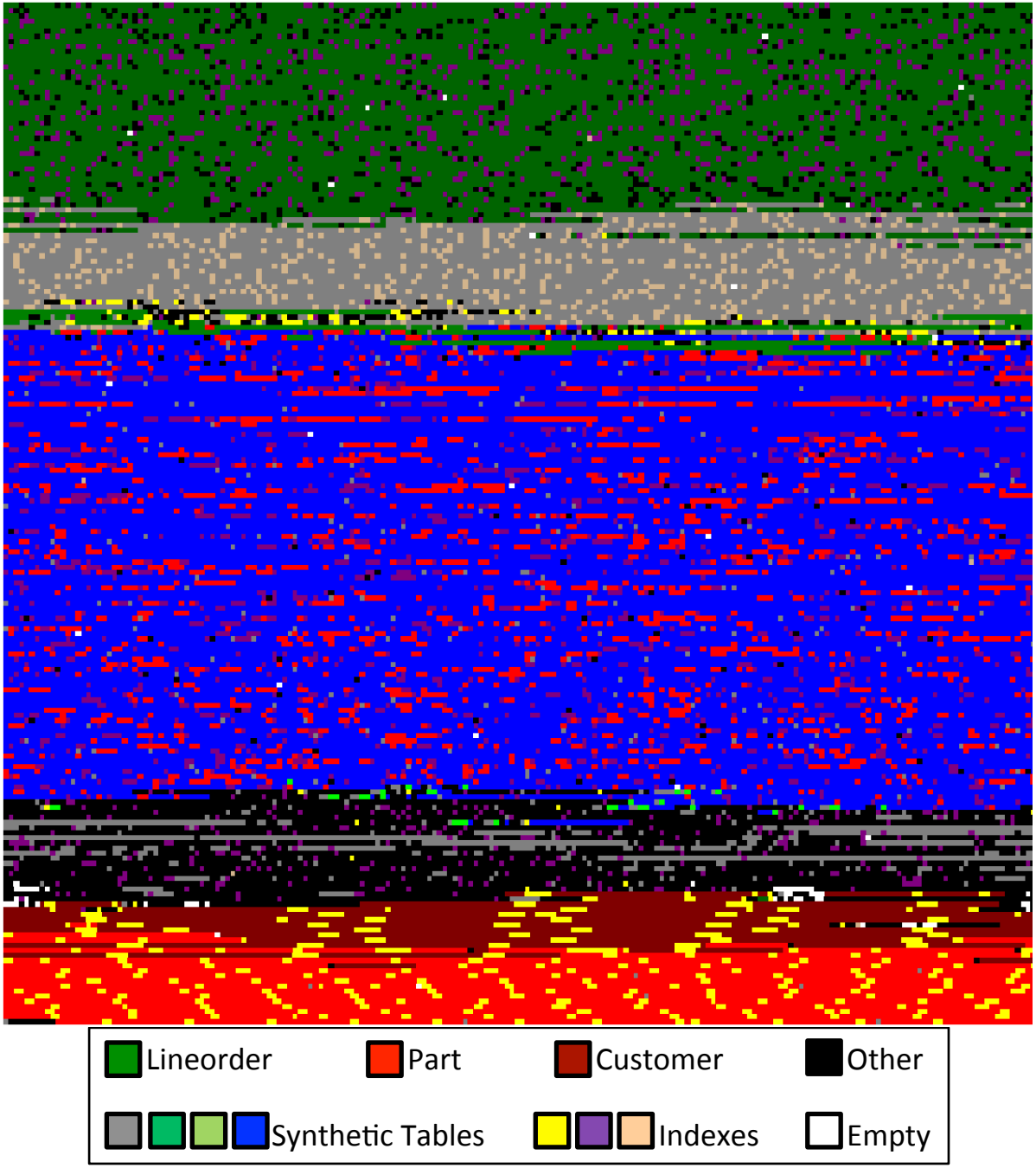
Recovering Corrupted Data

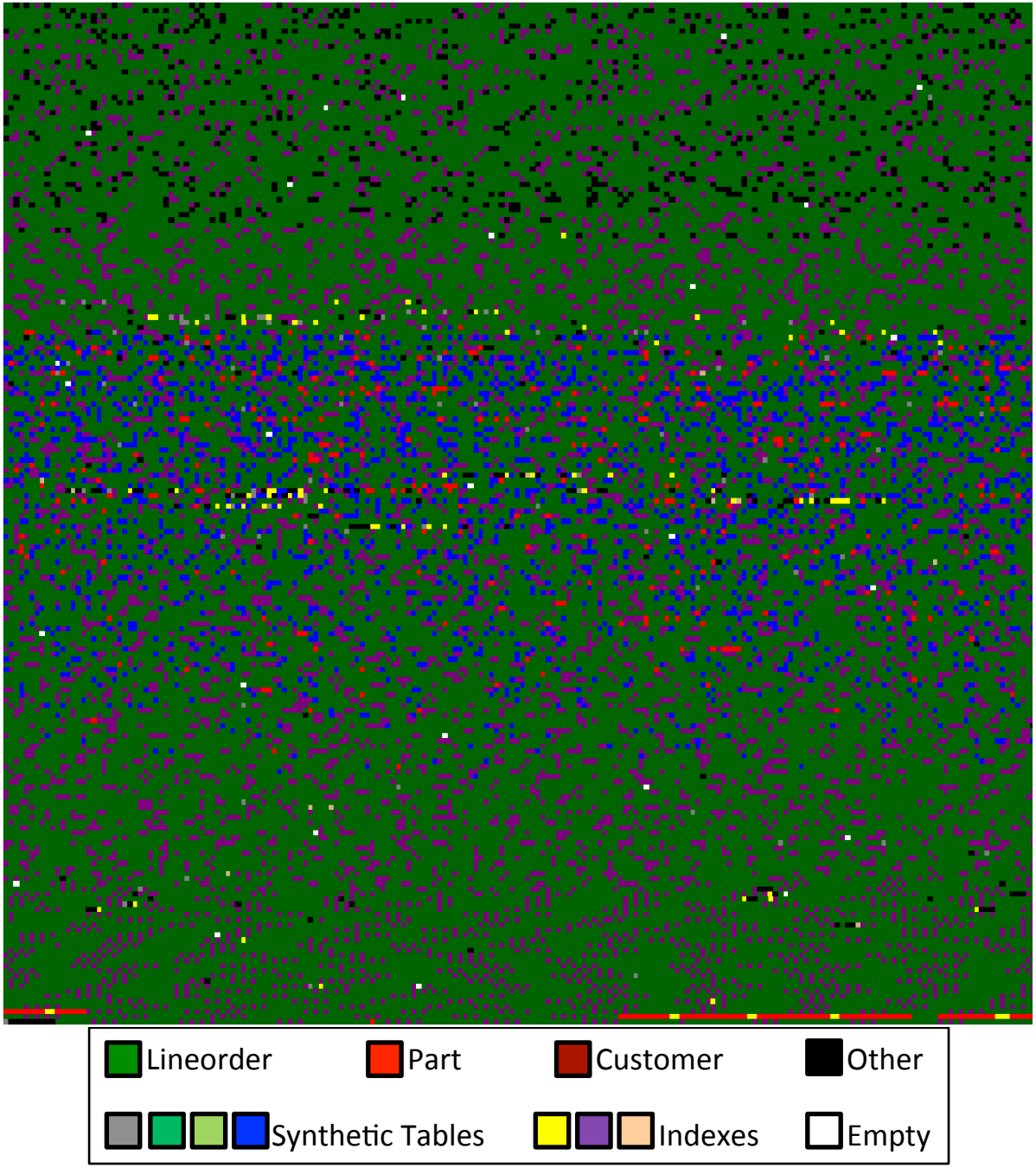
- An AWS instance + PostgreSQL
  1. Load SSBM benchmark data
  2. Delete the rows, then **delete all Postgres files**

	Damage = 0%	Damage = 10%	Damage = 25%
Dwdate	**35** (100%)	**31** (88.6%)	**20** (57.1%)
Supplier	**565** (100%)	**455** (80.5%)	**326** (57.7%)
Customer	**1915** (100%)	**1559** (81.4%)	**1075** (56.1%)
Part	**8659** (100%)	**6969** (80.5%)	**4864** (56.2%)
Lineorder	**115 K** (100%)	**104 K** (89.9%)	**87 K** (75.2%)
TOTAL	**416 K** (100%)	**375 K** (89.9%)	**312 K** (74.9%)


Lineorder
Part
Customer
Other
Synthetic Tables
Indexes
Empty


Lineorder
Part
Customer
Other
Synthetic Tables
Indexes
Empty


Lineorder
Part
Customer
Other
Synthetic Tables
Indexes
Empty


Lineorder
Part
Customer
Other
Synthetic Tables
Indexes
Empty

Data Modification

- DELETE/UPDATE/INSERT
- When is a value **really** deleted?
- Example
  - Customer table (Phone# column)
  - Indexed Phone#

Row Deletion

Event	Table		Index	
	HDD	RAM	HDD	RAM
$Time_0$	☎		✓	
Delete a Phone #	✓	✗	✓	✓
Run a few of queries	✓	✗	✓	
Many more queries	✗		✓	
Rebuild the index	✗			
Rebuild the table				

Roadmap

- Motivation
  - File carving (for databases)
  - Stochastic analysis
- Results
- Future work

Future Work

- Column-stores and Key-value stores
- Automated database deconstruction
- Database performance tuning
- Monitoring user behavior (DB cache)
- Independent database audit/verification


time for questions