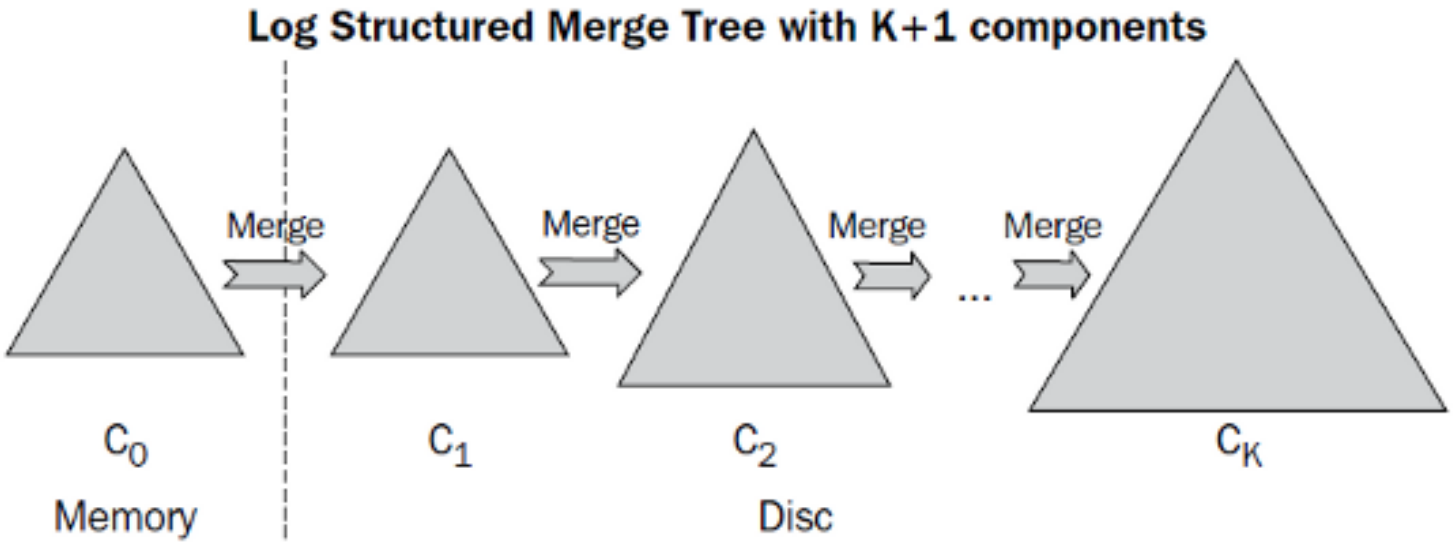
Хранение данных на виниле

kostja@tarantool.org
Konstantin Osipov

План доклада

- вводный обзор log structured merge tree
- устройство vinyl engine в Tarantool
- варианты использования
- планы

Классическое LSM



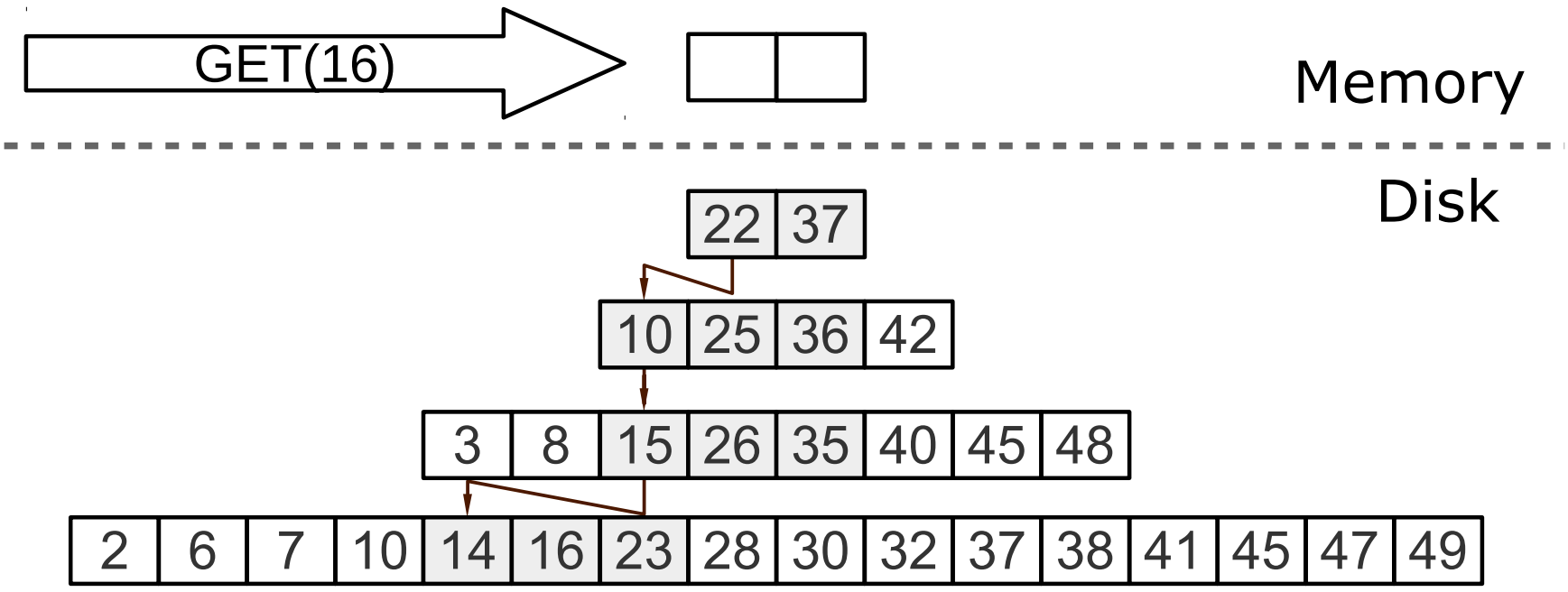
DELETE в LSM tree

- нет возможности удалять из всех уровней
- вставляется маркер удаления

SELECT в LSM tree

- ищет во всех уровнях, пока не найдёт нужный ключ
- оптимистичный поиск для point selet
- слияние результатов поиска для range select

SELECT в LSM tree

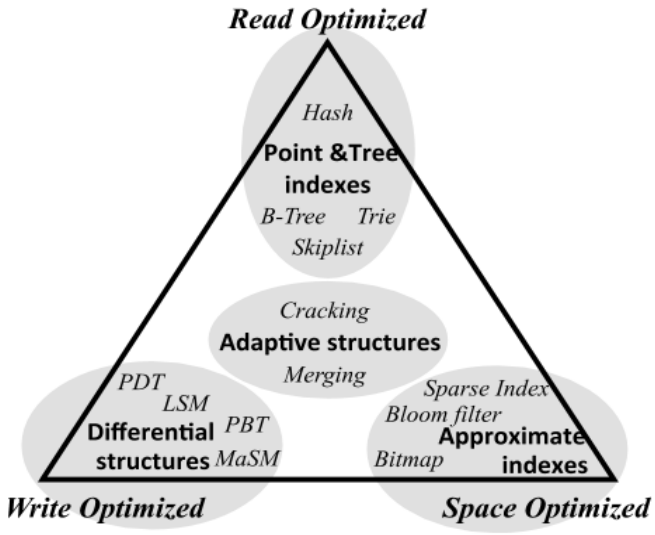


LSM: алгортмический потолок

	LSM tree	**B-tree**
Search	$K * O(\log_2 N/B)$	$O(\log_B N)$
Delete	$O(\log_2(N)/B)$	$O(\log_B N)$
Insert	$O(\log_2(N)/B)$	$O(\log_B N)$

RUM conjecture

The ubiquitous fight between the Read, the Update, and the Memory overhead of access methods for modern data systems



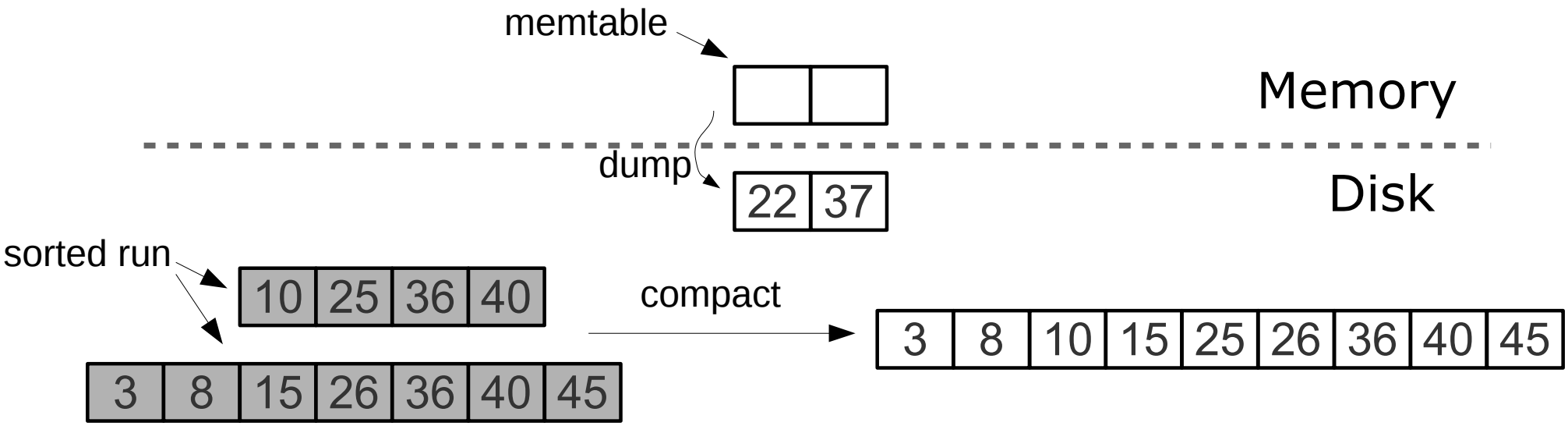
Проблемы применения LSM в Web/OLTP

- медленные чтения: **read amplification**
- потенциально очень много пишем на диск, много больше чем сами данные: **write amplification**
- храним мусор: **space amplification**
- непредсказуемая работа в моменты перестроения индекса: **latency spikes**

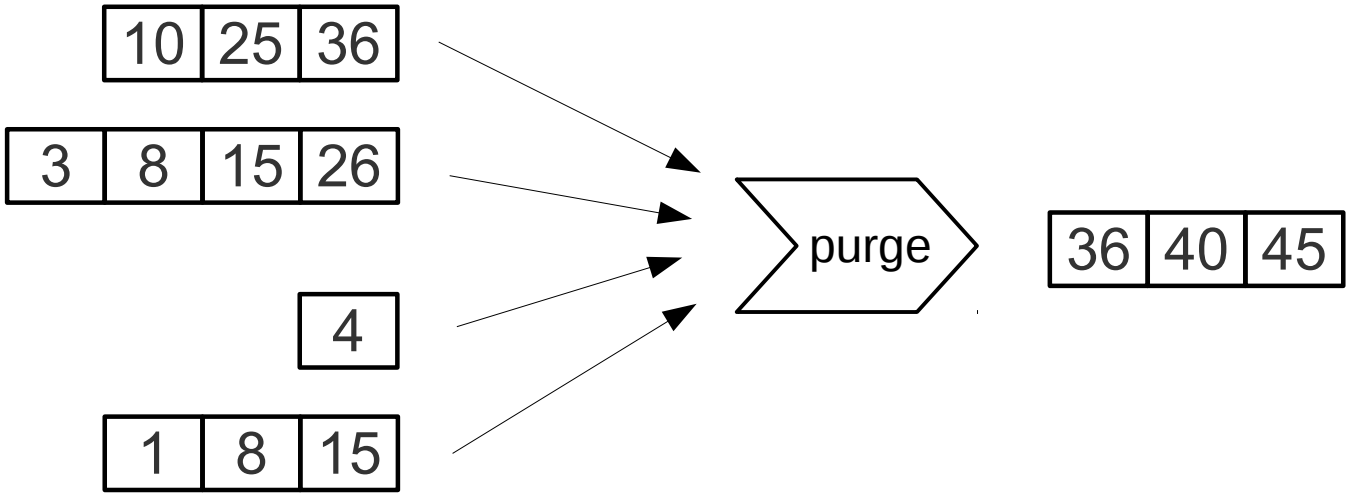
Vinyl: memtable, sorted run, dump & compact

- хранит операции, а не данные:
  - REPLACE,
  - DELETE,
  - UPSERT
- каждое значение помечено LSN
- append-only, garbage collection, message passing

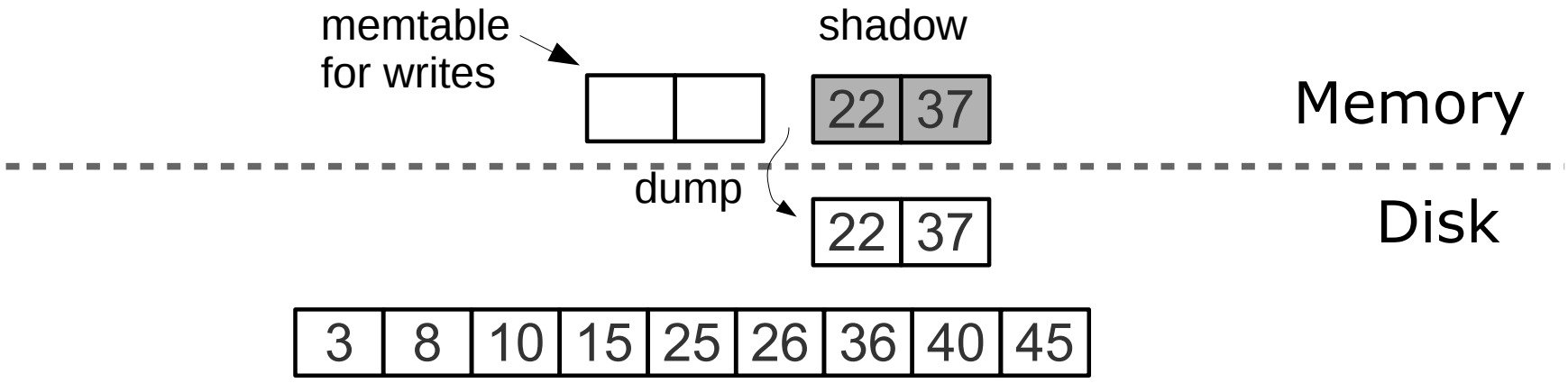
Memtable, sorted run, dump & compact



Vinyl: чтение

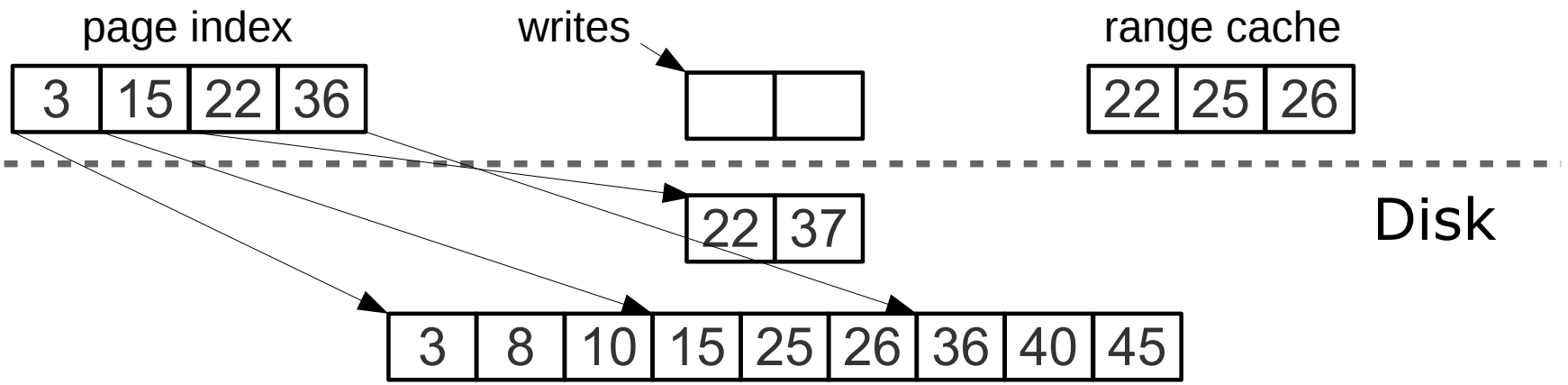


Делаем latency предсказуемым



- упреждающие dumpы
- throttling

Снижаем read amplification



- page index
- tuple range cache
- multi-level compaction

Снижаем write amplification

- multi-level compaction
- ranges, split around median key

Снижаем space amplification

- ranges

Поддержка вторичных ключей

- хранят значение вторичного и первичного ключа
- нивелируют преимущества REPLACE/UPSERT

UPSERT

```
space_object:upsert(tuple_value, {{operator, field_no, value}, ...})
```

- non-reading update or insert
- отложенное выполнение
- есть только в Tarantool
- squashing

Планировщик

- единственная активная сущность в Vinyl
- обрабатывает две очереди: dump и merge
- очереди содержат все ranges

Transactions

- MVCC
- the first transaction to commit wins
- no waits, no deadlocks, only aborts
- yields don’t abort transactions

Репликация

Работает из коробки

Ограничения

- небольшие таплы
- помните об оверхеде вторичных ключей
- несколько тысяч файловых дескрипторов
- просто планировщик транзакций: долгие read-write транзакции могут абортиться
- нет cross-engine транзакций
- space amplification требует до 2x резервирования

Ссылки

Leveled compaction in Apache Cassandra
http://www.datastax.com/dev/blog/leveled-compaction-in-apache-cassandra
http://www.scylladb.com/kb/compaction/
https://github.com/facebook/rocksdb/wiki/Universal-Compaction
https://dom.as/2015/04/09/how-innodb-lost-its-advantage/

Спасибо!


@kostja_osipov


fb.com/TarantoolDatabase


kostja@tarantool.org


www.tarantool.org

Делаем latency предсказуемым

- упреждающие dumpы
- throttling