МАЛЕНЬКИЕ ДАННЫЕ

начинают расти

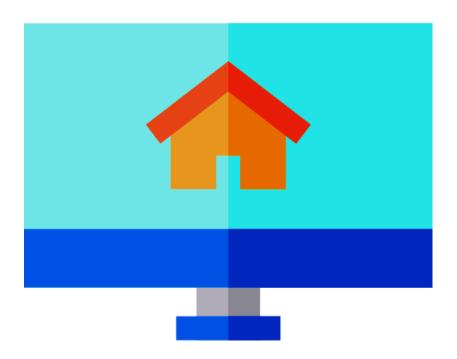
Томак Ксения.

Старший программист

ЗАО «Связь инжиниринг М»

Предметная область

Сбор данных о потреблении электричества и мониторинг состояния объектов.



С чего все начиналось





SQL. Пытаемся спасти ситуацию.



- Перегруженный сервер
- Неоптимальный запрос
- Плохая архитектура
- Нет индекса
- Фрагментация индекса

SQL. Перегруженный сервер

System resource monitor

SQL Server Activity monitor

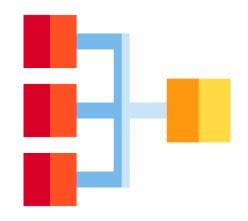
SQL Server instance standard reports

System views (sys.*)

SQL. Неоптимальный запрос



Построение индекса



Вынос агрегатов



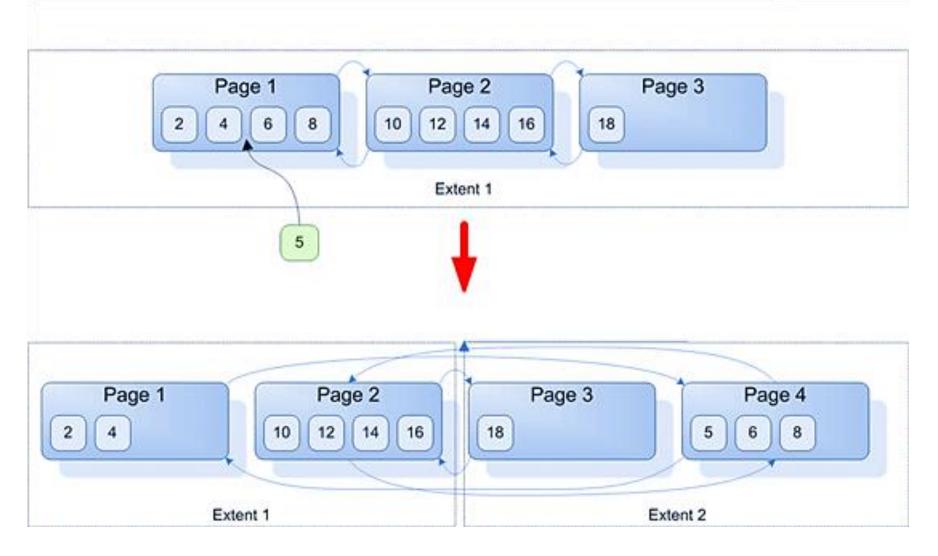
А оно здесь надо?

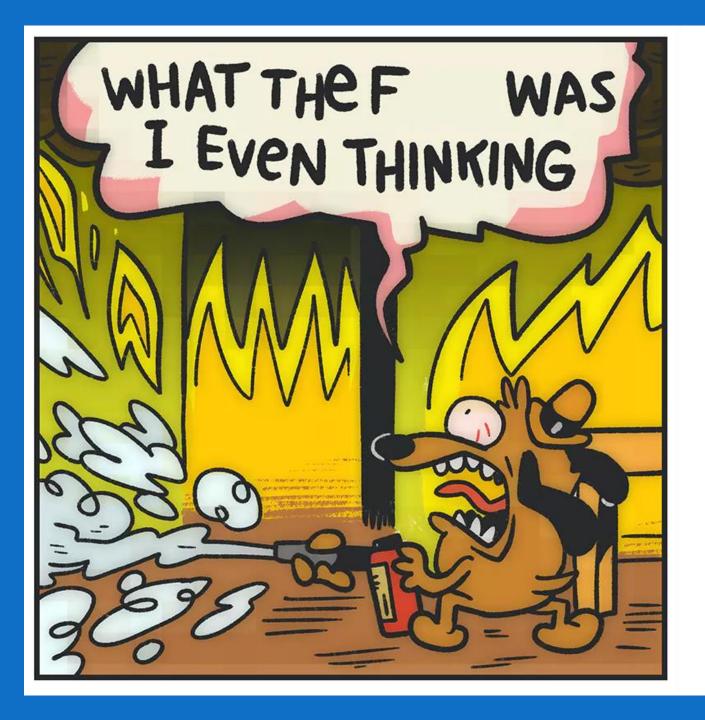
SQL. Неоптимальный запрос

```
Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [dbo].[vTimeSeries]
Missing Index (Impact 64.2527): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>]
                                        Clustered Index Scan (Clustered)
                                       [DimProduct].[PK DimProduct Product...
                 (Inner Join)
Cost: 0
                  Cost: 11 %
                                                  Cost: 14 %
                                                   Hash Match
                                                                         Clustered Index Scan (Clustered)
                                                  (Inner Join)
                                                                        [DimDate].[PK DimDate DateKey] [d]
                                                   Cost: 13 %
                                                                                    Cost: 1 %
                                                                         Clustered Index Scan (Clustered)
                                                                        [FactInternetSales].[PK FactInterne...
                                                                                    Cost: 60 %
```

SET STATISTICS IO { ON | OFF }
SET STATISTICS TIME { ON | OFF }

SQL. Индекс и фрагментация индекса



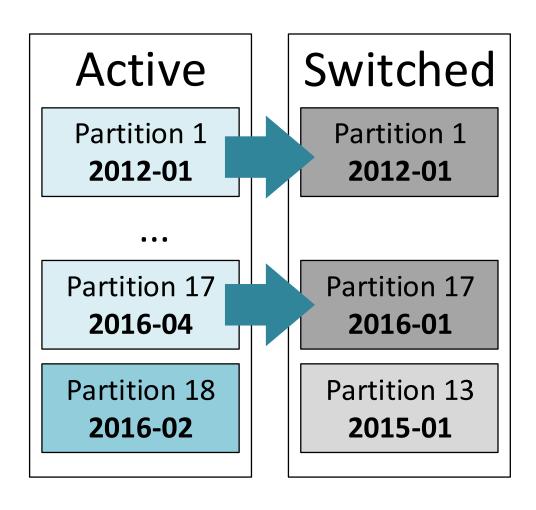


SQL. Секционирование

SQL. Секционирование

Делим таблицу на секции по каждому кварталу

SQL. Секционирование. Переключение секций



Partition Number	Filegroup Name	Active	Switched
1	VALUE-2012-1	2011	4296800
2	VALUE-2012-2	0	10195255
3	VALUE-2012-3	0	13876984
4	VALUE-2012-4	1	22937812
5	VALUE-2013-1	0	29097347
6	VALUE-2013-2	0	32452467
7	VALUE-2013-3	0	34200711
8	VALUE-2013-4	0	37956789
9	VALUE-2014-1	6	47557847
10	VALUE-2014-2	0	79331901
11	VALUE-2014-3	0	123221691
12	VALUE-2014-4	0	251419041
13	VALUE-2015-1	0	325291391
14	VALUE-2015-2	154	349616295
15	VALUE-2015-3	31863	367851397
16	VALUE-2015-4	31894	379660056
17	VALUE-2016-1	83928	400804206
18	VALUE-2016-2	633283910	0

SQL. Секционирование. Bulk inserts

```
USE [DB]
GO
ALTER DATABASE [DB] SET RECOVERY BULK LOGGED
GO
INSERT [dbo].[Table] with(TABLOCK)
   SELECT *
  FROM [dbo].[NonPartitionedTable]
GO
                                            Transaction Log
```

ALTER DATABASE [DB] SET RECOVERY FULL GO

SQL. Очистка таблицы

TRUNCATE TABLE db.schema.table

*имеет ряд ограничений

SQL. Секционирование

Pros

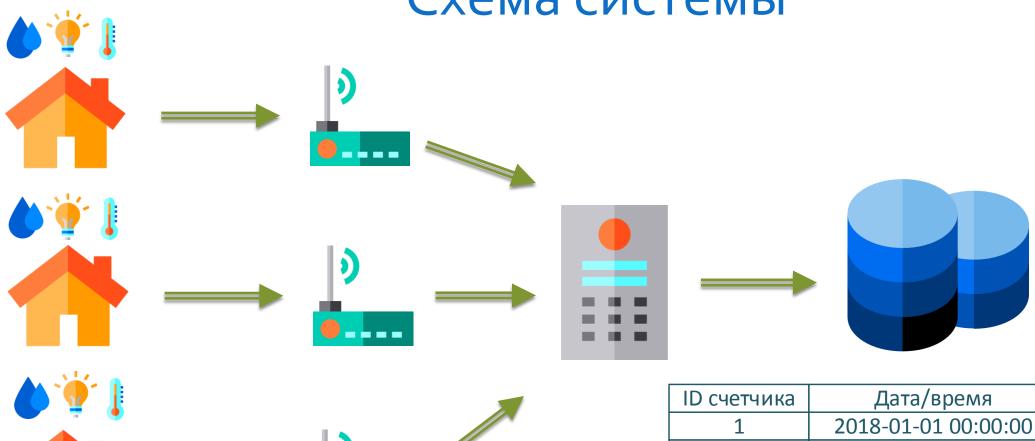
- Секция − отдельная компонента обслуживания
- Включение/выключение секций
- Быстрое удаление всей секции, вставка в пустые секции

- Администрирование
- Неравномерное распределение записей
- До SQL Server 2016 − в EE версии

Смена хранилища

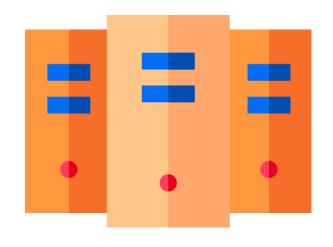


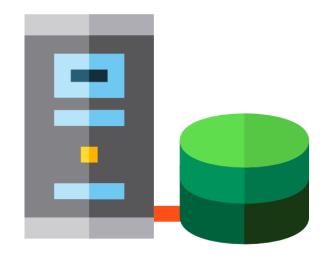
Схема системы

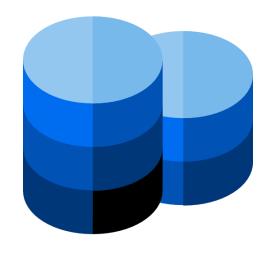


ID счетчика	Дата/время	Показание
1	2018-01-01 00:00:00	10.0
1	2018-01-01 00:30:00	10.5
•••	•••	•••
150000	2018-06-01 13:30:00	12455.1

Осистеме







Около 3о различных продакшенов

SQL server 2008+ MongoDB 3.0+ 7 млрд записей интенсивность вставок до 200 записей/сек

Пример сервера БД

CPU

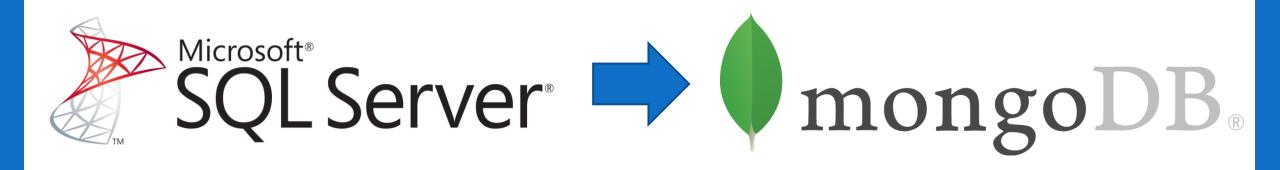
12 CPUs (2.66 GHz) **RAM**

36Gb

Disk

HDD Fiber Channel

Смена хранилища



Выбор хранилища Подготовка бекенда

Перенос данных

Выбор хранилища



Работа с БД (или кластером)



Сообщество



Драйвер .NET



Хорошая документация

Выбор хранилища

Параметр	Примечание
CAP	A - 😵 (failover ~ max 5 minutes)
ACID	(AC - single document, ID - depends)
Indexes	
Queries	(range queries, where)
In-Memory	
Concurrency	Optimistic

MongoDB 3.0

New storage engine (WiredTiger)

MongoDB 3.2

Enhanced replication protocol: stricter consistency & durability

WiredTiger default storage engine

Config server manageability improvements

Read concern majority

MongoDB 3.4

Shard membership awareness

MongoDB 3.6

Consistent secondary reads in sharded clusters

Logical sessions

Retryable writes

Causal Consistency

Cluster-wide logical clock

Storage API changes to use timestamps

Read concern majority feature always available

Collection catalog versioning

Make collection drops two phase

UUIDs in sharding

Fast in-place updates to large documents in WT

MongoDB 4.0

REPLICA-SET TRANSACTIONS

Storage support for prepared transactions

Make catalog timestamp-aware

Replica set point-in-time reads

Recoverable rollback via WT checkpoints

Recover to a timestamp

Sharded catalog improvements

MongoDB 4.2

SHARDED TRANSACTIONS

Transaction - compatible chunk migration

More extensive WiredTiger repair

Transaction manager

Global point-in-time reads

Oplog applier prepare support for transactions

...

Legend

DONE

IN PROGRESS

PLANNED

TRANSACTION EPIC

Тесты скорости







- ✓Вставки в непустую таблицу
- ✓Стараться эмулировать естественную нагрузку
- ✓Перед тестами прочитать мануалы и сконфигурировать СУБД ©

Конфигурация хранилища и драйвера

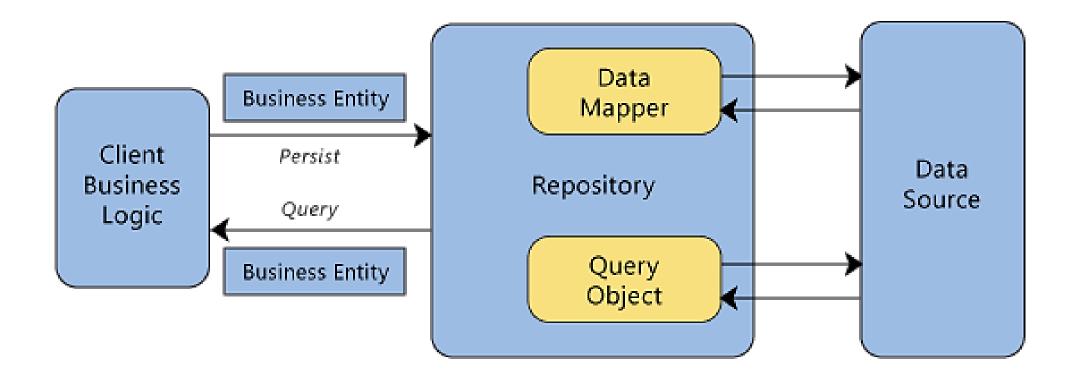
• Помнить о структуре данных

```
{
   "datetime": "2017-01-01T00:00:00",
   "equipmentid": 101,
   "value": 0
}

{
   "d": "2017-01-01T00:00:00",
   "e": 101,
   "v": 0
}
```

- Настройки логирования
- Настройки драйвера БД

Подготовка backend'a

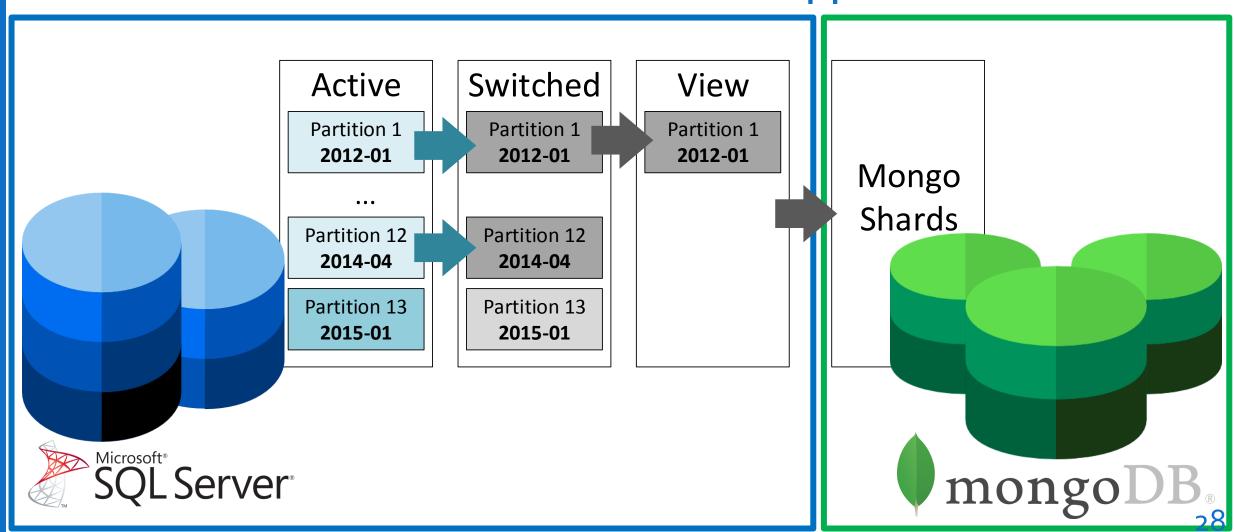


Перенос данных

Если инструмента переноса нет:

- Важны не фасады, а скорость;
- ADO вместо EF;
- Разные уровни логирования;
- Возможность начать перенос с какого-то ключа (shit happens).

Перенос данных. Большое количество данных

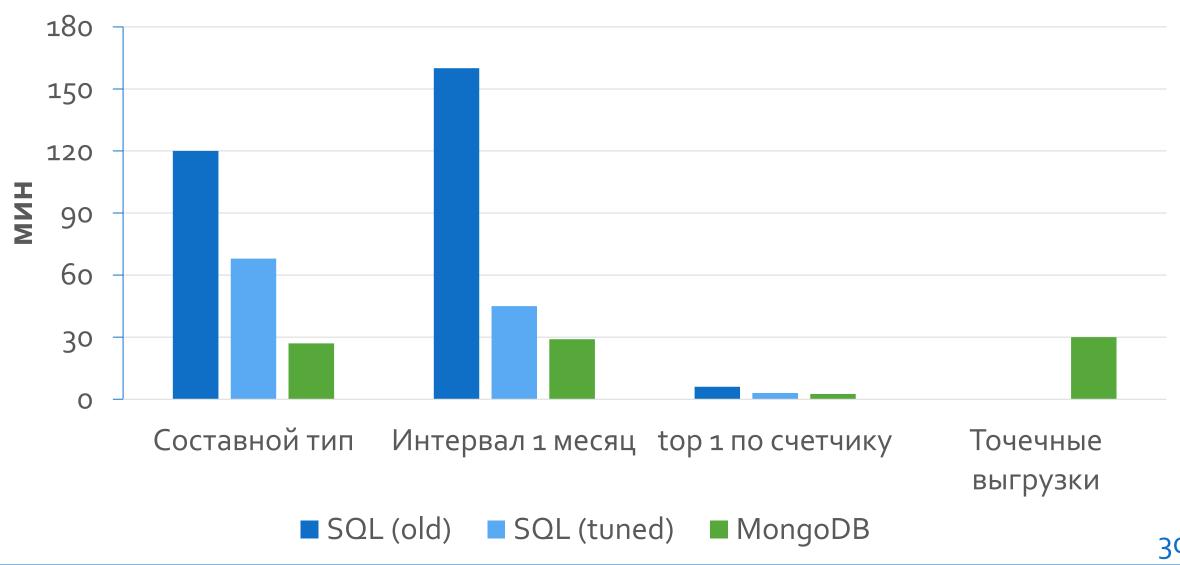


Смена хранилища с SQL на MongoDB Pros

- 🗸 Скорость
- Не нужна лицензия MS SQL
- Нужно меньше ресурсов сервера
- Гибкая схема БД

- 😵 Другой подход
- **М**ного конфигурации
- 🗴 Целостность

Скорости выгрузки типовых отчетов



Итоги



Скорость выгрузки основного представления выросла в 60 раз.



Скорость выгрузки отчетов выросла в 1,5-2 раза.



Ресурсов сервера хватило еще на х6 объемов данных.



Размер БД уменьшился в 1,5 раза

SQL. Пытаемся спасти ситуацию. Почитать

- Ritesh Shah, Bihag Thaker Microsoft SQL Server 2012 Performance Tuning Cookbook
- Strate J., Krueger T. Expert Performance Indexing for SQL Server 2012 (The Expert Voice in SQL Server)
- Kalen Delaney's Weekly Webinar Episode 4 (Data storage) and later
- BRENT OZAR
- О самой частой причине выбора неэффективного плана запроса. Часть 1.
- <u>Check SQL Server a specified database index fragmentation</u> percentage (но можно дополнить до генерации dynamic SQL)

SQL. Секционирование. Почитать

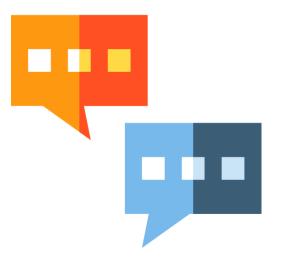
- Partitioned Table and Index Strategies Using SQL Server 2008
- Project REAL: Data Lifecycle Partitioning
- <u>SQL Server Table Partitioning: Resources</u>

Смена хранилища. Почитать

- NoSQL Distilled
- Polyglot Persistence
- Monolithic Persistence antipattern
- Документации ваших любимых СУБД ©



Вопросы



Томак Ксения tomak.ksenia@gmail.com