When to NoSQL and When to Know SQL

Simon Elliston Ball
Head of Big Data

@sireb
#nosqlknowsql

http://nosqlknowsql.io
what is NoSQL?

SQL
NoSQL
Not only SQL
No, SQL

Many many things
before SQL

files

multi-value

ur... hash maps?
everything is relational
ORMs fill in the other data structures
scale up rules
data first design
and now NoSQL

datastores that suit applications

polyglot persistence: the right tools

scale out rules

APIs not EDWs
why should you care?

data growth

rapid development

fewer migration headaches... maybe

machine learning

social
big bucks.

Salary figures are for US respondents only.

Given the two patterns we have just examined—the relationships between cluster tools and respondents' overall tool counts, and between tool counts and salary—it should not be surprising that there is a significant difference in how each cluster correlates with salary. Using more tools from the Hadoop cluster correlates positively with salary, while using more tools from the SQL/Excel cluster correlates (slightly) negatively with salary.

O’Reilly 2013 Data Science Salary Survey
So many NoSQLs...
document databases

mongoDB

RAVENDB

Couchbase
document databases

rapid development

JSON docs

complex, variable models

known access pattern
document databases

learn a very new query language
denormalize
document form

joins? JUST DON’T

document vs SQL

what can SQL do?

query all the angles

sure, you can use blobs…

… but you can’t get into them
documents in SQL

SQL xml fields
mapping xquery paths is painful

native JSON

but still structured
query everything: search

class of database database

full-text indexing
you know google, right...

range query
span query
keyword query
you know the score

```
"query": {
  "function_score": {
    "query": {
      "match": { "title": "NoSQL" }
    },
    "functions": [
      "boost": 1,
      "gauss": {
        "timestamp": {
          "scale": "4w"
        }
      },
      "script_score": {
        "script": "_score * doc['important_document'].value ? 2 : 1"
      }
    ],
    "score_mode": "sum"
  }
}
```
SQL knows the score too

declare @origin float = 0;
declare @delay_weeks float = 4;

SELECT TOP 10 * FROM (  
    SELECT title,
    score *
    CASE
        WHEN p.important = 1 THEN 2.0
        WHEN p.important = 0 THEN 1.0
    END
    * exp(-power(timestamp-@origin,2)/(2*@delay*7*24*3600)) + 1
    AS score
    FROM posts p
    WHERE title LIKE '%NoSQL%'
) as found
ORDER BY score
you know google, right...

more like this: instant tf-idf

```json
{
    "more_like_this": {
        "fields": ["name.first", "name.last"],
        "like_text": "text like this one",
        "min_term_freq": 1,
        "max_query_terms": 12
    }
}
```
Facets
SELECT a.name, count(p.id) FROM people p
  JOIN industry a on a.id = p.industry_id
  JOIN people_keywords pk on pk.person_id = p.id
  JOIN keywords k on k.id = pk.keyword_id
WHERE CONTAINS(p.description, 'NoSQL')
  OR k.name = 'NoSQL'
...  
GROUP BY a.name

SELECT a.name, count(p.id) FROM people p
  JOIN area a on a.id = p.area_id
  JOIN people_keywords pk on pk.person_id = p.id
  JOIN keywords k on k.id = pk.keyword_id
WHERE CONTAINS(p.description, 'NoSQL')
  OR k.name = 'NoSQL'
...  
GROUP BY a.name
Elastic search:
{
    "query": {
        "query_string": {
            "default_field": "content",
            "query": "keywords"
        }
    },
    "facets": {
        "myTerms": {
            "terms": {
                "field": "lang",
                "all_terms": true
            }
        }
    }
}
logs

untyped free-text documents

timestamped

semi-structured

discovery

aggregation and statistics
key: value

close to your programming model

distributed map | list | set

keys can be objects
SQL extensions

hash types
hstore
SQL and polymorphism

inheritance

ORMs hide the horror
turning round the rows

columnar databases

physical layout matters
turning round the rows

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>Home</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>Work</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>Work</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>Work</td>
</tr>
</tbody>
</table>

Row storage

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>Home</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>Work</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>Work</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>Work</td>
</tr>
</tbody>
</table>

Column storage

<table>
<thead>
<tr>
<th>key</th>
<th>value</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>Home</td>
<td>Work</td>
</tr>
<tr>
<td>Work</td>
<td>Work</td>
<td>Work</td>
</tr>
</tbody>
</table>
teaching an old SQL new tricks

MySQL  InfoBright  SQL Server  Columnar Indexes

```
CREATE NONCLUSTERED COLUMNSTORE INDEX idx_col
ON Orders (OrderDate, DueDate, ShipDate)
```

Great for your data warehouse, but no use for OLTP
millions of columns
eventually consistent
CQL
set | list | map types

http://cassandra.apache.org/
http://www.datastax.com/
column for hadoop and other animals

ORC files

Parquet  http://parquet.io
cell level security

SQL: so many views, so much confusion

accumulo https://accumulo.apache.org/
Time series
time

retrieving time series and graphs

window functions

SELECT business_date, ticker, 
close, 
close / 
    LAG(close,1) OVER PARTITION BY ticker ORDER BY business_date ASC) - 1 AS ret
FROM sp500
CREATE procedure [dbo].[Dequeue] AS
set nocount on

declare @BatchSize int
default @BatchSize = 10

declare @Batch table (QueueID int, QueueDateTime datetime, Title nvarchar(255))

begin tran
insert into @Batch
select Top (@BatchSize) QueueID, QueueDateTime, Title from QueueMeta
WITH (UPDLOCK, HOLDLOCK)
where Status = 0
order by QueueDateTime ASC

declare @ItemsToUpdate int
set @ItemsToUpdate = @@ROWCOUNT

update QueueMeta
SET Status = 1
WHERE QueueID IN (select QueueID from @Batch)
AND Status = 0

if @@ROWCOUNT = @ItemsToUpdate
begin
commit tran
select b.*, q.TextData from @Batch b
inner join QueueData q on q.QueueID = b.QueueID
print 'SUCCESS'
end
else
begin
rollback tran
print 'FAILED'
end
queues in SQL

index fragmentation is a problem

but built in logs of a sort
message queues

- specialised apis
- capabilities like fan-out
- routing
- acknowledgement
relationships count

Graph databases

Neo4j

GraphX

Apache Giraph
relationships count

trees and hierarchies

overloaded relationships

fancy algorithms
hierarchies with SQL

adjacency lists

materialised path

nested sets (MPTT)

```
CONSTRAIN fk_parent_id_id
FOREIGN KEY parent_id REFERENCES some_table.id
```

```
path = 1.2.23.55.786.33425
```

<table>
<thead>
<tr>
<th>Node</th>
<th>Left</th>
<th>Right</th>
<th>Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>22</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>G</td>
<td>11</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>H</td>
<td>19</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>I</td>
<td>12</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>J</td>
<td>14</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>K</td>
<td>16</td>
<td>17</td>
<td>4</td>
</tr>
</tbody>
</table>
Big Data
SQL on Hadoop

- Hive
- Hawq
- CitusData
- Hadapt
- Spark
More than SQL

Shark
Drill
Cascading
Map Reduce
System issues,
Speed issues,
Soft issues
the ACID, BASE litmus

Atomic
Consistent
Isolated
Durable

Basically Available
Soft-state
Eventually consistent

what matters to you?
CAP it all

Consistency

Partition

Availability
write fast, ask questions later

SQL writes cost a lot
mainly write workload: NoSQL
low latency write workload: NoSQL
is it web scale?

most NoSQL scales well

but clusters still need management

are you facebook?

one machine is easier than n

can ops handle it?

app developers make bad admins
who is going to use it?

analysts: they want SQL

developers: they want applications

data scientists: they want access
choose the right tool
Thank you!

Simon Elliston Ball
simon@simonellistonball.com

@sireb
#noSQLknowSQL

http://nosqlknowsql.io
Questions

Simon Elliston Ball
simon@simonellistonball.com

@sireb

http://nosqlknowsql.io