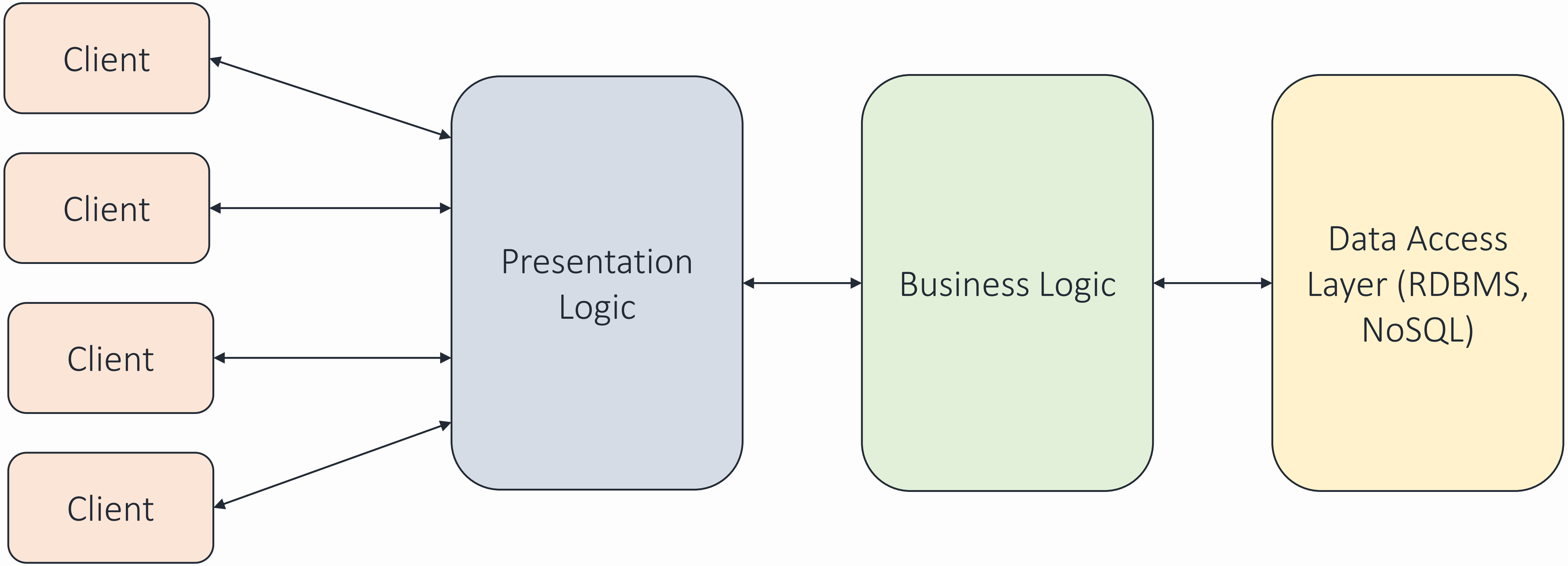
# Leveraging in-memory computation: Using Spark for textual queries

Presented by: Tejas Kannan

Date: 28/11/2018

# Traditional Applications

Complex textual queries are generally expensive to run on traditional database platforms



# Elasticsearch[1] Background

Elasticsearch relies on inverted indexes to enhance search efficiency

1. winter is coming
2. yours is the fury
3. the choice is yours

→

| Term | Frequency | Documents |
| --- | --- | --- |
| choice | 1 | 3 |
| coming | 1 | 1 |
| fury | 1 | 2 |
| is | 3 | 1,2,3 |
| the | 2 | 2 |
| winter | 1 | 1 |
| yours | 2 | 2,3 |

[1]Elasticsearch, https://www.elastic.co/
Example from: https://www.elastic.co/blog/found-elasticsearch-from-the-bottom-up

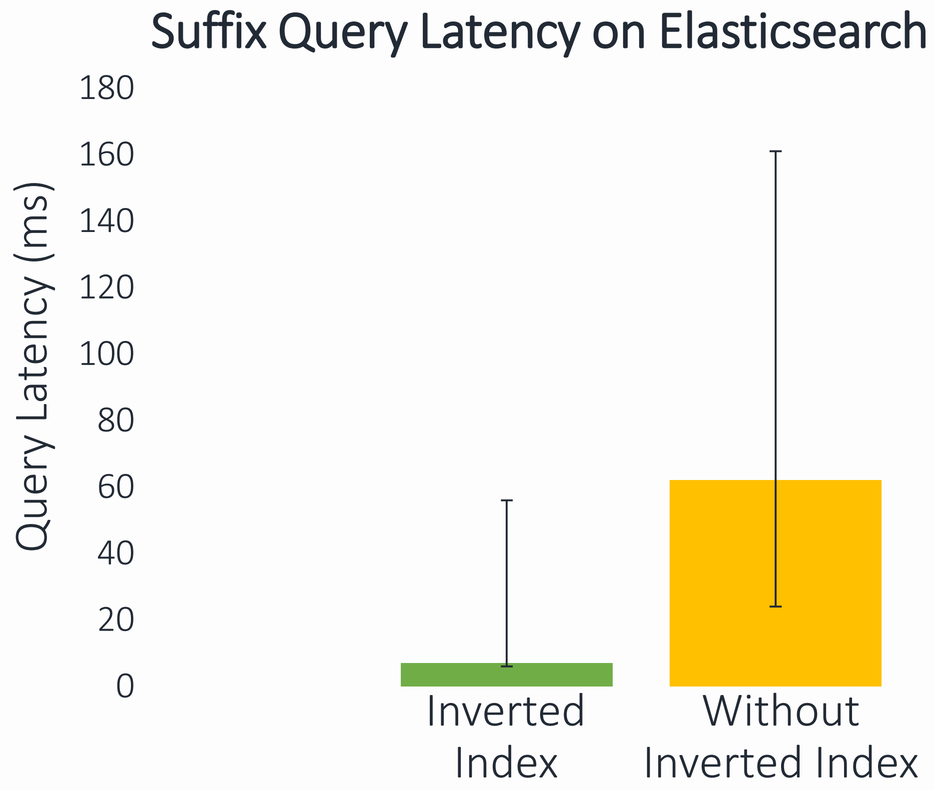
Elasticsearch requires...

...explicitly marking searchable fields at ingestion time

...dedicated index for each searchable field

# Initial Elasticsearch Benchmark

Inverted indexes provide large performance improvements at the expense of additional storage




| Raw Data Size | Elasticsearch Size with Inverted Index |
|---|---|
| 12.25 MB | 56.73 MB |

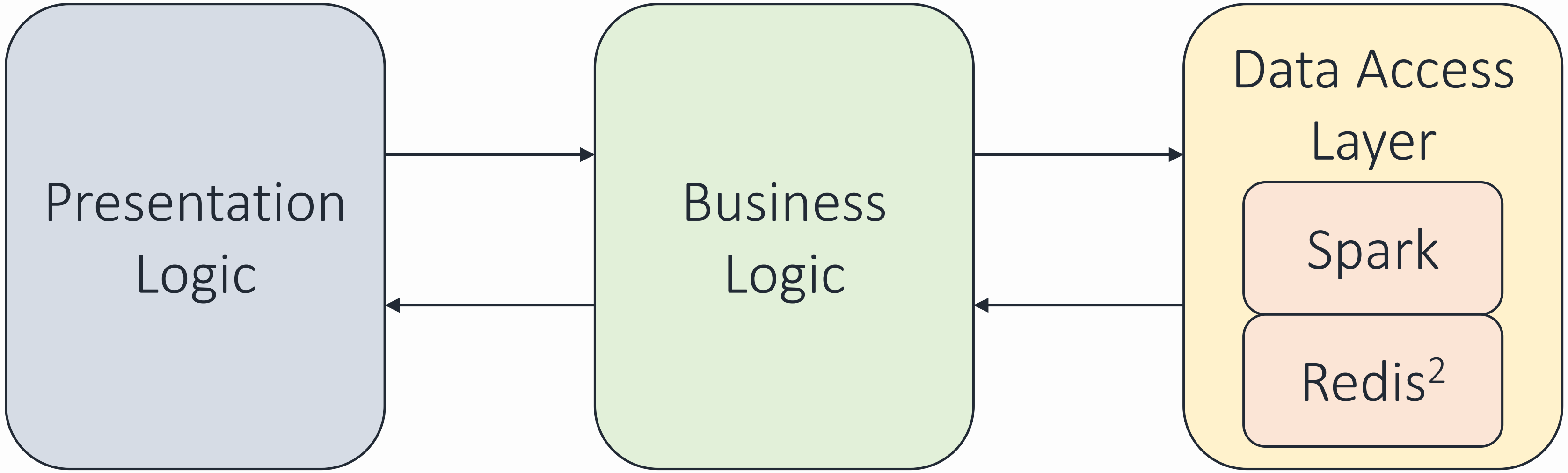
**Notes on Experiment:**

- 600,000 documents of actor/actress names from IMDb dataset[1]
- Queries were 2 character strings based on common English names
- Error bars represent $25^{th}$-$50^{th}$-$75^{th}$ percentiles; data collected from 1000 trials

[1]IMDb Dataset, https://www.imdb.com/interfaces/

# Using Spark[1] for Query Execution

Instead of requiring explicit indexes, we can try and use Spark as a computation engine for executing complex textual queries




Data Access Layer must use a persistent SparkContext to reduce job overhead

[1]Zaharia, Matei, et al. "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing." *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 2012.
[2]Redis, https://redis.io/

# Why might using Spark + Redis be a good idea?

- FiloDB is an open-source database which uses Spark as a computation engine on top of Cassandra for real-time stream analysis[1]

- Using Spark with Redis can provide over a 45x increase in performance over Spark + HDFS[2,3]

- Spark as a computation engine provides flexibility of query execution

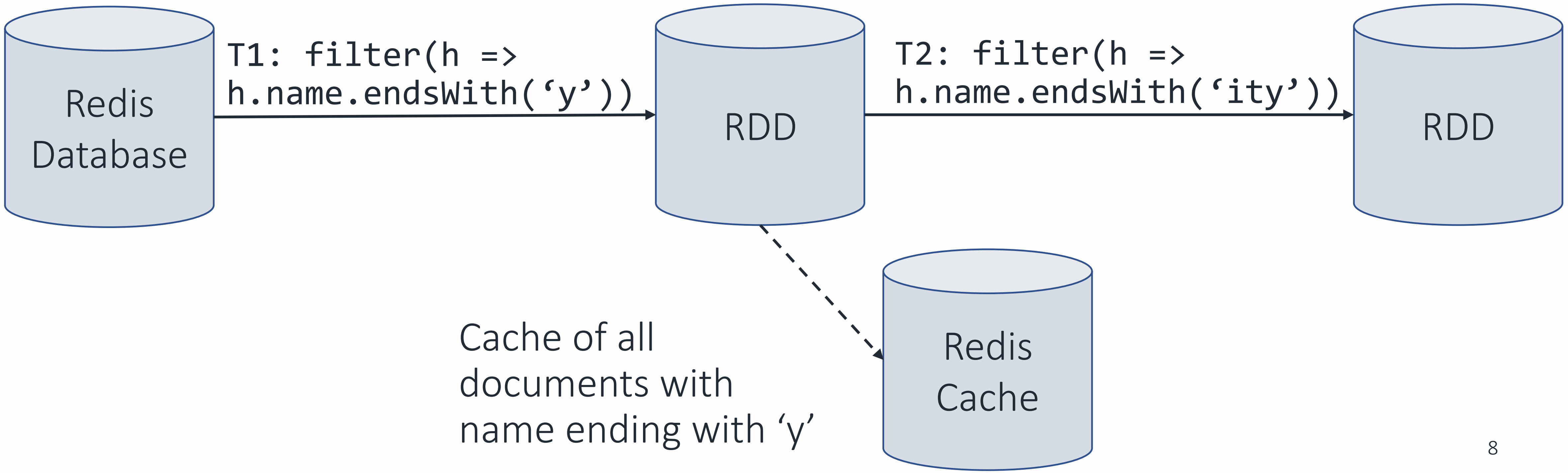
- By not requiring indexes for every searchable field, such a system can reduce the memory footprint

[1]FiloDB, https://velvia.github.io/Introducing-FiloDB/
[2]Shvachko, Konstantin, et al. "The hadoop distributed file system." *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*. Ieee, 2010.
[3]Redis Accelerates Spark by over 100 times, https://redislabs.com/press/redis-accelerates-spark-by-over-100-times/

# Caching Intermediate Results

Using some additional memory, we can cache intermediate results to speed up future queries

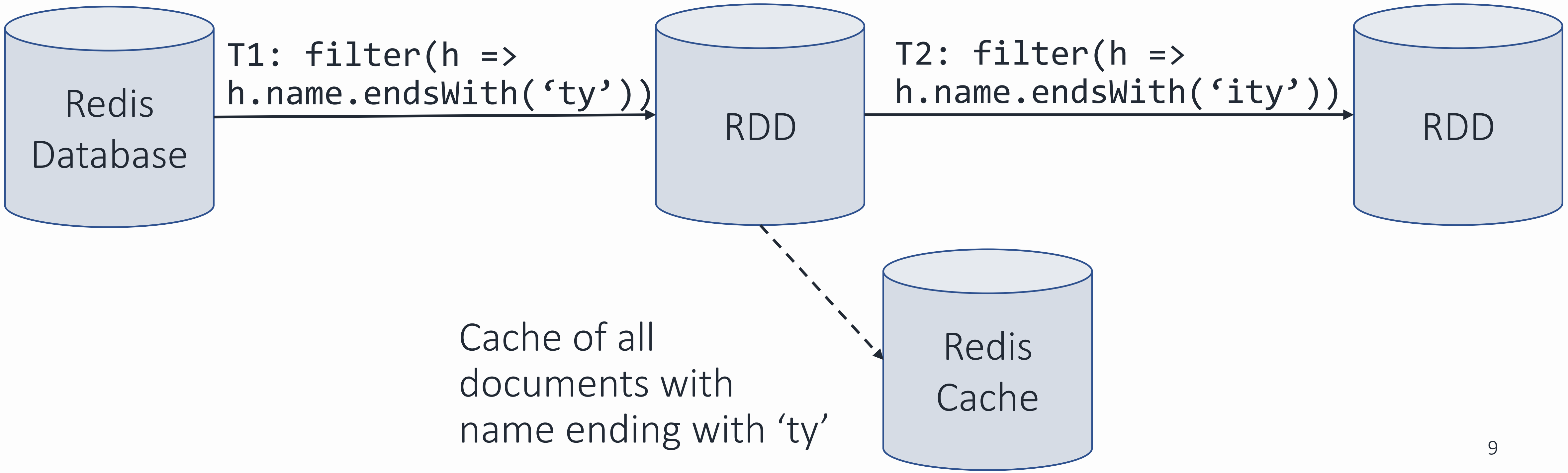
```
SELECT * FROM Hospitals WHERE name ends with “ity”
```



# Caching Intermediate Results

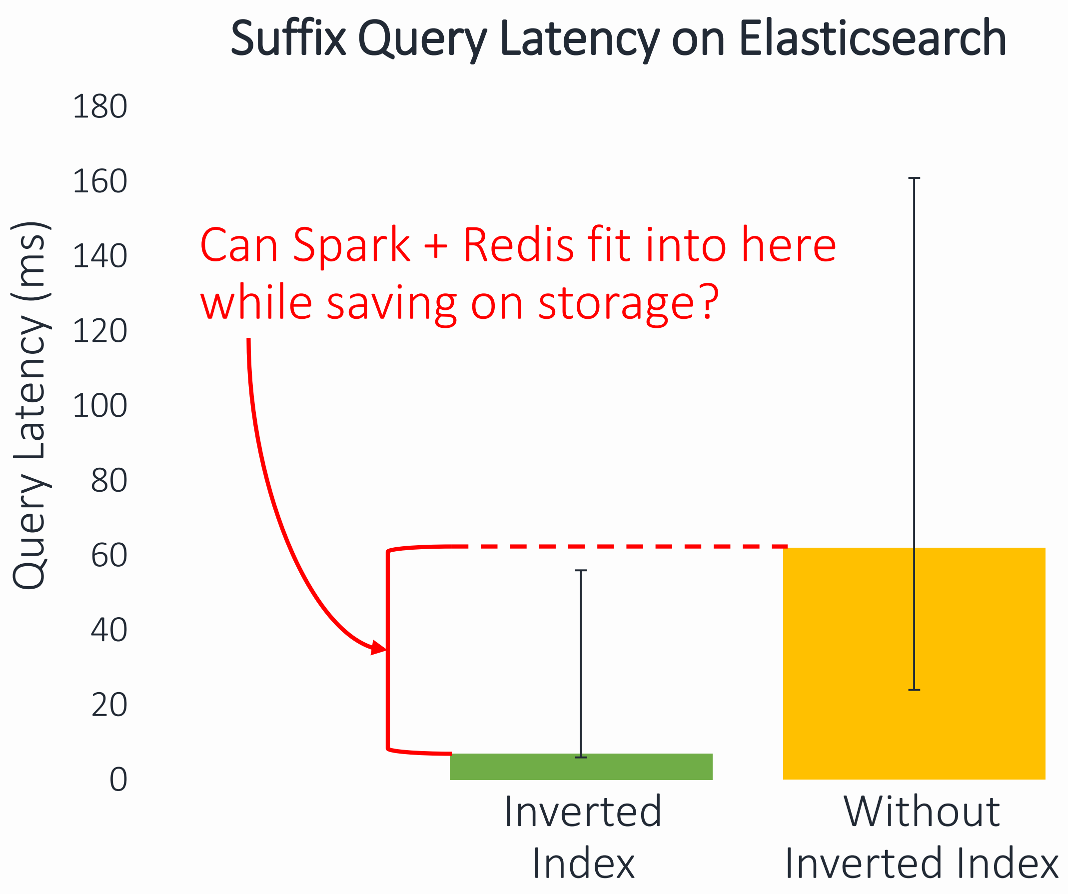
Caching patterns can be chosen based on common phrases to maximize effectiveness with limited memory

```
SELECT * FROM Hospitals WHERE name ends with "ity"
```



# Goals of this Project

- Create a Spark + Redis platform which can handle “prefix,” “suffix,” and “contains” queries
- Implement a caching feature using a configurable memory limit
- Benchmark the results against Elasticsearch to compare query latency and memory usage



# Questions?

# References


1. Elasticsearch, https://www.elastic.co/
2. Elasticsearch from the Bottom Up, https://www.elastic.co/blog/found-elasticsearch-from-the-bottom-up
3. FiloDB, https://velvia.github.io/Introducing-FiloDB/
4. IMDb Dataset, https://www.imdb.com/interfaces/
5. Redis, https://redis.io/
6. Redis Accelerates Spark by over 100 times, https://redislabs.com/press/redis-accelerates-spark-by-over-100-times/
7. Shvachko, Konstantin, et al. "The hadoop distributed file system." *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*. Ieee, 2010.
8. Zaharia, Matei, et al. "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing." *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 2012.