Dremel: Interactive Analysis of Web-Scale Datasets
Introduction

- Need: Large-scale data processing
- Challenge: Nested data formats
  - Expensive to normalise
Features & Contributions

- SQL-like querying syntax
- In-situ data processing to avoid data loading and transformations
- Serverless and multi-tenant
- Columnar format for nested data
  - Multi-level execution trees
Splitting records into columnar format

Repetition Level -
“At what repeated field in the field’s path has the value repeated?”

Definition Level -
“How many fields in the path that could be undefined (repeated/optional) are actually present?”
Repetition & Definition

Name.Language.Code

At what repeated field in the field’s path has the value repeated?
- Both Name and Language can be repeated.

How many fields in the path that could be undefined (repeated/optional) are actually present?
- Both Name and Language are undefined fields (repeated, in this case).
Repetition & Definition

Figure 1

```
message Document {
  required int64 DocId;
  optional group Links {
    repeated int64 Backward;
    repeated int64 Forward; }
  repeated group Name {
    repeated group Language {
      required string Code;
      optional string Country;
    }
    optional string Url; }}
```

Figure 2
Record assembly

Finite State Machine

- Each state is a field reader for the field in the query
- State transitions are repetition levels of the field
- For a subset of fields, construct a simpler FSM
Query Execution

- Root server and Leaf servers
- Partitions are called Tablets
- Scheduling Slots

Figure 4
Experimentation

Figure 5: Comparing execution time with MapReduce

Figure 6: Comparing execution time by varying the number of leaf servers
Experimentation

Q2: SELECT country, SUM(item.amount) FROM T2 GROUP BY country
Q3: SELECT domain, SUM(item.amount) FROM T2 WHERE domain CONTAINS ’.net’ GROUP BY domain

Figure 6: Execution times for Q2 and Q3
Summary

- SQL-like syntax
- In-situ data processing
- Serverless
- Columnar format for nested data
Discussion

- In-situ data processing unsuitable in case of external data management tools
- Columnar format for nested data encodes redundant information
- Record assembly for deeply nested data or large number of columns is inefficient
In recent years,

- Unifying framework for SQL dialects
- Hybrid approach with managed and in-situ data
- Query execution as a Directed Acyclic Graph, with a shuffle persistence layer

Thank you!