UMD DATA605 - Big Data Systems

Deploying an Application

Serialization Formats

- Programs need to send data to each other (on the network, on disk)
  - E.g., Remote Procedure Calls
  - Several recent technologies based around schemas
    - JSON, YAML, Protocol Buffer
- Serialization formats are data models

JSON

- JSON = JavaScript Object Notation
- Data is nested dictionaries and arrays
- Very similar to XML
  - More human-readable
  - Less boilerplate
  - Executable in JavaScript (and Python)

```
{
   "firstName": "John",
   "lastName": "Smith",
   "isAlive": true,
   "age": 25,
   "height_cm": 167.6,
   "address": {
      "streetAddress": "21 2nd Street",
      "city": "New York",
      "state": "NY",
      "postalCode": "10021-3100"
   },
   "phoneNumbers": [
      {
      "type": "home",
      "number": "212 555-1234”
      },
      {
      "type": "office",
      "number": "646 555-4567"
      }
   ],
   "children": [],
   "spouse": null
}
```

Protocol Buffers

- Developed by Google
- Open-source
- Represent data structures in:
  - Language agnostic
  - Platform agnostic
  - Versioning
- Schema is mostly relational
  - Optional fields
  - Types
  - Default values
  - Structures
  - Arrays
- Schema specified using a .proto file
- Compiled by protoc to produce C++, Java, or Python code to initialize, read, serialize objects

```
message Person {
  optional string name = 1;
  optional int32 id = 2;
  optional string email = 3;

  enum PhoneType {
    MOBILE = 0;
    HOME = 1;
    WORK = 2;
  }

  message PhoneNumber {
    optional string number = 1;
    optional PhoneType type = 2;
  }

  repeated PhoneNumber phones = 4;
}
```

```
import addressbook_pb2
person = addressbook_pb2.Person()
person.id = 1234
person.name = "John Doe"
person.email = "jdoe@example.com"
phone = person.phones.add()
phone.number = "555-4321"
   phone.type =
   addressbook_pb2.Person.HOME
```

Serialization Formats

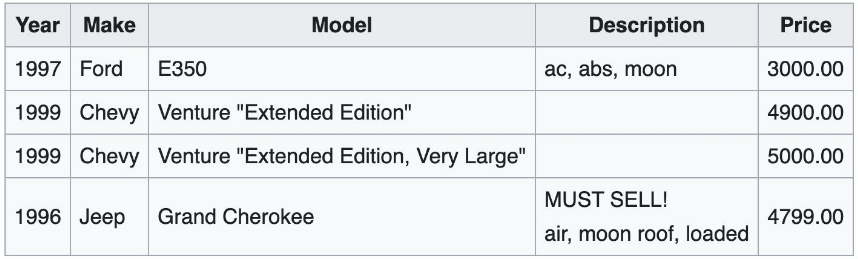
- Avro
  - Richer data structures
  - JSON-specified schema

```
{
      "namespace": "example.avro",
      "type": "record",
      "name": "User",
      "fields": [
                 {"name": "name", "type": "string"},
                 {"name": "favorite_number", "type": ["int", "null"]},
                 {"name": "favorite_color", "type": ["string", "null"]}
      ]
 }
```

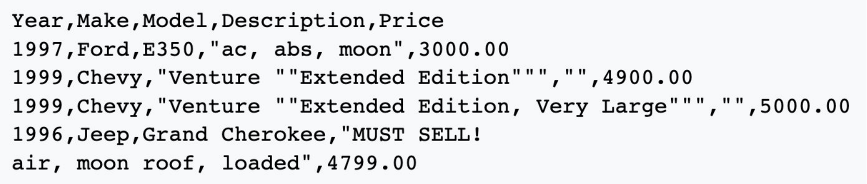
- Thrift
  - Developed by Facebook
  - Now Apache project
  - More languages supported
  - Supports exceptions and sets

Comma Separated Values (CSV)

- CSV stores data row-wise as text without schema
  - Each line of the file is a data record
  - Each record consists of one or more fields, separated by commas
- **Pros**
  - Very portable
    - It's text
    - Supported by every tool
  - Human-friendly
- **Cons**
  - Large footprint
    - Compression
  - Parsing is CPU intensive
  - No easy random access
  - No read only a subset of columns
  - No schema / types
    - Annotate CSV files with schema
  - Mainly read-only, difficult to modify



Year	Make	Model	Description	Price
1997	Ford	E350	ac, abs, moon	3000.00
1999	Chevy	Venture "Extended Edition"		4900.00
1999	Chevy	Venture "Extended Edition, Very Large"		5000.00
1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799.00



```
Year,Make,Model,Description,Price
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture ""Extended Edition""","",4900.00
1999,Chevy,"Venture ""Extended Edition, Very Large""","",5000.00
1996,Jeep,Grand Cherokee,"MUST SELL!
air, moon roof, loaded",4799.00
```

(Apache) Parquet

- Parquet allows to read tiles of data
  - That’s what the name comes from
- Supports multi-dimensional and nested data
  - A generalization of dataframes
- Column-storage
  - Each column is stored together, has uniform data type, and compressed (efficiently)
- Queries can be executed by IO layer
  - Only the necessary chunks of data is read from disk
- **Pros**
  - 10x smaller than CSV
  - 10x faster (with multi-threading)
  - You can read only a subset of columns and rows
- **Cons**
  - Binary, non-human friendly
  - Need ingestion step converting the inbound format to Parquet
  - Mainly read-only, difficult to modify

Python Pickle

- Pickle