Tips and Tricks for Data Engineering in the #GOshop

Miriah Peterson

weave

Bio

- Data Engineer at Weave in Lehi, Utah
- Board Member with Forge Foundation
- Proud Dog Mom
- Twitter @captainnobody1
- GitHub soypete
- LinkedIn

weave

Intro

> I am wondering if you actually get to use Go in your data pipelines or is it still mostly SQL/Python for you? I am curious how do you go about ensuring data quality in your data pipelines? Does Weave have a formal framework for monitoring/alerting on quality?

weave

Intro

Tips and Tricks I have learned doing Data Engineering in Go:

- Platform
- Extract Data
- Load Data
- Transform Data

weave

Platform

Platform: Infrastructure for storing, transfering, analyzing, and transforming data.

weave

Platform

If possible don't build your own home grown solution

beam

weave

Platform

Do not over engineer it

- Anticipate constraints and security
- Anticipate compliance
- Budget time to iterate on your design

weave

Platform

Plan for Monolith or Microservice upfront. *No Hybrids*

- You don't want to maintain/create unnecessary services
- Use Caching Layers sparingly
- Consider Number of transfers that the data makes
- Don't cheat yourself with Go routines and channels

weave

Platform

Use a messaging/streaming services to manage data flow

- APIs slow things down
- Allows for multiple worker consumption
- Manage batch processing jobs
- Services like Kafka allow for replaying jobs

weave

Platform

Use a schema repository

- Standardize data types across services (microservice)
- Leverage Generation for all the repos
- Protects againsts stupidity
- (protobuf + go = ❤️)

weave

Extraction

Pick the Data type at Extration and stick to it

- Avoid non-typed data
- Optimize type for storage
- Can you leverage scraping?

weave

Extraction

Create Struct types for complex data structures

- Pointers make things lighter
- Memory doesn't matter until it does
- Leverage interfaces for complete picture

weave

Extraction

Fail Fast isn't always your friend

- Always pass errors up to stop workers/routines/sub-processes
- Leverage back-off and retry
- Track failed data

weave

Load

Find a good sql library

- SQLX
- SQLC
- Logic in SQL can help protect data

weave

Load

Find the datastore that fits your needs

- Do you need a database, data lake, data warehouse or something else?
- Do you need more than one?

weave

Load

Manage Your Migrations

- Letting migrations get out of hand is gonna kill your data store

weave

Load

Monitor your data change. It makes debugging easier.

- Distributed data systems increase complications

weave

Transform

Do not infuse business data model into your storage

- Optimize for Storage
- Pick a type and stick to it
- Make data discoverable

weave

Transform

Don't be afraid of generating code

```
type example{{ $Type }} struct {
{{- range .TypeDef.TypeMembers}}
  {{ .Name }} {{ .DataType }} {{ if (ne .JSONName "")}}`json:"{{ .JSONName }}"`{{ end }}
{{- end }}
}
func (ex *example{{ $Type }}) Convert(ctx context.Context) (record.Record, error) {
        var dst = record.{{ $Type }}{
    {{- range .Conversions }}
            {{ .ProtoMember }} : {{ if (eq .ConversionType "null.UUID")}}{{ ToPrivate .Cast }}{{ else }}ex.{{ .Cast }}{{end}},
    {{- end}}
            }
        if ex.ID == "" {
                return nil, werror.New("id must be supplied").SetCode(werror.CodeInvalidArgument)
        }
        return &dst, nil
}
```

Transform

Business Logic Here!

weave

Transform

Live Data insights

weave

In Summary

Tips and Tricks Go:

- Build a Smart Data Platform
- Know what Data you Extract
- Load the right Data to the right place
- Transform Data carefully

weave

Conclusions

- There are lots of pitfalls when engineerng for data
- If you have to do it yourself do not!
- I do recommend doing using all the tools out there, but Go is the best!

weave

Contact

- Twitter @captainnobody1
- GitHub soypete
- LinkedIn

weave

REFERENCE:

- https://blog.gopheracademy.com/advent-2018/apache-beam/

weave