Before adopting rust:
Memory & gRPC

Bochun Bai
Vancouver Rust Meetup @ 2020.11

Norman Maurer @normanmaurer · Oct 17

What book / resources you can recommend to re-learn modern c++ after spending the last years writing java and c ? I think I forgot everything about c++ 🙍‍♂️

💬 33   🔁 32   ♡ 177   ⬆️

Dr Peter Brett
@PeterTBBrett

Replying to @normanmaurer and @pati_gallardo

As a C++ standards committee member, I recommend 'The Rust Programming Language' by Steve Klabnik and Carol Nichols.

6:18 AM · Oct 19, 2020 · Twitter for Android

57 Retweets **16** Quote Tweets **246** Likes

💬   🔁   ♡   ⬆️
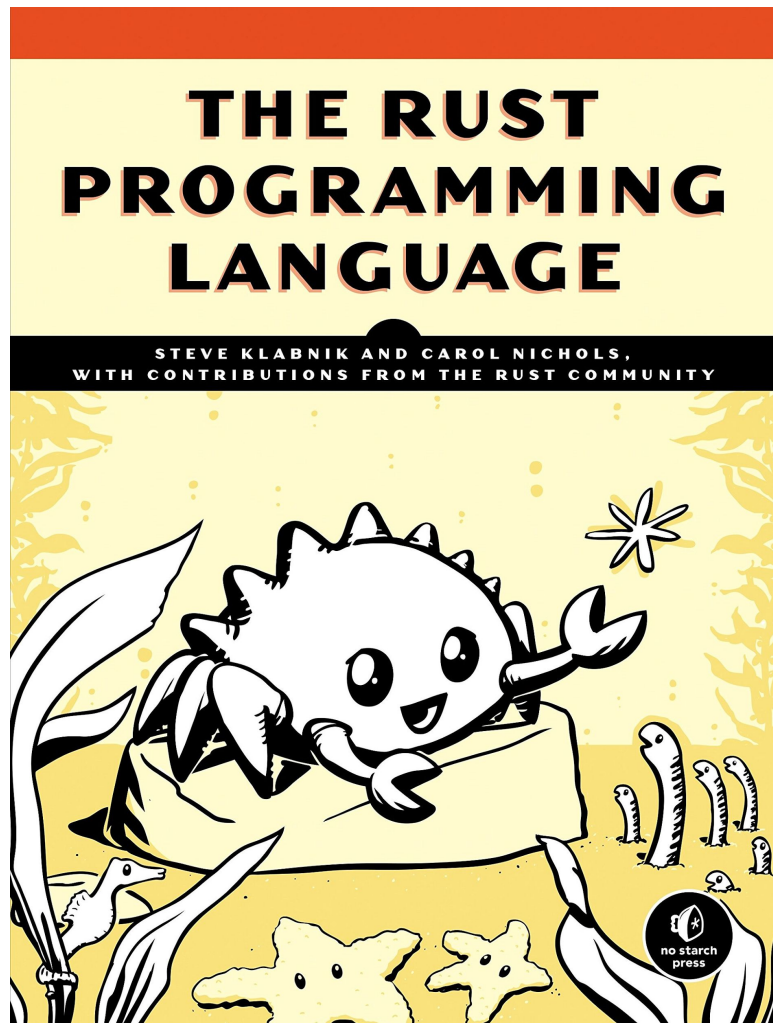
Dr Peter Brett @PeterTBBrett · Oct 19
Replying to @PeterTBBrett @normanmaurer and @pati_gallardo
Since this tweet is getting undue attention:

P.S. It's a joke

💬   🔁   ♡ 24   ⬆️

THE RUST
PROGRAMMING
LANGUAGE

STEVE KLABNIK AND CAROL NICHOLS,
WITH CONTRIBUTIONS FROM THE RUST COMMUNITY

no starch
press

Overview

- Microservices migration in progress
  - .NET → Java
  - SQL Server → MongoDB
  - Monolithic → Kubernetes
  - On Premises → Public Cloud
- Considerations:
  - Java memory
    - Golang
    - Rust
  - Inter services cost
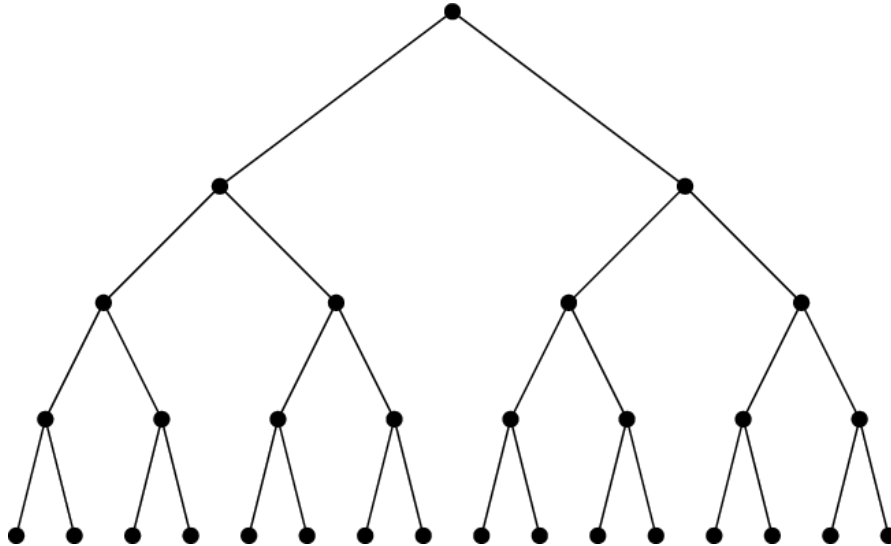   - External REST
    - Internal gRPC

Memory compare

- Java vs other languages
  - Memory usage
  - GC performance

gRPC compare

- Small requests
- Large requests
- Different gRPC libraries

Memory

Benchmarks game

https://benchmarksgame-team.pages.debian.net/benchmarksgame/
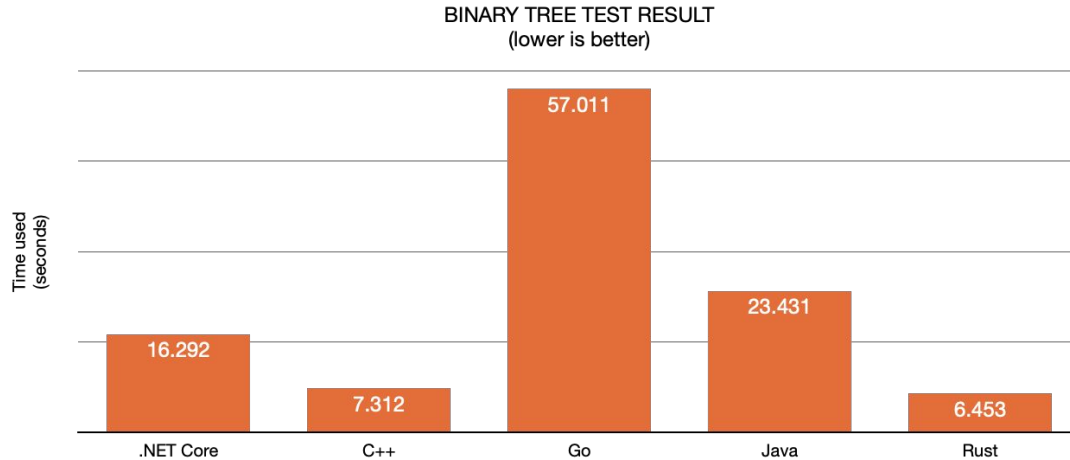
Binary trees

https://benchmarksgame-team.pages.debian.net/benchmarksgame/description/binarytrees.html#binarytrees

Code

https://github.com/sinofool/bench_gc

Memory compare result

BINARY TREE TEST RESULT
(lower is better)

Time used (seconds)

.NET Core 16.292	C++ 7.312	Go 57.011	Java 23.431	Rust 6.453

Create then destroy one 25 levels tree

Create one 24 levels long lived tree

Create then destroy short lived trees:

2^24 * 4 levels
2^22 * 6 levels
……
2^6 * 22 levels
2^4 * 24 levels

Destroy long lived tree

Memory compare conclusion

- development efforts vs runtime performance

- Java for rich 3rd party libraries

- .NET Core for legacy code

- Golang for heavy IO (e.g. API facade / message bridge)

- Rust for core performance

Rust gRPC compare

grpcio

https://github.com/tikv/grpc-rs

43 contributors / Latest release Jul 27 / Since 2017

tonic

https://github.com/hyperium/tonic

72 contributors / Latest release Aug 20 / Since 2019

grpc

https://github.com/stepancheg/grpc-rust

27 contributors / Latest release Sep 19 / Since 2015
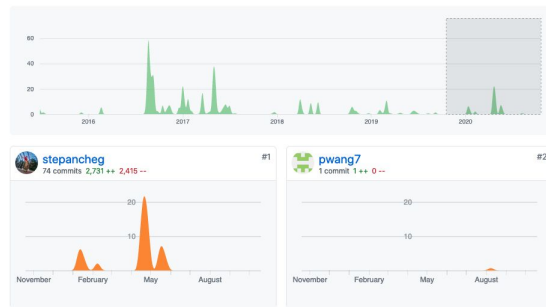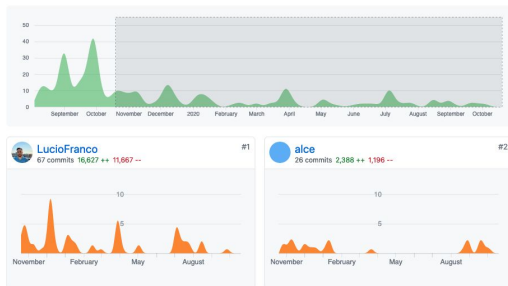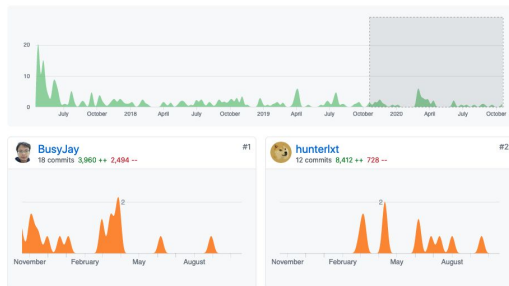
```rust
impl UserService for UserServiceImpl<'_>
```

grpcio

```rust
let env = Arc::new(Environment::new(64));
let channel = ChannelBuilder::new(env.clone());
let mut server = ServerBuilder::new(env)
    .register_service(svc)
    .bind("0.0.0.0", 50051)
    .channel_args(channel.build_args())
    .build().unwrap();
server.start();
```
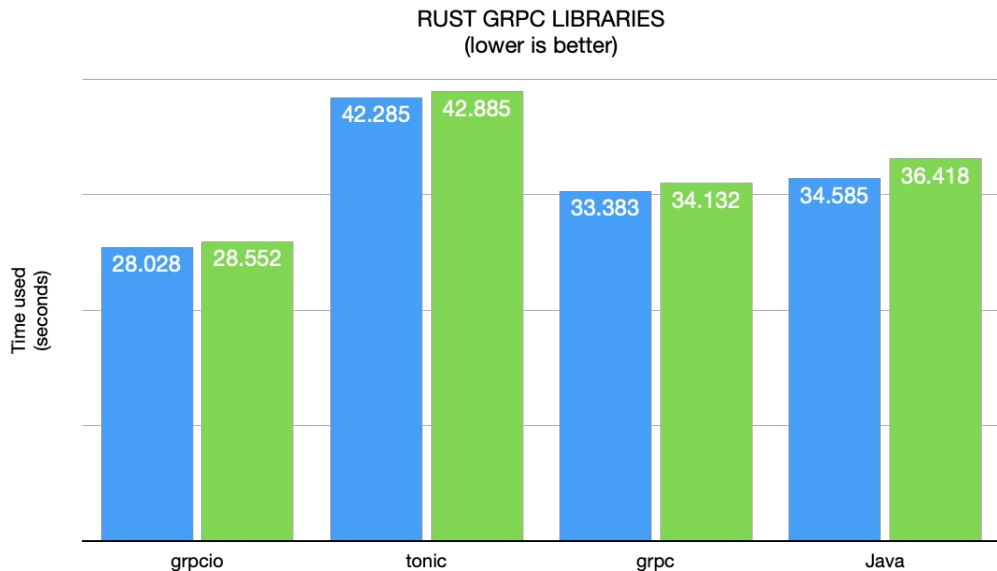
tonic

```rust
tonic::transport::Server::builder()
    .concurrency_limit_per_connection(256)
    .add_service(UserServiceServer::new(svc))
    .serve(addr)
    .await?;
```

grpc

```rust
let mut server = grpc::ServerBuilder::new_plain();
server.http.set_port(50051);
server.add_service(UserServiceServer::new_service_def(svc));
let _server = server.build().expect("server");
```

Rust gRPC result

RUST GRPC LIBRARIES
(lower is better)

Time used (seconds)

	grpcio	tonic	grpc	Java
Blue	28.028	42.285	33.383	34.585
Green	28.552	42.885	34.132	36.418

All libraries:
Easy of use

✔️ grpcio:
Wrapper of gRPC C core library

✔️ tonic:
More 3rd party dependencies (pros/cons)

⛔ grpc:
Less active community

Q & A

Bochun Bai

https://www.linkedin.com/in/bochun/
https://github.com/sinofool/
https://sinofool.net/