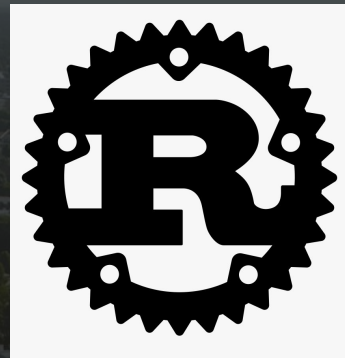


FPX Talks - Rust

Vad är rust och varför är det intressant



Andreas Franzén

- Systemutvecklare i ~10 år
- Webbutveckling och datahantering
- Generellt intresserad av programmering

Just nu:

- Dataplattform för FPX



Vad är Rust?

- Rust är ett systemprogrammerings språk.
- Statiskt typat, liknande C och C++ med delar från funktionella språk
- Plattformsberoende, Linux, Windows, macOS etc
- Använder LLVM som backend
- Kan kompilera till en massa system, processorer, webassembly

Vad är Rust?

- Prestanda
- Pålitlighet
- Produktivitet

Prestanda

- Liknande C/C++ i prestanda
- Ingen garbage collection
- Zero-cost abstraktioner

Pålitlighet

- Garanterar minnessäkerhet (inga null pointers, dangling pointers eller data races)
- Garanterar trådsäkerhet
- Automatisk minneshantering liknande RAII i C++

Produktivitet

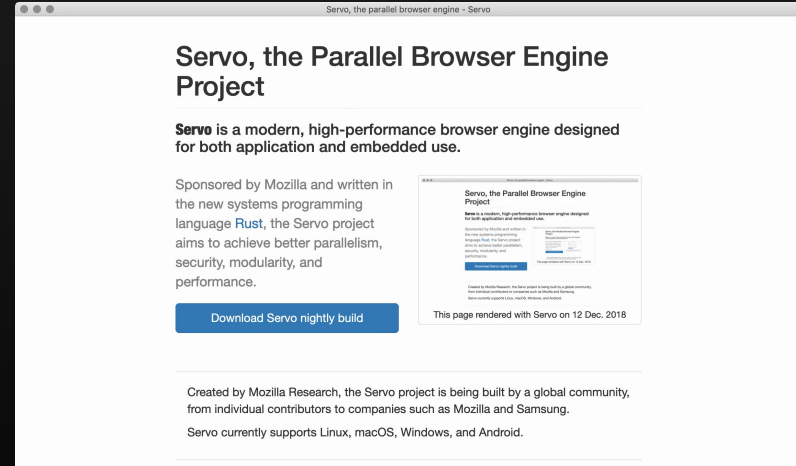
- Pakethanterare
- Dokumentation
- Felmeddelanden från kompilatorn
- IDE integrationer (rls och rust-analyzer för vscode och liknande, IntelliJ Rust)

Historia

- Graydon Hoare's sidoprojekt 2006
- Mozilla börjar sponsra 2009
- Rust 1.0 släpps 2015

Servo

- flertrådad rendering
- experimentell browser påbörjas 2012
- style komponenten integreras 2017
- renderings komponenten integreras 2019











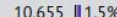
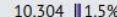
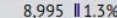


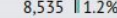
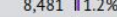
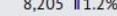
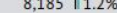


Projekt med Rust

- Amazon's firecracker, micro-vm som AWS Lambda och Fargate körs på
- Dropbox lagringsbackend
- npm använder det för cpu intensiva delar i sitt backend

Web Framework benchmarks

- actix, ett webb-ramverk leder eller tangerar högsta plats i benchmarks 2019

Best fortunes responses per second, Dell R440 Xeon Gold + 10 GbE (371 tests)												
Rnk	Framework	Best performance (higher is better)	Errors	Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA
1	actix-core	702,165 	0	Plt	Rus	Non	act	Lin	Pg	Lin	Raw	Rea
2	actix-pg	632,672 	0	Mcr	Rus	Non	act	Lin	Pg	Lin	Raw	Rea
3	h2o	456,058 	0	Plt	C	Non	Non	Lin	Pg	Lin	Raw	Rea
4	atreugo-prefork-quicktemplate	435,874 	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
5	vertx-postgres	403,232 	0	Plt	Jav	ver	Non	Lin	Pg	Lin	Raw	Rea
6	ulib-postgres	359,874 	0	Plt	C++	Non	ULI	Lin	Pg	Lin	Mcr	Rea
7	fasthttp-postgresql-prefork	352,914 	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
8	greenlightning	341,347 	0	Mcr	Jav	Non	Non	Lin	Pg	Lin	Raw	Rea
9	cpoll_cppsp-raw	326,149 	0	Plt	C++	Non	Non	Lin	My	Lin	Raw	Rea
10	fasthttp-postgresql	324,171 	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
304	flask	10,655 	0	Mcr	Py	Mei	Non	Lin	My	Lin	Ful	Rea
305	django	10,304 	0	Ful	Py	Non	Mei	Lin	My	Lin	Ful	Rea
306	flask-pypy2	8,995 	0	Mcr	Py	Non	Tor	Lin	My	Lin	Ful	Rea
307	rails	8,807 	0	Ful	Rby	Rac	Pum	Lin	My	Lin	Ful	Rea
308	tornado	8,778 	0	Plt	Py	Non	Tor	Lin	Mo	Lin	Raw	Rea
309	hapi-nginx	8,535 	0	Mcr	JS	Non	Non	Lin	Pg	Lin	Ful	Rea
310	rails-unicorn	8,481 	0	Ful	Rby	Rac	Uni	Lin	My	Lin	Ful	Rea
311	lumen	8,205 	0	Mcr	PHP	Non	ngx	Lin	My	Lin	Ful	Rea
312	express-postgres	8,185 	0	Mcr	JS	Non	Non	Lin	Pg	Lin	Ful	Rea

Ekosystem

- rustup, versionshanterare
- cargo, pakethanterare, allt i allo
- crates.io, paketregistret

Exempel 1

```
fn main() {  
    println!("Hello World!");  
}
```

Exempel 2

```
vim src/main.rs
use actix_web::{get, web, App, HttpServer, Responder};

#[get("/{name}")]
async fn hello(name: web::Path<String>) -> impl Responder {
    format!("Hello {}!", name)
}

#[actix_rt::main]
async fn main() -> std::io::Result<()> {
    HttpServer::new(|| App::new().service(hello))
        .bind("127.0.0.1:8080")?
        .run()
        .await
}
~
10,47 All
```

Varför är det intressant?

- Både lågnivå och högnivå språk
- Hängslen och livrem att prova lösningar som tidigare inte varit alternativ för mig

Frågor?
