Considering Rust

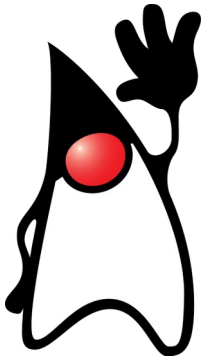
October 2021

Jesus Guzman, Jr.

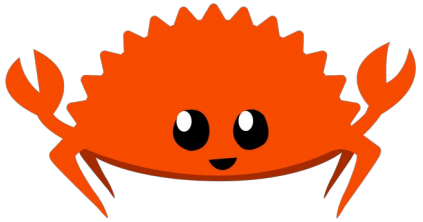
About me

Philosophy

- Humans make mistakes
- Rust catches mistakes as you type



Hey Java



Meet Rust

No Data Races

- *with Safe Rust
- Data Race != Race Condition
- Data Race + Race Condition

```
transfer1 (amount, account_from, account_to) {
  if (account_from.balance < amount) return NOPE;
  account_to.balance += amount;
  account_from.balance -= amount;
  return YEP;
}
```

Data Race and Race Condition

```
transfer1 (amount, account_from, account_to) {
  if (account_from.balance < amount) return NOPE;
  account_to.balance += amount;
  account_from.balance -= amount;
  return YEP;
}
```

NO Data Race; Race Condition

```
transfer2 (amount, account_from, account_to) {
  atomic {
    bal = account_from.balance;
  }
  if (bal < amount) return NOPE;
  atomic {
    account_to.balance += amount;
  }
  atomic {
    account_from.balance -= amount;
  }
  return YEP;
}
```

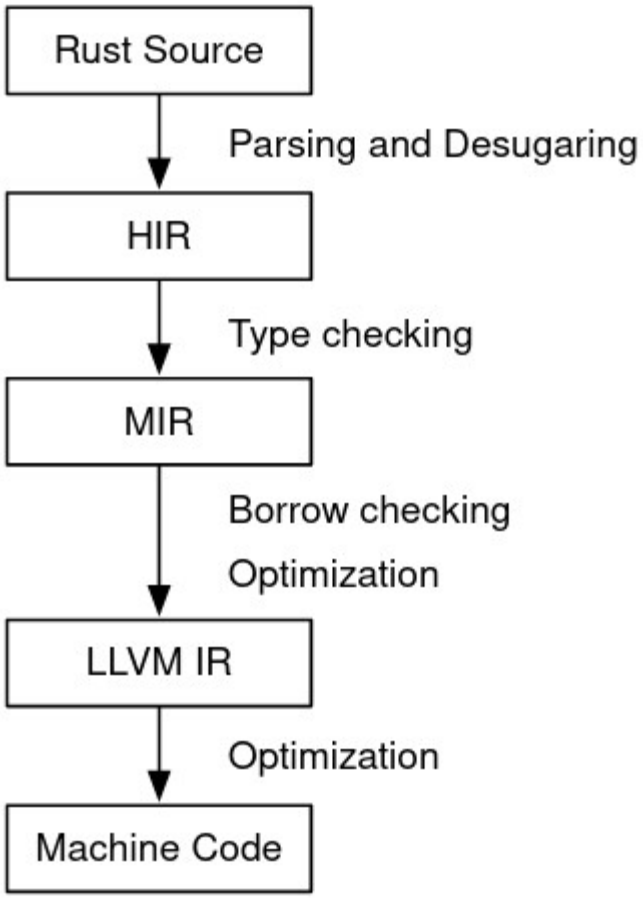
NO Data Race; NO Race Condition

```
transfer3 (amount, account_from, account_to) {
  atomic {
    if (account_from.balance < amount) return NOPE;
    account_to.balance += amount;
    account_from.balance -= amount;
    return YEP;
  }
}
```

Compilation Differences

Compilation

- Java → bytecode
- JRE loads classes and interprets bytecode



home > jesus > Desktop > rust1.rs

```rust
pub struct Tortoise {
    pub adorableness: i32,
    pub is_earth_bender: bool,
}

fn main() {
    let toodl: Tortoise = Tortoise {
        adorableness: 15,
        is_earth_bender: true,
    };

    let galapagos: Tortoise = Tortoise {
        adorableness: 1072,
        is_earth_bender: false,
    };
}
```

home > jesus > Desktop > mir1.rs

```mir
fn main() -> () {
    let mut _0: ();                          // return place in scope 0 at src/main.rs:6:11: 6:11
    let _1: Tortoise;                        // in scope 0 at src/main.rs:8:13: 8:18
    scope 1 {
        debug toodl => _1;                   // in scope 1 at src/main.rs:8:13: 8:18
        let _2: Tortoise;                    // in scope 1 at src/main.rs:14:9: 14:18
        scope 2 {
            debug galapagos => _2;           // in scope 2 at src/main.rs:14:9: 14:18
        }
    }

    bb0: {
        StorageLive(_1);                     // scope 0 at src/main.rs:8:13: 8:18
        (_1.0: i32) = const 15_i32;          // scope 0 at src/main.rs:8:21: 11:10
        (_1.1: bool) = const true;           // scope 0 at src/main.rs:8:21: 11:10
        StorageLive(_2);                     // scope 1 at src/main.rs:14:9: 14:18
        (_2.0: i32) = const 1072_i32;        // scope 1 at src/main.rs:14:21: 17:6
        (_2.1: bool) = const false;          // scope 1 at src/main.rs:14:21: 17:6
        StorageDead(_2);                     // scope 1 at src/main.rs:18:1: 18:2
        StorageDead(_1);                     // scope 0 at src/main.rs:18:1: 18:2
        return;                              // scope 0 at src/main.rs:18:2: 18:2
    }
}
```

home > jesus > Desktop > rust2.rs

```rust
pub struct Tortoise {
    pub adorableness: i32,
    pub is_earth_bender: bool,
}

fn main() {
    // introduce new scope
    {
        let toodl = Tortoise {
            adorableness: 15,
            is_earth_bender: true,
        };
    } // scope ends; toodl vanishes from heap

    let galapagos = Tortoise {
        adorableness: 1072,
        is_earth_bender: false,
    };
}
```

home > jesus > Desktop > mir2.rs

```
fn main() -> () {
    let mut _0: ();                      // return place in scope 0 at src/main.rs:6:11: 6:11
    let _1: Tortoise;                    // in scope 0 at src/main.rs:8:13: 8:18
    let _2: Tortoise;                    // in scope 0 at src/main.rs:14:9: 14:18
    scope 1 {
        debug toodl => _1;               // in scope 1 at src/main.rs:8:13: 8:18
    }
    scope 2 {
        debug galapagos => _2;           // in scope 2 at src/main.rs:14:9: 14:18
    }

    bb0: {
        StorageLive(_1);                 // scope 0 at src/main.rs:8:13: 8:18
        (_1.0: i32) = const 15_i32;      // scope 0 at src/main.rs:8:21: 11:10
        (_1.1: bool) = const true;       // scope 0 at src/main.rs:8:21: 11:10
        StorageDead(_1);                 // scope 0 at src/main.rs:12:5: 12:6
        StorageLive(_2);                 // scope 0 at src/main.rs:14:9: 14:18
        (_2.0: i32) = const 1072_i32;    // scope 0 at src/main.rs:14:21: 17:6
        (_2.1: bool) = const false;      // scope 0 at src/main.rs:14:21: 17:6
        StorageDead(_2);                 // scope 0 at src/main.rs:18:1: 18:2
        return;                          // scope 0 at src/main.rs:18:2: 18:2
    }
}
```

Stack Comparison

Stack	Java	Rust
domain name	java.marzipan.club	rust.marzipan.club
health endpoint	http://java.marzipan.club/info	http://rust.marzipan.club/info
language	Java SE 11	Rust 1.55.0
compiler	javac 11.0.12	rustc 1.55.0 (c8dfcfe04 2021-09-06)
compilation target	java bytecode 55.0	stable-x86_64-unknown-linux-gnu
compiles to	bytecode	machine code
runtime	OpenJDK	Tokio
web framework	Apache Tomcat and Spring Boot	Actix Web
package manager	mvn	cargo
manifest file	pom.xml	cargo.toml
configuration file	application.properties	config.ron
compile and run	`mvn spring-boot:run`	`cargo run`
prod compile and run	*same as above*	`cargo run --release`
clean command	`mvn clean`	`cargo clean`
dependency tree	`mvn dependency:tree`	`cargo tree`

Source

How many requests can it handle?

- Deserialize HTTP request to internal type
- Validate fields
- Process fields (e.g. hash password)
- Send to Postgres
- Serialize to json

Load Testing

- `cargo run --release --`
  `--host http://{java,rust}.marzipan.club`
  `-t10m` (10 minutes)
  `-u80` (80 users/threads)
  `-r20` (spawn 20 users per second)

HETZNER

CPX11

€ 3.99
monthly
€ 0.007 / h

2 vCPU AMD
2 GB RAM
40 GB Disk space
20 TB Traffic
Locations

Results

- Spring Boot 5,631
- Actix Web 25,324

Recommended Talks

- Considering Rust by Jon Gjengset
  - https://www.youtube.com/watch?v=DnT-LUQgc7s
  - More technical
- Summer of Rust by Bryan Cantrill
  - https://www.youtube.com/watch?v=LjFM8vw3pbU
  - About the “feel” of using various languages

Demo Code

https://github.com/MarzipanClub/JavaRust