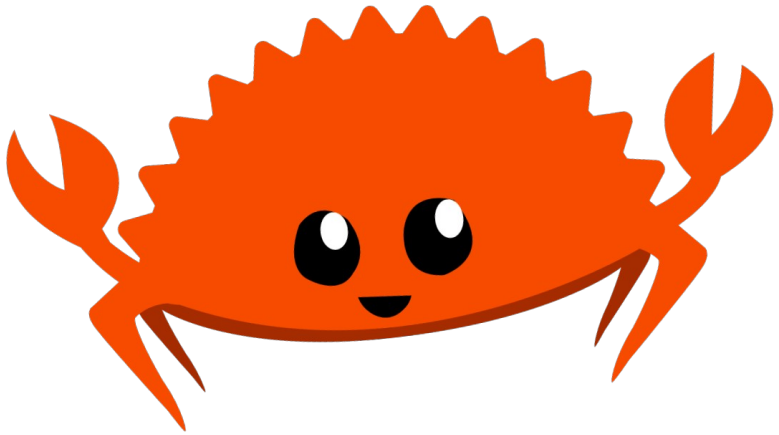
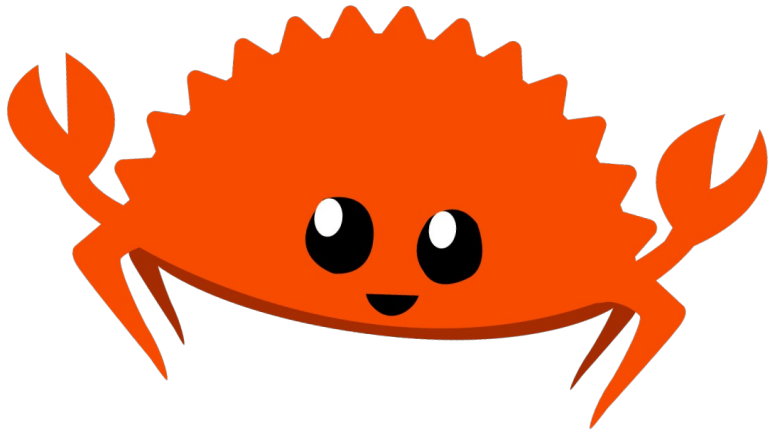
TOX-RS

+

Roman Proskuryakov at
ToxCon 2018 12. – 14.10.2018





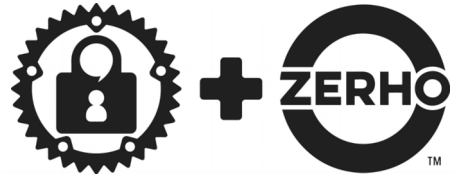
TOX-RS TEAM

Roman Proskuryakov, Bachelor in Computer Science, MEPhI, Russia

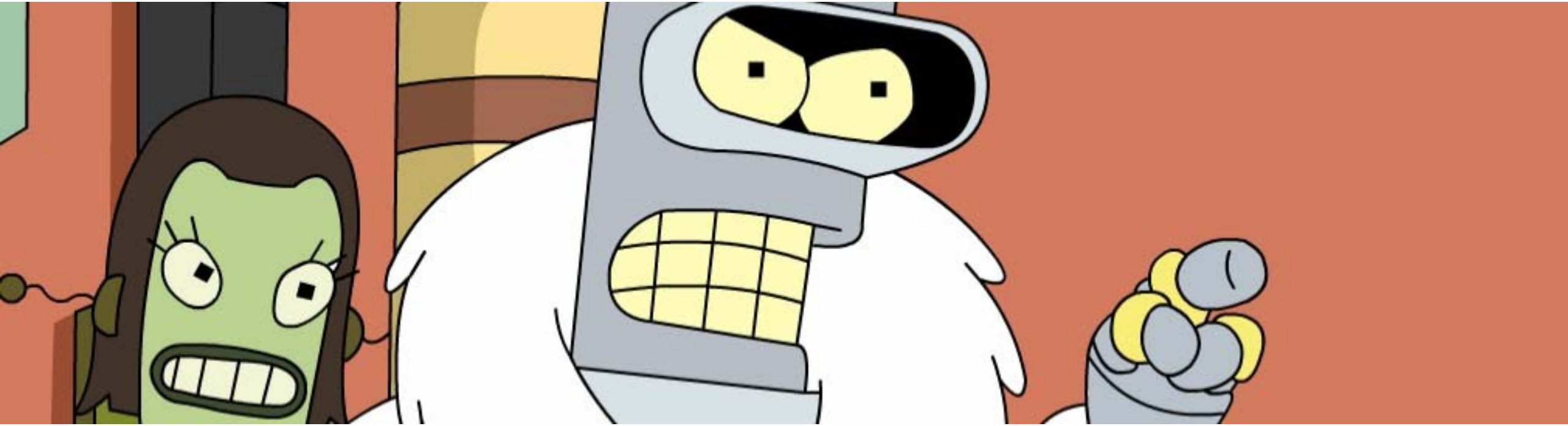
Evgeny Kurnevsky, Master in Computer Science, BSUIR, Belarus

Namsoo CHO, Master in Computer Science, Soongsil University, South Korea

Working for Zerho LLC



REWRITE TOX IN RUST



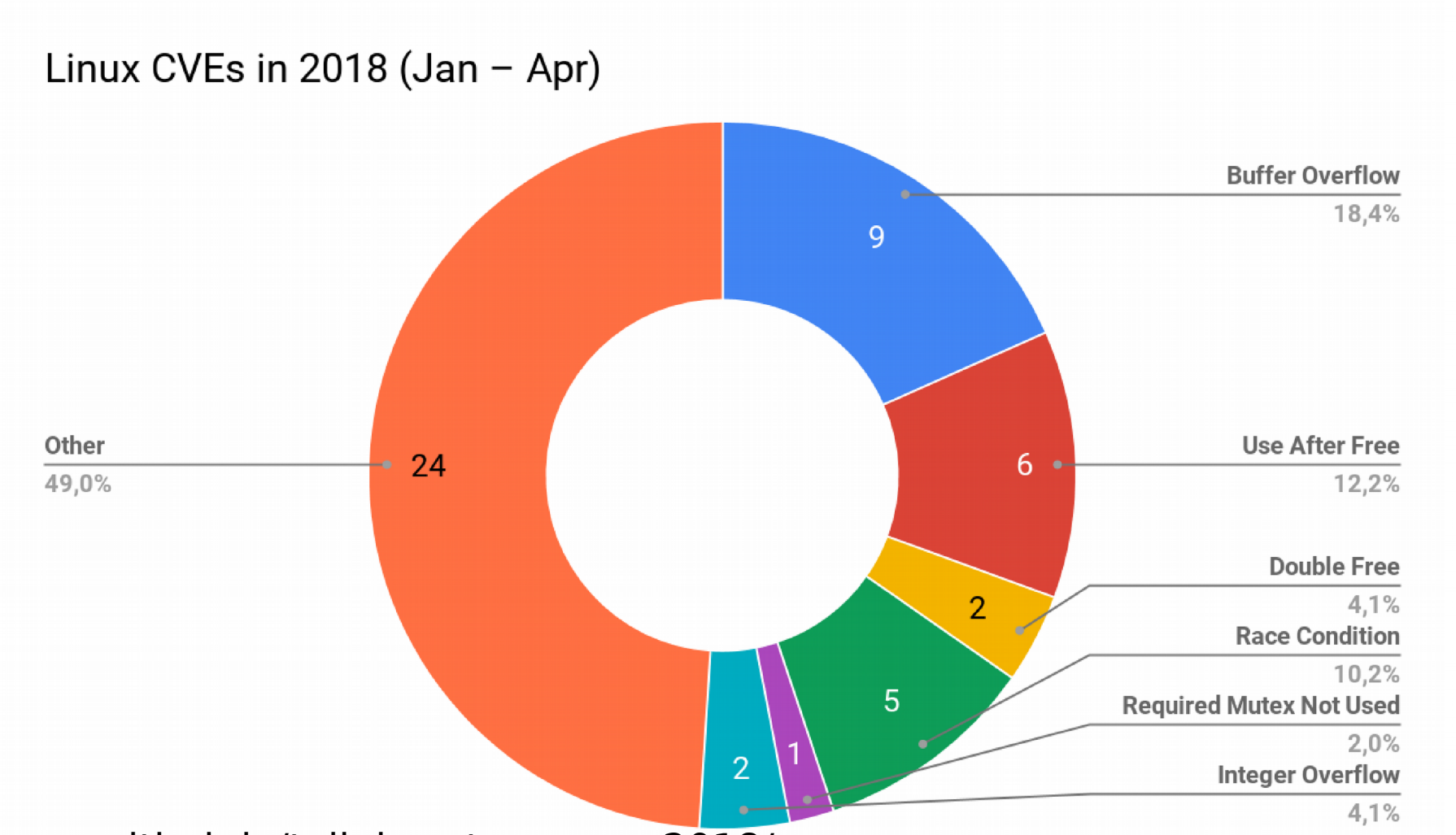
C++

With Rust you mostly wont need them:

- Valgrind
- asan
- tsan
- ubsan

PROGRAMMING IS HARD

LINUX GOT BUGS TOO!

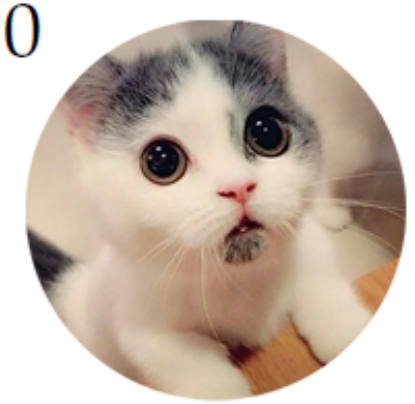


https://phil-opp.github.io/talk-konstanz-may-2018/

EVERYBODY LOVES CATS!



$$i\hbar\frac{\partial\psi}{\partial t} + \frac{\hbar^2}{2m}\nabla^2\psi - V\psi = 0$$

MATH

By the end of September 2018 RustBelt has proved:

- The type system, ownership and lifetime are correct
- The program is safe if all unsafe parts are correct
- Arc, Rc, Cell, RefCell, Mutex, RwLock are safe

As a byproduct found two bugs in:

- MutexGuard
- Arc

http://plv.mpi-sws.org/rustbelt/

BUGS IN UNSAFE RUST

```
1  #[allow(unsafe_code)]
2  fn sockaddr_to_ipaddr(sockaddr: *const posix_sockaddr) -> Option<IpAddr> {
3      let sa_family = u32::from(unsafe { *sockaddr }.sa_family);
4      if sockaddr.is_null() {
5          return None;
6      }
7      ...
8  }
```

https://github.com/maidsafe/get_if_addrs/pull/33

PROBLEMS: ABSTRACTIONS

```
1       switch (data[0]) {
2           ...
3           case TCP_PACKET_OOB_SEND: {
4               ...
5               return handle_TCP_oob_send(tcp_server, con_id, data + 1,
6                                          data + 1 + CRYPTO_PUBLIC_KEY_SIZE,
7                                          length - (1 + CRYPTO_PUBLIC_KEY_SIZE));
8           }
9       }
10  }
11  handle_TCP_oob_send(tcp_server, con_id, public_key, data, length) {
12      ...
13      resp_packet[0] = TCP_PACKET_OOB_RECV;
14      memcpy(resp_packet + 1, con->public_key, CRYPTO_PUBLIC_KEY_SIZE);
15      memcpy(resp_packet + 1 + CRYPTO_PUBLIC_KEY_SIZE, data, length);
16      write_packet_TCP_secure_connection(&tcp_server->accepted_connection_array[other_index],
17                                         resp_packet,
18                                         SIZEOF_VLA(resp_packet), 0);
19      ...
```

PROBLEMS: HARD TO READ

```
 The base nonce is the one TCP client wants the TCP server to use to encrypt the
- packets sent to the TCP client.
+ packets received from the TCP client.

 contains a base nonce which will be used later for
- encrypting packets sent to the TCP client.
+ decrypting packets received from the TCP client.

 The base nonce is the one the TCP server wants the TCP client to use to encrypt
- the packets sent to the TCP server.
+ the packets received from the TCP server.
```

https://github.com/TokTok/spec/pull/63

PROBLEMS: CONTRACTS

```
1   for (uint32_t i = 0; i < DHT_FAKE_FRIEND_NUMBER; ++i) {
2       uint8_t random_key_bytes[CRYPTO_PUBLIC_KEY_SIZE];
3       random_bytes(random_key_bytes, sizeof(random_key_bytes));
4       ...
5   }
```

https://github.com/TokTok/c-toxcore/issues/1169

PROBLEMS: VULNERABILITY

```
1    static int handle_recv_1(...)
2    {
3        Onion *onion = (Onion *)object;
4
5        if (length > ONION_MAX_PACKET_SIZE) {
6            return 1;
7        }
8
9        if (length <= 1 + RETURN_1) {
10           return 1;
11       }
12   +   if (packet[1 + RETURN_1] != NET_PACKET_ANNOUNCE_RESPONSE &&
13   +           packet[1 + RETURN_1] != NET_PACKET_ONION_DATA_RESPONSE) {
14   +       return 1;
15   +   }
16
```

CIA OR KGB?

THE RED DOWNTIME

https://nodes.tox.chat 120% Search

85.172.30.117	-	33445	8E7D0B859922EF569298B4D261A8CCB5FEA14FB91ED412A7603A585A25698832	ray65536	OFFLINE
node.tox.biribiri.org	-	33445	F404ABAA1C99A9D37D61AB54898F56793E1DEF8BD46B1038B9D822E8460FAB67	nurupo	OFFLINE
130.133.110.14	-	33445	461FA3776EF0FA655F1A05477DF1B3B614F7D6B124F7DB1DD4FE3C08B03B640F	Manolis	OFFLINE
205.185.116.116	-	33445	A179B09749AC826FF01F37A9613F6B57118AE014D4196A0E1105A98F93A54702	Busindre	OFFLINE
198.98.51.198	2605:6400:1:fed5:22:45af:ec10:f329	33445	1D5A5F2F5D6233058BF0259B09622FB40B482E4FA0931EB8FD3AB8E7BF7DAF6F	Busindre	OFFLINE
185.25.116.107	2a00:7a60:0:746b::3	33445	DA4E4ED4B697F2E9B000EEFE3A34B554ACD3F45F5C96EAEA2516DD7FF9AF7B43	MAH69K	OFFLINE
217.182.143.254	2001:41d0:302:1000::e111	2306	7AED21F94D82B05774F697B209628CD5A9AD17E0C073D9329076A4C28ED28147	pucetox	OFFLINE
tox.verdict.gg	-	33445	1C5293AEF2114717547B39DA8EA6F1E331E5E358B35F9B6B5F19317911C5F976	Deliran	OFFLINE
213.183.51.211	2a06:f901:1:100::98	33445	B71E91E2F5029B0A84D3B1136319CDD3D1DB6D3702B6CEFA66A4BEB25A635916	Skey	OFFLINE
toxnode.nek0.net	toxnode.nek0.net	33445	20965721D32CE50C3E837DD75B33908B33037E6225110BFF209277AEAF3F9639	Phsm	OFFLINE
163.172.136.118	2001:bc8:4400:2100::13:41d	33445	2C289F9F37C20D09DA83565588BF496FAB3764853FA38141817A72E3F18ACA0B	LittleVulpix	OFFLINE
78.46.73.141	2a01:4f8:120:4091::3	33445	02807CF4F8BB8FB390CC3794BDF1E8449E9A8392C5D3F2200019DA9F1E812E46	Sorunome	OFFLINE
37.97.185.116	-	33445	E59A0E71ADA20D35BD1B0957059D7EF7E7792B3D680AE25C6F4DBBA09114D165	Yani	OFFLINE
80.87.193.193	-	33445	B38255EE4B054924F6D79A5E6E5889EC94B6ADF6FE9906F97A3D01E3D083223A	linxon	OFFLINE
46.229.52.198	-	33445	813C8F4187833EF0655B10F7752141A352248462A567529A38B6BBF73E979307	Stranger	OFFLINE
tox.ngc.zone	tox.ngc.zone	33445	15E9C309CFCB79FDDF0EBA057DABB49FE15F3803B1BFF06536AE2E5BA5E4690E	Nolz	OFFLINE
149.56.140.5	-	33445	7E5668E0EE09E19F320AD47902419331FFEE147BB3606769CFBE921A2A2FD34C	velusip	OFFLINE
95.215.46.114	2a02:7aa0:1619::bdbd:17b8	33445	5823FB947FF24CF83DDFAC3F3BAA18F96EA2018B16CC08429CB97FA502F40C23	isotoxin	OFFLINE
37.48.122.22	2001:1af8:4700:a115:6::b	33445	1B5A8AB25FFFB66620A531C4646B47F0F32B74C547B30AF8BD8266CA50A3AB59	Pokemon	OFFLINE
tox.novg.net	-	33445	D527E5847F8330D628DAB1814F0A422F6DC9D0A300E6C357634EE2DA88C35463	blind_oracle	OFFLINE

WE DIDN'T KILL THEM ALL

TOX-RS in https://nodes.tox.chat/

33445	A04F5FE1D006871588C8EC163676458C1EC75B20B4A147433D271E1E85DAF839	kpp	ONLINE

TCP	Version	MOTD
33445	3000000004	Start date: Sat Sep 29 12:55:23 2018, uptime: 08 days 07 hours 03 minutes

33445	82EF82BA33445A1F91A7DB27189ECFC0C013E06E3DA71F588ED692BED625EC23	kurnevsky	ONLINE

USAGE

~/tox-node

```
:~/tox-node$ cargo run -- --help
    Finished dev [unoptimized + debuginfo] target(s) in 0.11s
     Running `target/debug/tox-node --help`
tox-node 0.0.4
Roman Proskuryakov <humbug@deeptown.org>
Evgeny Kurnevsky <kurnevsky@gmail.com>
Namsoo CHO <nscho66@gmail.com>
A server application to run tox node

USAGE:
    tox-node [FLAGS] [OPTIONS] --keys-file <keys-file> --tcp-address <tcp-address>... --udp-address <udp-address>

FLAGS:
    -h, --help       Prints help information
        --no-lan     Disable LAN discovery
    -V, --version    Prints version information

OPTIONS:
    -b, --bootstrap-node <public key> <address>    Node to perform initial bootstrap
    -k, --keys-file <keys-file>                    Path to the file where DHT keys are stored
    -l, --log-type <log-type>
            Where to write logs [default: Stderr]  [possible values: Stderr, Stdout, Syslog,
            None]
    -m, --motd <motd>
            Message of the day. Must be no longer than 256 bytes. May contain next variables placed in {{ }}:
            - start_date: time when the node was started
            - uptime: uptime in the format 'XX days XX hours XX minutes'
             [default: This is tox-rs]
    -t, --tcp-address <tcp-address>...             TCP address to run TCP relay
    -j, --threads <threads>
            Number of threads to use. The value 'auto' means that the number of threads will be determined automatically
            by the number of CPU cores [default: 1]
    -u, --udp-address <udp-address>                UDP address to run DHT node
        :~/tox-node$
```

FEATURES

- 100% compatible with C node
- DoS free
- Fearless concurrency
- No buffer overflows
- 96% code coverage
- 200 lines of io code

FUTURE

- Implement client part
- #[no_std]
- Optimizations
- WebSockets
- CAPI
- WebAssembly

LINKS

- https://github.com/tox-rs/tox
- https://github.com/tox-rs/tox-node
- https://www.parity.io/why-rust/
- https://blog.rust-lang.org/2015/04/10/Fearless-Concurrency.html
- http://plv.mpi-sws.org/rustbelt/