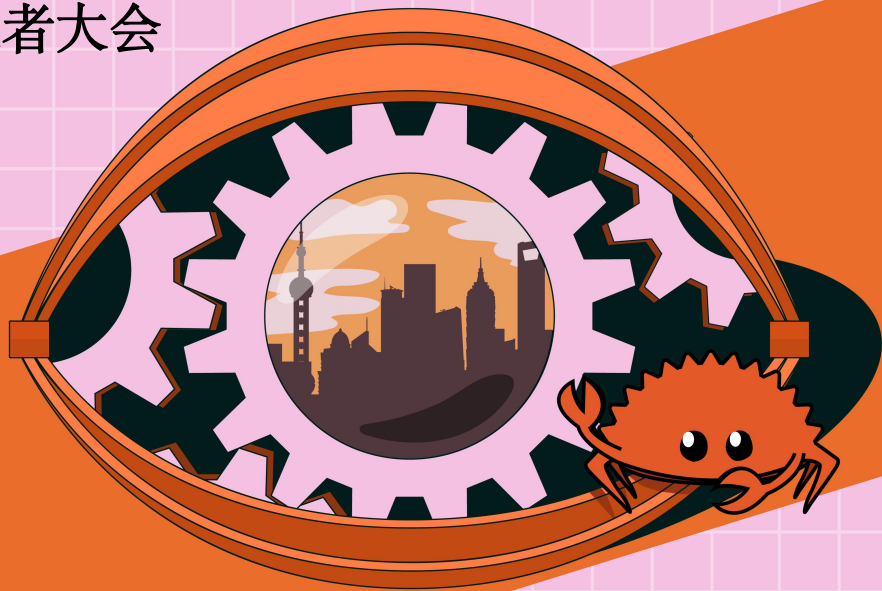


RUST CHINA CONF 2023

第三届中国Rust开发者大会



6.17-6.18 @Shanghai

Rust 在算法交易中的实际应用 与积极效应

非凸科技首席架构师 乔丹



公司介绍

上海非凸智能科技有限公司（简称“非凸科技”）成立于2018年，是国内领先的智能算法和交易系统服务公司，专注于智能算法交易领域的研究和开发。

公司正基于Rust生态，结合机器学习、深度学习等新兴技术，打造高效率、低延迟、高可靠、全内存高频交易平台，满足自身在量化行业交易业务的同时也为券商、量化私募等众多大型金融机构提供优质的算法交易解决方案。

100+

团队规模

80亿+

日均成交额

100+

合作机构



国家高新技术企业
科技型中小企业



证监会信息技术系统备案



双软企业资质认证



智能投研技术联盟核心成员

公司结构

50%

工程师

20%

策略研究

20%

运维中台

10%

市场商务

徐汇漕河泾 总部
浦东世纪汇 策略中心



上海

北京/成都



分公司 研发中心

自建超算中心



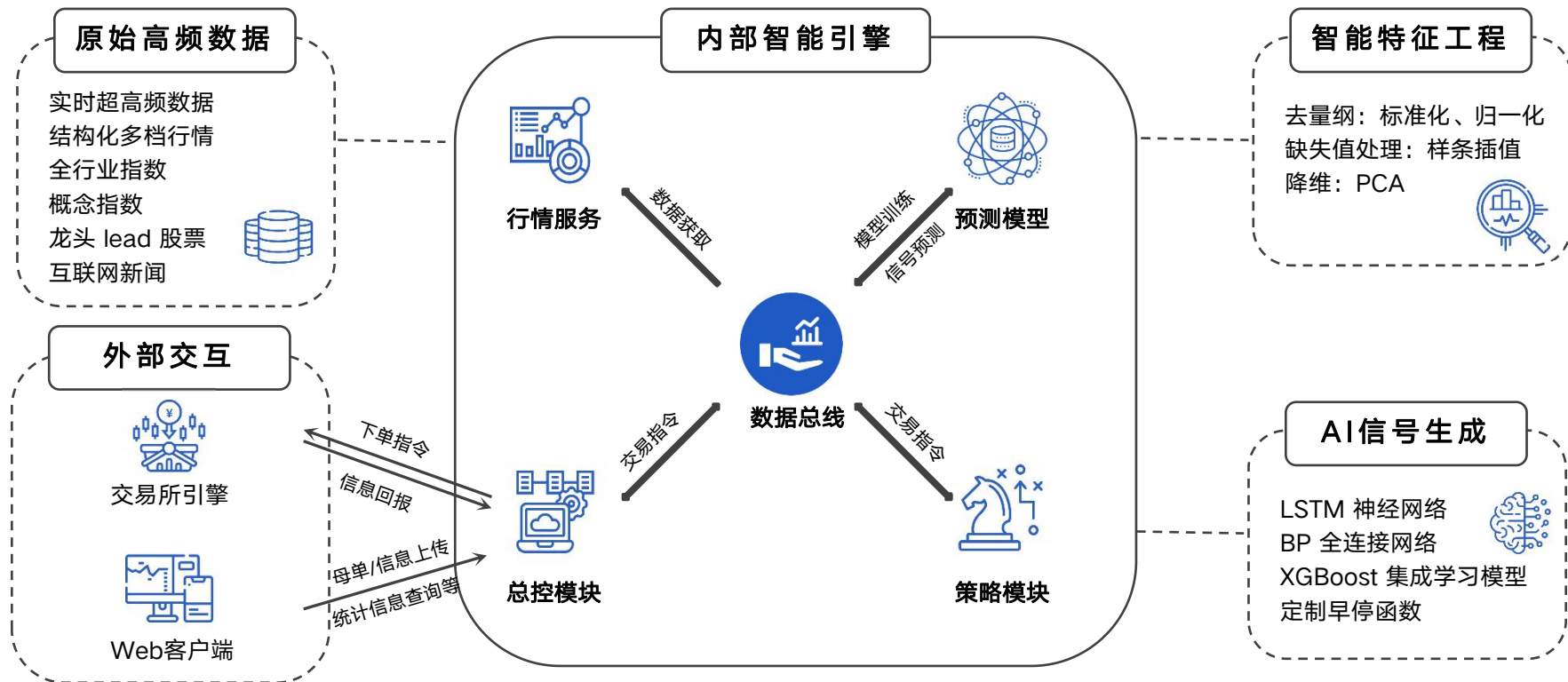
徐州

新加坡/美国



海外 研发中心

Rust 在非凸算法交易服务中的全栈应用实践

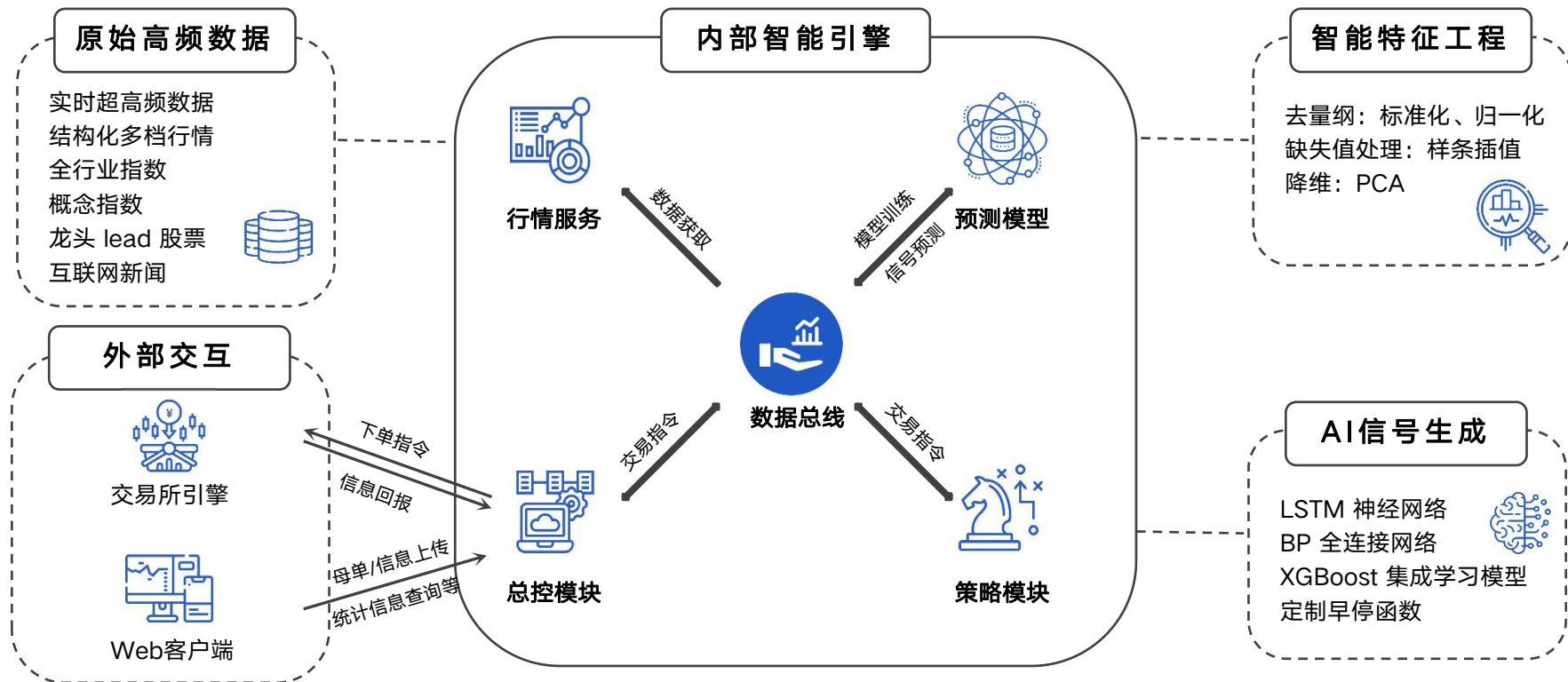


Rust 全栈应用实践 - 行情、与API接入

● Rust 进程间通信 XSHM

1. 低延迟高吞吐的一写多读消息队列
2. 基于共享内存，全用户态，零拷贝(配合 capnproto)
3. 针对 x86_64、AArch64 架构分别优化，CPU Cache 友好

Rust 在非凸算法交易服务中的全栈应用实践



Rust 全栈应用实践 - 行情、与API接入

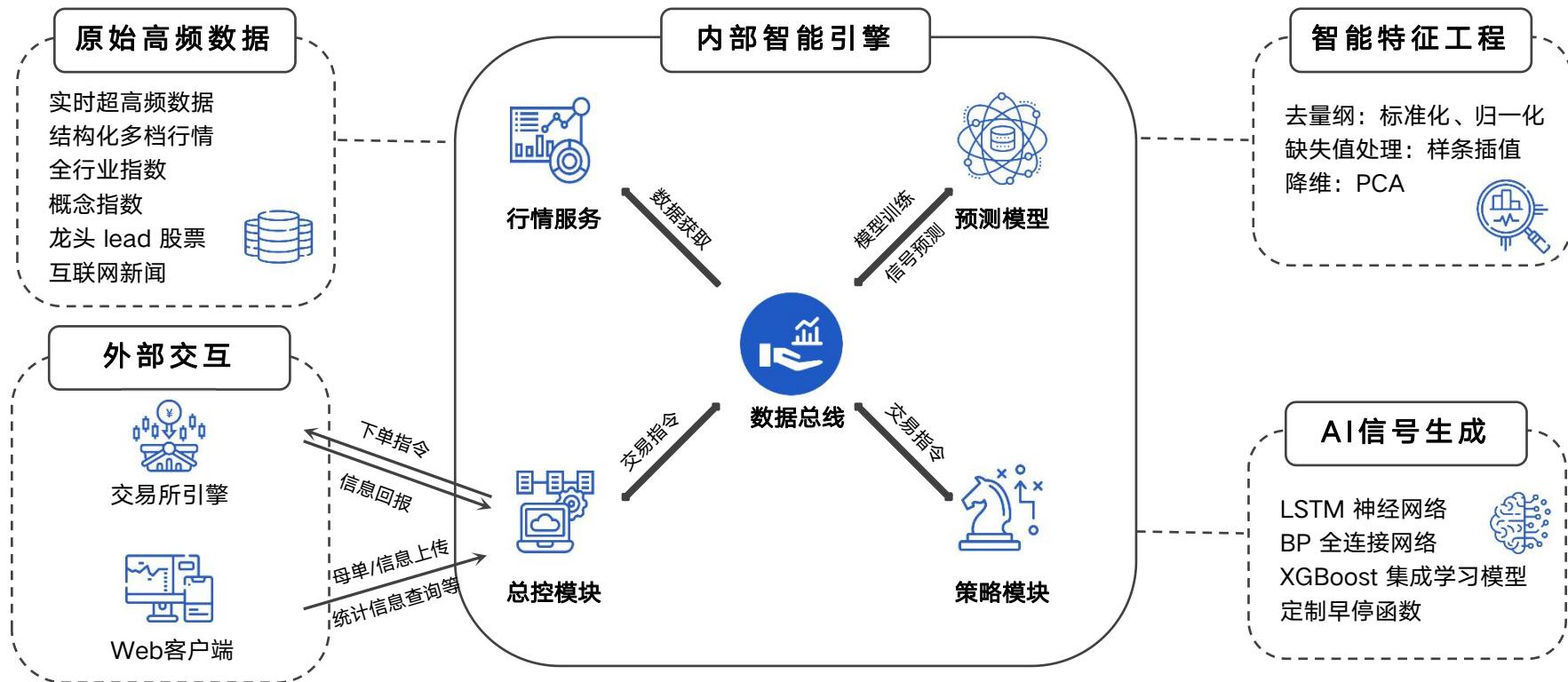
● Rust FFI

与非 Rust 世界交互的主要边界

● UDP 组播

网络协议栈参数优化，多路行情冗余互备

Rust 在非凸算法交易服务中的全栈应用实践



Rust 全栈应用实践 - 模型训练与在线预测

预测

森林模型的 Rust 展开

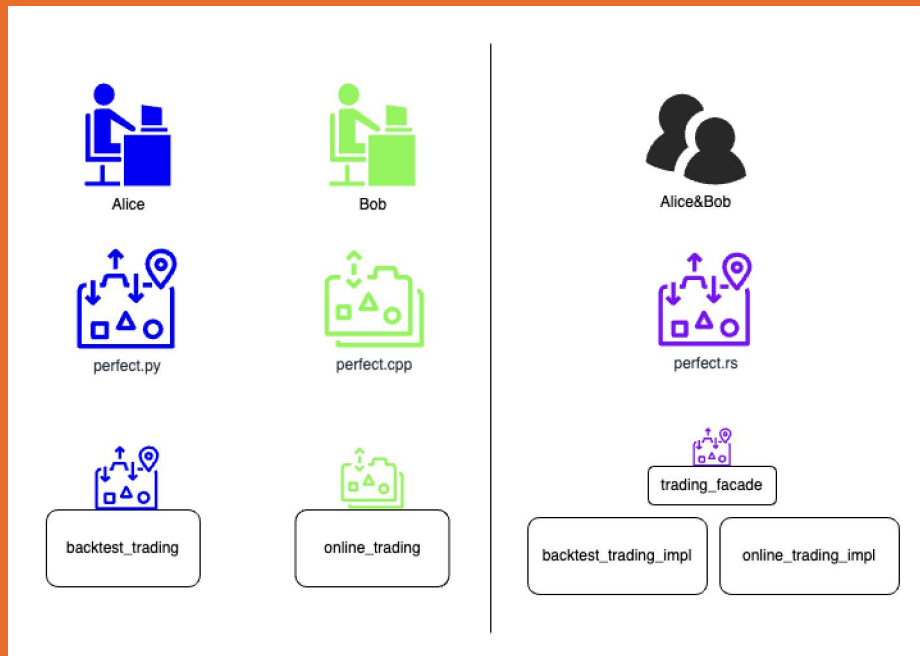
互联网新闻因子

大规模的新闻实时 spider 以及 NLP 分析

自建机房

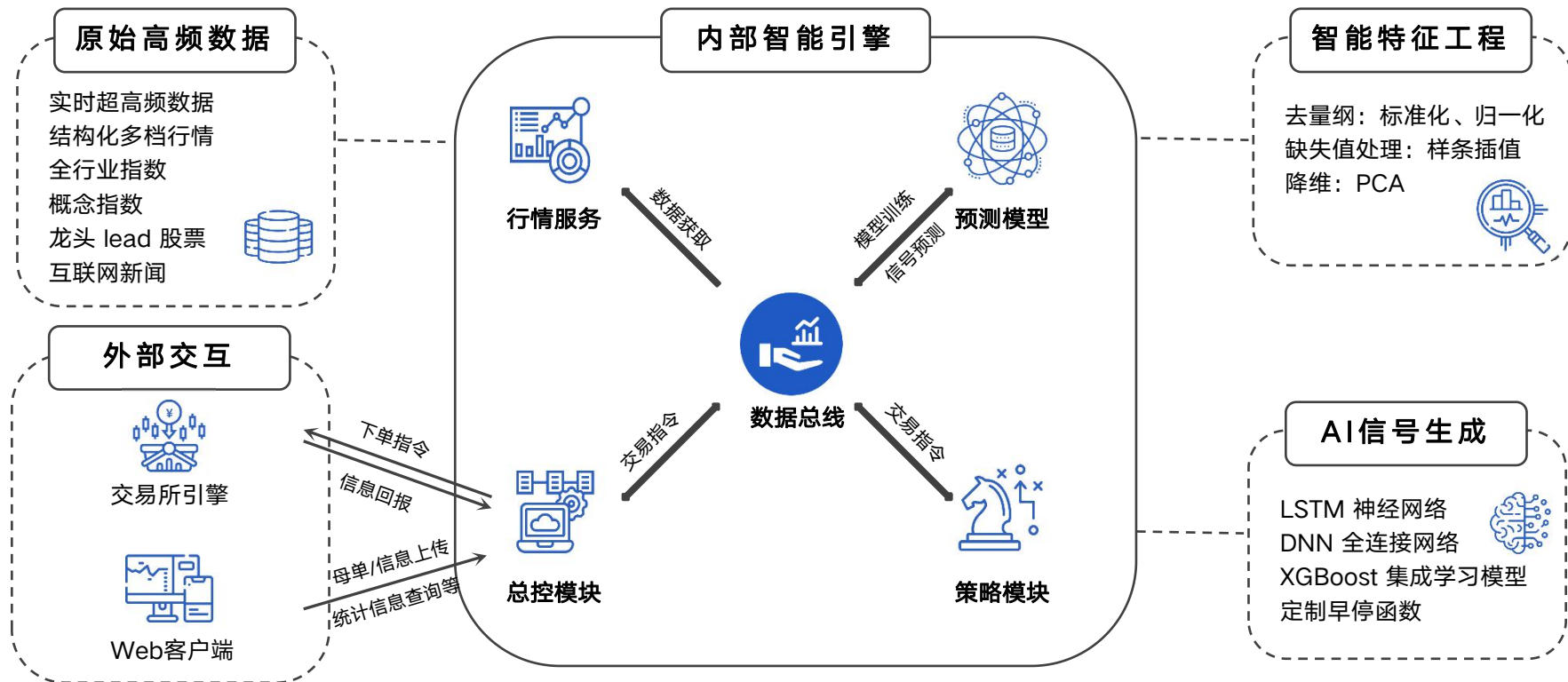
配备 8TB 内存 +8 显卡定制机型，分布式超大规模的 LSTM 等训练，单台价值 100w，支持全天候多周期预测

Rust 全栈应用实践 - 模型训练与在线预测



- “说同一种语言的一群人，将无所不能”
- 离线回测、在线模拟盘/实盘中策略代码统一
- 降低多栈维护成本
- 降低代码细节不一致导致的实验失真

Rust 在非凸算法交易服务中的全栈应用实践



Rust 全栈应用实践 - 桌面客户端、风控

● 涵盖异常的事前、事中、事后的全方位监控

● 风控服务端和桌面客户端

基于 Rust 异步编程框架开发，风控逻辑可插拔，兼顾部署灵活性与执行性能

Rust 全栈应用实践 - 开源

高性能异步日志库 ftlog

1. 特性丰富实用:

时间切分

受限写入

异步写入

按业务多目标写入

....

2. 性能优秀

Performance

Rust: 1.67.0-nightly

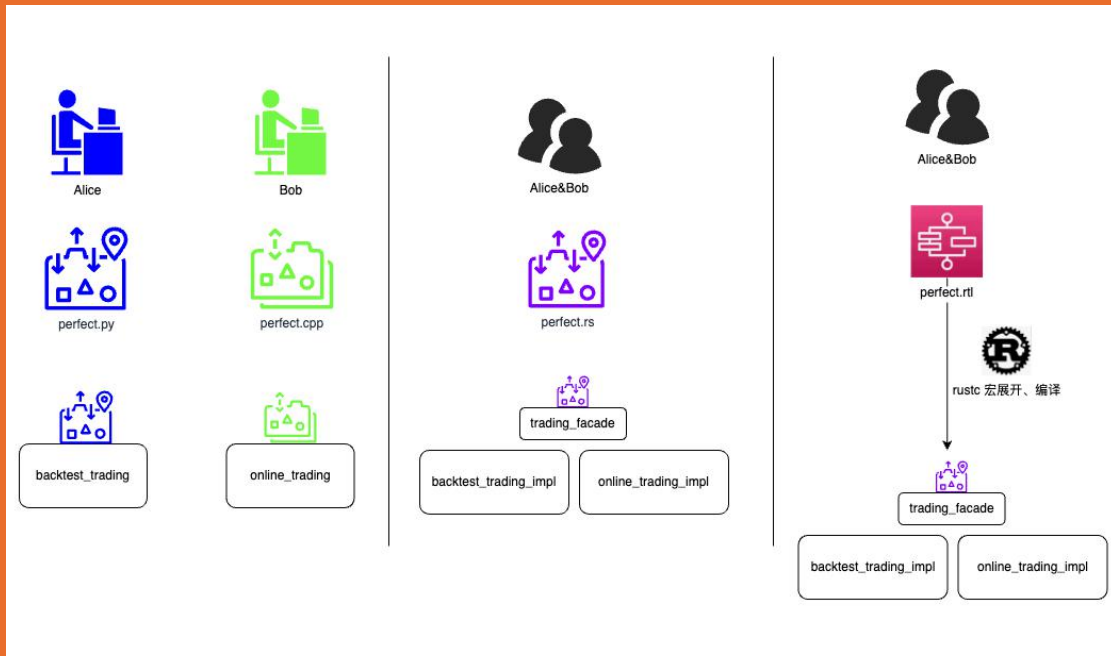
	message type	Apple M1 Pro, 3.2GHz	AMD EPYC 7T83, 3.2GHz
ftlog	static string	75 ns/iter	385 ns/iter
ftlog	with i32	106 ns/iter	491 ns/iter
env_logger output to file	static string	1,674 ns/iter	1,142 ns/iter
env_logger output to file	with i32	1,681 ns/iter	1,179 ns/iter
env_logger output to file with BufWriter	static string	279 ns/iter	550 ns/iter
env_logger output to file with BufWriter	with i32	278 ns/iter	565 ns/iter

License: MIT OR Apache-2.0

```
use ftlog::{
    appender::{Duration, FileAppender, Period},
    FtLogFormatter, LevelFilter,
};

let time_format = time::format_description::parse_owned::<1>(
    "[year]-[month]-[day] [hour]:[minute]:[second].[subsecond digits:6]",
)
.unwrap();
// configure logger
let logger = ftlog::builder()
    // global max log level
    .max_log_level(LevelFilter::Info)
    // custom timestamp format
    .time_format(time_format)
    // set global log formatter
    .format(FtLogFormatter)
    // use bounded channel to avoid large memory consumption when overwhelmed with logs
    // Set `false` to tell ftlog to discard excessive logs.
    // Set `true` to block log call to wait for log thread.
    // here is the default settings
    .bounded(100_000, false) // .unbounded()
    // define root appender, pass anything that is Write and Send
    // omit `Builder::root` will write to stderr
    .root(FileAppender::rotate_with_expire(
        "./current.log",
        Period::Day,
        Duration::days(7),
    ))
    // Do not convert to local timezone for timestamp, this does not affect worker thread,
    // but can boost log thread performance (higher throughput).
    .utc()
    // level filter for root appender
    .root_log_level(LevelFilter::Warn)
    // write logs in ftlog::appender to "./ftlog-appender.log" instead of "./current.log"
    .filter("ftlog::appender", "ftlog-appender", LevelFilter::Error)
    .appender("ftlog-appender", FileAppender::new("ftlog-appender.log"))
    .try_init()
    .expect("logger build or set failed");
```

Rust 应用场景展望 - RTL



离线回测、在线模拟盘/
实盘中策略代码统一，但
仍需研究员写 Rust 代码

Rust Trading Language

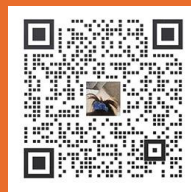
1. 基于 Rust 宏
2. 在编译时展开为 Rust 代码，
静态编译为 native 二进制
3. 进一步提升研究->回测->实盘
的迭代效率

Rust 应用场景展望 - 高性能可视化

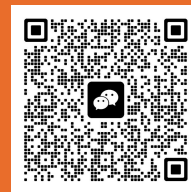
- 在浏览器内基于 Rust + WebAssembly 的高性能可视化工具
- 直接在 web 中直接看因子值
- 预测值和策略执行细节并即席调优



商务合作



技术交流



追求卓越算法 赋能量化交易



官网: ft.tech

电话: 021-5071 9392

地址: 上海市徐汇区漕河泾开发区 凯科国际大厦

Thank you!

