



GOTC 2023

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

OPEN SOURCE, INTO THE FUTURE

「大前端新趋势」专场

本期议题：Rspack 基于 Rust 的高性能 Web 构建工具

何庚坤 2023年05月28日



何庚坤

字节跳动前端工程师

Rspack 团队成员

GitHub @ahabhgk

1. Rspack 是什么？
2. Rspack 解决了什么问题？
3. Rspack 有哪些特性？
4. 如何从 Webpack 迁移到 Rspack？
5. Rspack 的架构设计
6. Rspack 的生态与未来

► Rspack 是什么?

Rspack 是一个基于 Rust 的高性能构建引擎，具备与 Webpack 生态系统的互操作性，可以从 Webpack 低成本的迁移，并提供更好的构建性能。



<https://www.rspack.dev>

◆ 开发环境性能

巨型项目中 Dev 启动时间 5-10 分钟，HMR 时间 10-20 秒，
甚至更久，开发体验极差

► Rspack 解决了什么问题?

◆ 生产环境性能

巨型项目中 **Build** 时间 **10-20** 分钟，甚至更久，拖慢 **CI/CD** 效率

► Rspack 解决了什么问题?



◆ 灵活丰富的配置

不止覆盖 **Web** 开发场景，能够应对各种场景

► Rspack 解决了什么问题?



◆ 强大的生产环境优化

强大可自定义配置的拆包能力等决定了产物的性能

► Rspack 解决了什么问题?

◆ ♂ 方便从原有方案（Webpack）迁移

用最小的成本优化巨型项目的开发体验

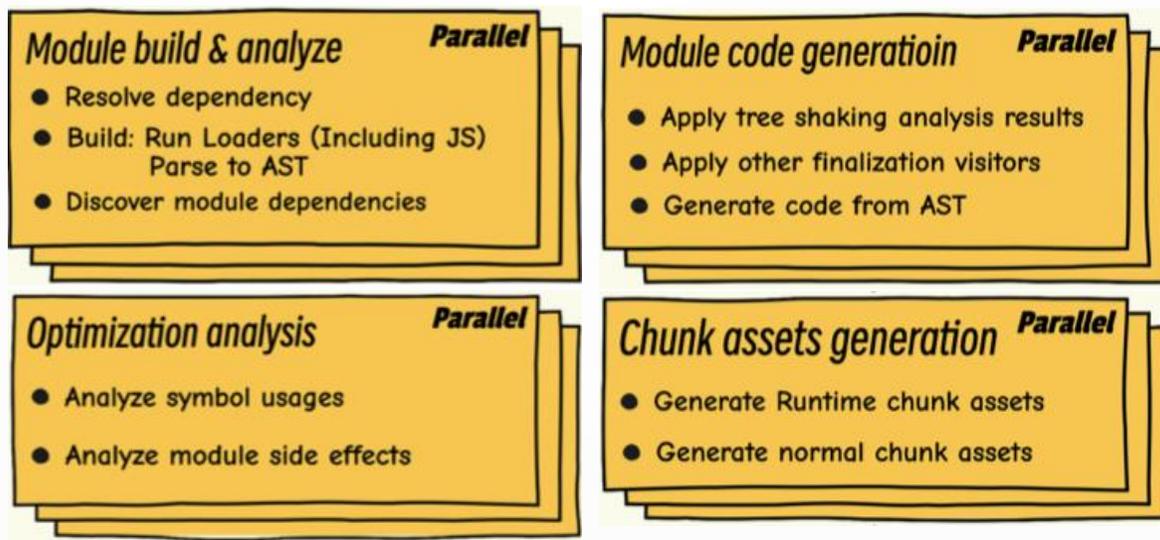
► Rspack 解决了什么问题?

◆ 社区现有方案往往能解决其中一两个问题，
但没有一个方案能解决所有问题

◆ 快!

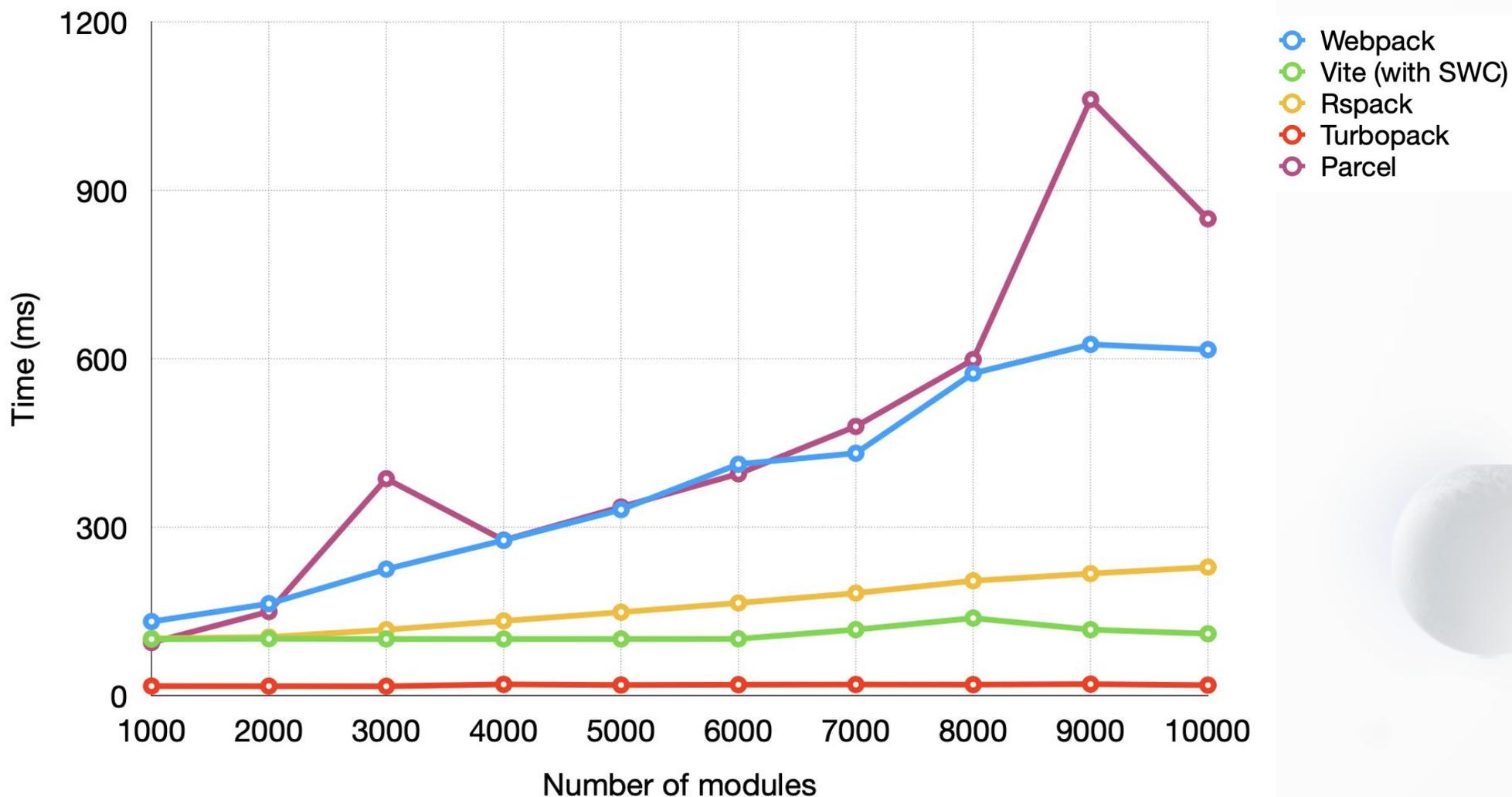
performance is the best feature!

◆ Rust 实现核心部分，充分利用并发和 LTO 优化



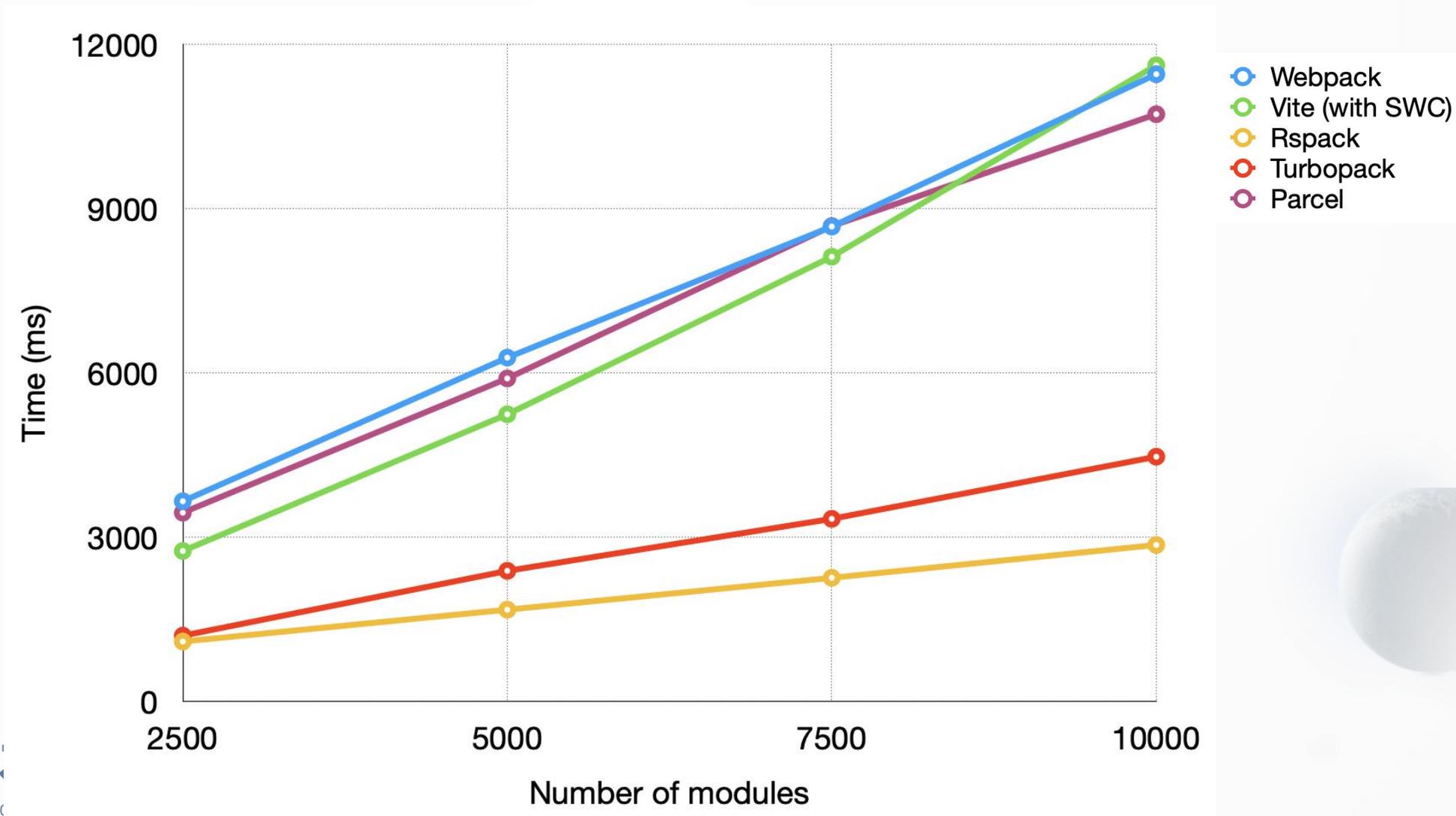
Rspack 有哪些特性?

◆ $O(\approx 1)$ 增量 HMR, 只编译修改的模块



► Rspack 有哪些特性?

⚡ Dev 启动耗时



一些业务收益

	Webpack	Rspack	提升
build	380s	28s	13 倍
dev	368s	25s	14 倍
hmr	21s	1s	21 倍

	Webpack	Rspack	提升
build	508s	59s	9 倍
dev	243s	24s	10 倍
hmr	18s	1.4s	13 倍

◆ 与 Webpack 部分兼容

The subset of Webpack!



常用配置兼容

```
/** @type {import('@rspack/cli').Configuration} */
const config = {
  entry: { main: "./src/index.tsx" },
  devServer: { port: 5555, websocketServer: "sockjs", historyApiFallback: true },
  mode: prod ? "production" : "development",
  devtool: false,
  cache: false,
  module: {
    rules: [
      { test: /\.module\.less$/, use: "less-loader", type: "css/module" },
      { test: /\.svg$/, use: "@svgr/webpack" },
      { test: /\.png$/, type: "asset" },
    ]
  },
  resolve: { alias: { "@": path.resolve(__dirname, "src") } },
  output: { publicPath: "/", filename: "[name].[contenthash].js" },
  optimization: {
    realContentHash: true,
    splitChunks: { cacheGroups: { someVendor: { chunks: "all", minChunks: 2 } } }
  },
  plugins: [new WebpackBundleAnalyzer()],
  builtins: { progress: true, html: { title: 'Arco Pro App' } },
};
```

◆ 常用 Loader 兼容

- babel-loader
- style-loader
- css-loader
- postcss-loader
- sass-loader
- less-loader
- raw-loader
- file-loader
- url-loader
- svelte-loader
- @mdx-js/loader
- @svgr/webpack
- image-webpack-loader
- thread-loader
- source-map-loader
- node-loader
- vue-loader (WIP, 预计 0.2 支持)
-

◆ 部分 Plugin 兼容/可替代

- `html-webpack-plugin` => `@rspack/plugin-html` or `builtins.html`
- `react-refresh-webpack-plugin` => `builtins.react.refresh`
- `webpack.DefinePlugin` => `builtins.define`
- `webpack.ProvidePlugin` => `builtins.provide`
- `mini-css-extract-plugin` => `experiments.css`
- `tsconfig-paths-webpack-plugin` => `resolve.tsconfigPath`
- `fork-ts-checker-webpack-plugin`
- `webpack-bundle-analyzer`
- `webpack-stats-plugin`
-

如何从 Webpack 迁移到 Rspack?

◆ 平滑!

得益于与 Webpack 良好的兼容性!

如何从 Webpack 迁移到 Rspack?

✓ 宽松模式

- 目前并不支持所有 webpack 配置，且有些配置会影响产物，为了保证产物的正确性，Rspack 默认开启了对配置的严格校验

```
# 开启宽松校验模式，会打印错误的配置但不会抛出错误
```

```
RSPACK_CONFIG_VALIDATE=loose rspack build
```

```
# 开启宽松校验模式，不打印错误也不抛出错误
```

```
RSPACK_CONFIG_VALIDATE=loose-silent rspack build
```

如何从 Webpack 迁移到 Rspack?

◆ Rust is faster than JavaScript

- 优先使用内置功能
- SWC > babel-loader, Rspack 使用 SWC 编译 JavaScript 代码
- experiments.css > style-loader + css-loader, Rspack 使用 SWC 实现了 experiments.css, 默认开启
- Asset Modules > file-loader + url-loader + raw-loader
- builtins.html > @rspack/plugin-html

如何从 Webpack 迁移到 Rspack?

more...

- <https://www.rspack.dev/zh/guide/migrate-from-webpack.html>
- <https://github.com/web-infra-dev/rspack-migration-showcase/pulls>

迁移

从 Webpack 迁移

Webpack 配置兼容性

从 Create React App 迁移

Storybook 迁移到 Rspack

web-infra-dev / rspack-migration-showcase

Code Issues 1 Pull requests 2

Labels Milestones New

Filters is:pr is:open

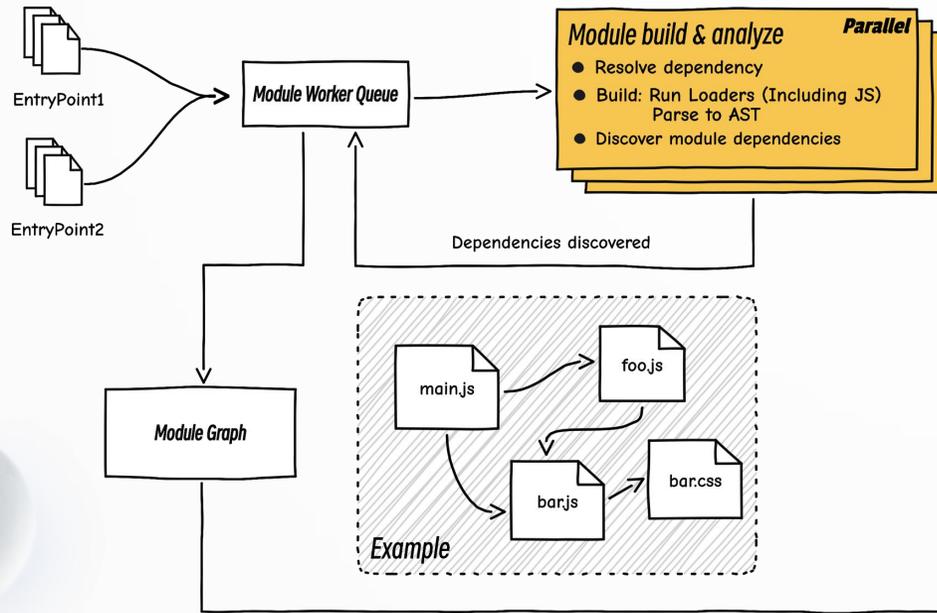
2 Open 5 Closed

Author	Label	Assignee	Sort
[WIP] feat: antd pro ✓			#8 opened 3 weeks ago by ycj1868
migrate ejected create-react-app to Rspack ✓			#6 opened on Feb 28 by ahabhgk

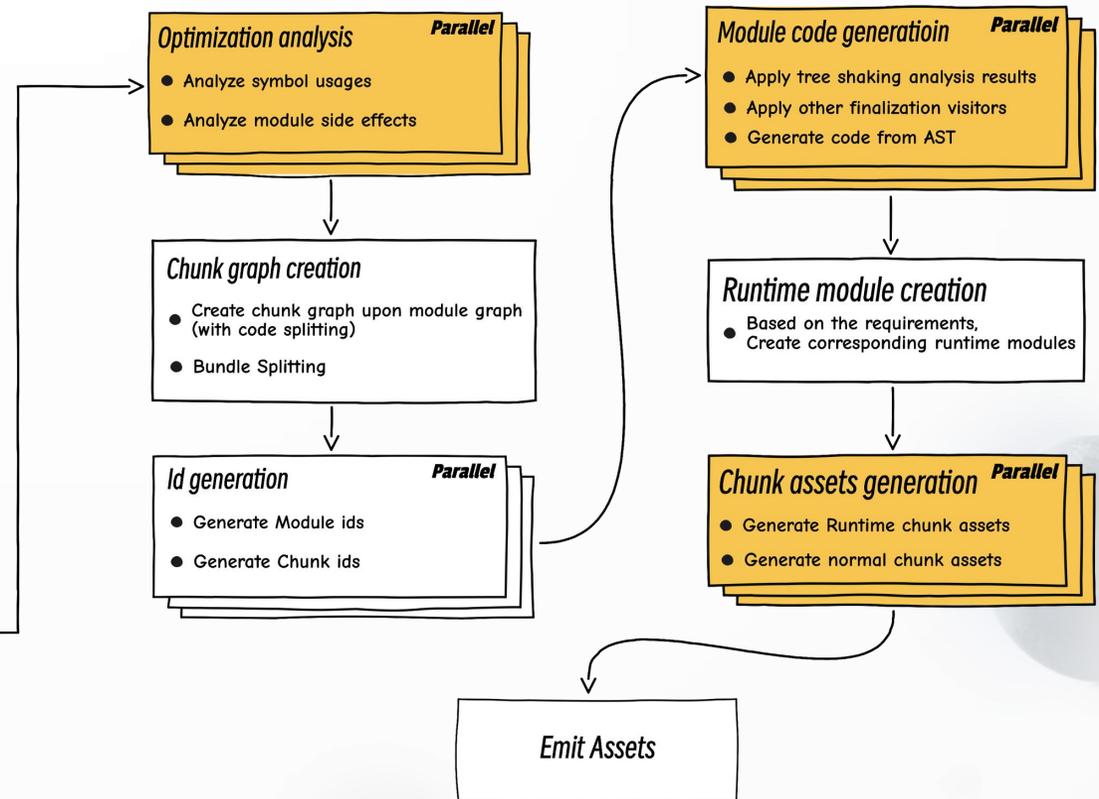
核心架构脱胎于 Webpack5

Parallel
= Running in parallel

Make Phase

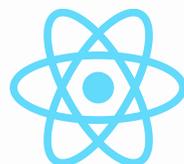


Seal Phase





Modern.js Doc



1. 支持更多常用框架：Vue、Qwik 等
2. 完善插件 API，兼容更多 Webpack 生态
3. 支持 Module Federation 模块联邦
4. 支持 Lazy Compilation 懒编译
5. 支持 Persistent Cache 持久缓存
6. 更多生产环境优化策略
7. 持续优化构建性能

◆ Rspack 的实现离不开以下项目



webpack





Rspack 交流群

ByteDance



扫描群二维码，立刻加入该群

该二维码永久有效

- 官方文档 (rspack.dev)
- GitHub ([web-infra-dev/rspack](https://github.com/web-infra-dev/rspack))
- 开发指南 (web-infra-dev.github.io/rspack-dev-guide)

THANKS