



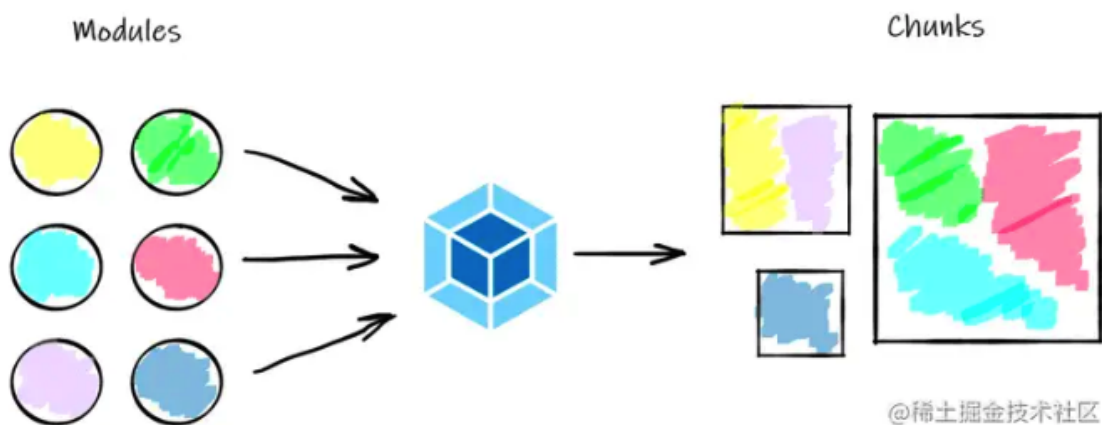
页面性能： 分包策略与 Chunk

单包解决的问题

- 页面初始代码包过大，影响首屏渲染
- 无法有效应用缓存

什么是 Chunk

此 webpack 特定术语在内部用于管理捆绑过程。输出束（bundle）由 chunk 组成，其中有几种类型（例如 entry 和 child）。通常，chunk 直接与 bundle 相对应，但是有些配置不会产生一对一的关系。



分包策略

Webpack 默认分包策略

- Initial Chunk: `entry` 模块及相应子模块打包成 Initial Chunk；

- Async Chunk: 通过 `import('./xx')` 等语句导入的异步模块及相应子模块组成的 Async Chunk;
- Runtime Chunk: 运行时代码抽离成 Runtime Chunk, 可通过 `entry.runtime` 配置项实现。(异步加载)

异步模块 Chunk

`require.ensure("./xx.js")` 或 `import("./xx.js")` 方式引入模块, 就可以实现模块的动态加载, 这种能力本质也是基于 Chunk 实现的。

Runtime 分包

Webpack 编译产物中还需要包含一些用于支持 webpack 模块化、异步加载等特性的支撑性代码, 这类代码在 webpack 中被统称为 `runtime`。

Chunks 优化

- `minSize`: 超过这个尺寸的 Chunk 才会正式被分包;
- `maxSize`: 超过这个尺寸的 Chunk 会尝试进一步拆分成更小的 Chunk;
- `maxAsyncSize`: 与 `maxSize` 功能类似, 但只对异步引入的模块生效;
- `maxInitialSize`: 与 `maxSize` 类似, 但只对 `entry` 配置的入口模块生效;
- `enforceSizeThreshold`: 超过这个尺寸的 Chunk 会被强制分包, 忽略上述其它 Size 限制。

service/config/splitChunk.js

```
const { BundleAnalyzerPlugin } = require("webpack-bundle-analyzer");
module.exports = (base) => {
```

```
// 入口优化
```

JavaScript

```

// runtime
base.entry = {
  index: { import: base.entry, runtime: "solid-runtime" }
  // import: base.entry,
  // runtime: "solid-runtime"
}

// 文件名
base.output.filename = "[name].[chunkhash:8].js"

base.plugins.push(new BundleAnalyzerPlugin({
  analyzerMode: "server",
  analyzerHost: "127.0.0.1",
  analyzerPort: 8889,
  reportFilename: "report.html",
  defaultSizes: "parsed",
  openAnalyzer: true,
  generateStatsFile: false,
  statsFilename: "stats.json",
  statsOptions: null,
  logLevel: "info",
})))

base.optimization = {
  splitChunks: {
    chunks: 'initial', // 从入口开始优化
    minSize: 200 * 1000, // 超过这个尺寸的 Chunk 才会正式被分包
    maxSize: 2000 * 1000, //超过这个尺寸的 Chunk 会尝试继续做分包
    maxAsyncSize: 2000 * 1000, //与 maxSize 功能类似，但只对异步引入的
    minRemainingSize: 2000 * 1000,
    minChunks: 1,
    maxAsyncRequests: 2,
    maxInitialRequests: 30,
    enforceSizeThreshold: 50000,
  },
}

return base;
};

```

URL	状态码	内容类型	资源名称	大小	加载时间	渲染时间	总时间
localhost	200	document	其他	544 B	572 毫秒		
17.3e838b2e.js	200	script	(索引)	168 kB	1.72 秒		
index.ae38e57d.js	200	script	(索引)	38.8 kB	999 毫秒		
11.04ce0efe.js	200	script	index.ae38e57d.js:1	22.9 kB	785 毫秒		
54.03b956ff.js	200	script	index.ae38e57d.js:1	16.2 kB	744 毫秒		
inject.js	200	script	content.js:57	1.3 MB	200 毫秒		
<input checked="" type="checkbox"/> normalize.css	200	stylesheet	875.js:2	1.8 kB	137 毫秒		
<input checked="" type="checkbox"/> blueprint.css	200	stylesheet	875.js:2	294 kB	137 毫秒		
<input checked="" type="checkbox"/> blueprint-select.css	200	stylesheet	875.js:2	2.9 kB	137 毫秒		
<input checked="" type="checkbox"/> cropper.css	200	stylesheet	875.js:2	3.7 kB	137 毫秒		
88ccd17efc924bf83304.png	200	png	17.3e838b2e.js:2	21.2 kB	923 毫秒		
d21a4b9edc288bd7c22f.png	200	png	17.3e838b2e.js:2	83.2 kB	1.62 秒		