

Server-Side JavaScript Development



Server-Side JavaScript Development

先驱



Aptana Jaxer (SpiderMonkey)



Helma (Rhino)

问题

- 缺乏统一标准
- 技术生态圈的问题





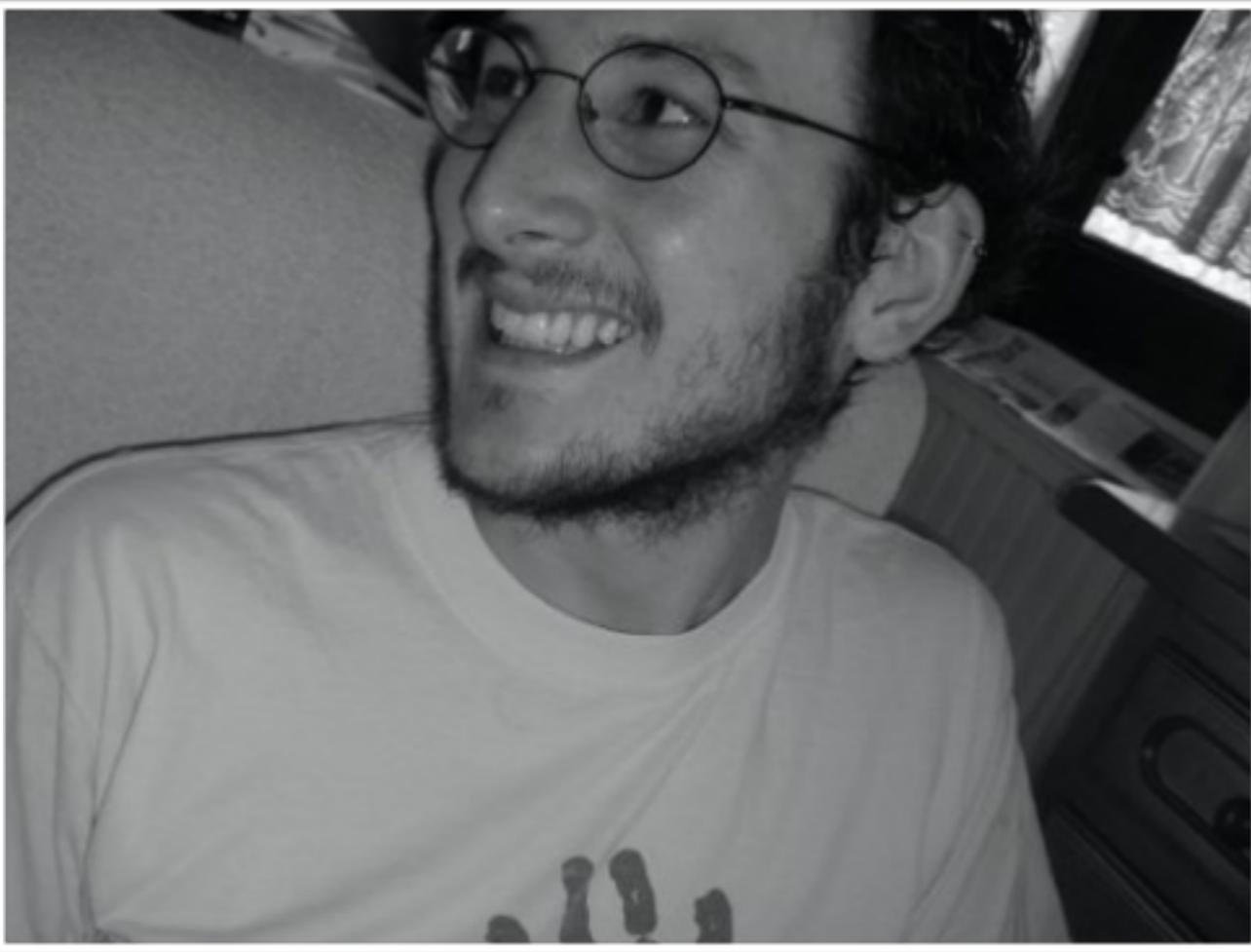
Revelutions in 2009

- CommonJS (ServerJS), by Dangoor
- JSConf, by Chris Williams & Iterative Designs
- Node, by Ray Dahl.



Node's goal is to provide an easy
way to build scalable network
programs

-- nodejs.org



Ray Dahl

发展历史

- 1/5/2009 Ryah Dahl 提出项目构想
- 2/15/2009 项目启动
- 5/31/2009 发布初始版本
-
- 11/16/2010 v0.3.1 发布

社区与生态圈

- 目前在官方wiki上有近600个module
- 超过20位活跃的开发者
- 每天都在进步

基础库

- libev (event loop)
- libeio (nonblocked posix, thread pool)
- v8 (javascript engine by google)

核心思想

- 非阻塞
- 单线程
- 事件驱动

传统模式下的io操作

```
var data = file.read("file.data");
// 等待io返回
doSomething(data);
```

程序逻辑需要等待io操作完成

非阻塞io操作

```
file.read("file.data", function(data){  
    doSomthingAfterRead(data);  
});  
doSomethingOnReading();
```

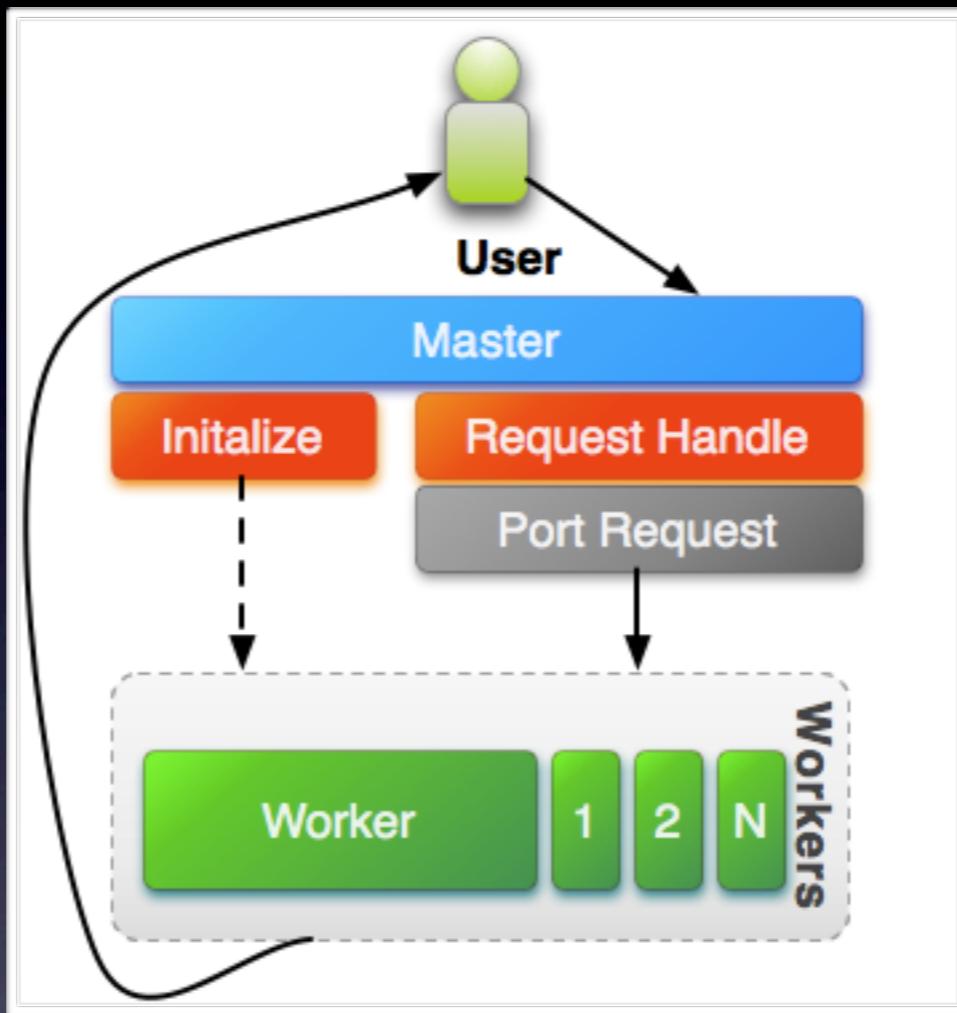
利用io操作的等待时间做其它工作

单线程 - 优势

- 程序逻辑简单
- 系统资源占用低
- 无通信与锁开销带来高性能

单线程 - 劣势

- 一个程序仅能在一个cpu上运行
- 一旦程序异常将导致整个进程崩溃



解决方案 - Web Worker

性能

- AMD Opteron 2200 单核, 4G RAM
- socket client 20,000+ qps
- socket server 17,000 qps
- http server 4,400 qps
- 内存消耗 30~40m
- see more. <<http://nodejs.kongwu.net/blog/?p=8>>

一些限制

- SSL支持不够成熟
- Windows
- IG 堆栈内存限制(by v8)

适用场景

- Web框架
- 大并发负载中间层服务
- Spider

