QCon

基于 Erlang/OTP 构建大规模实时系统

http://yunba.io

Yunba.io云巴

Brought by InfoQ

关于我

- 连续创业者
- Yunba.io 创始人，CEO
- JPush 创始人，原CTO
- @Tiger_张虎

议程

- Yunba.io 实时系统架构简介
- Erlang/OTP 对构建大并发系统的支持
- C/C++ 团队切换到 Erlang

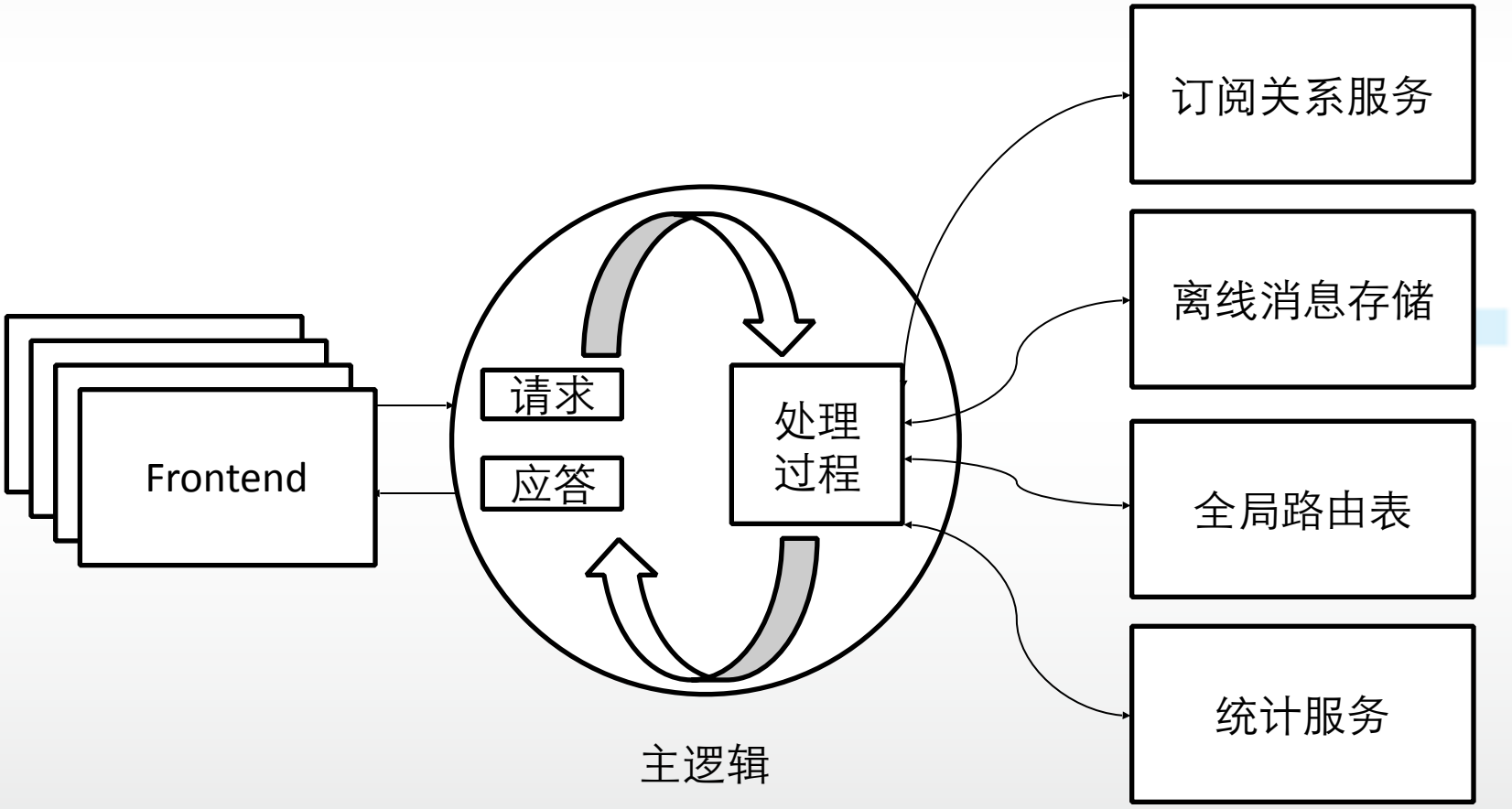
关于 Yunba.io

- 云巴： Cloud Bus
  - 云端实时系统
- MQTT as a service
  - 智能硬件、智能手机、PC、Mac
- Socket.io as a service
  - Web、Web App

Yunba.io 架构

- 支持亿级用户
- 在线弹性扩容
  - 云主机友好
- 部分机器宕机不影响业务

Yunba.io 架构



面临的问题

- 大并发
- 软实时
  - 秒内延迟
- 高可用
  - 单点失效
  - IDC失效
- 线性扩容

大并发

- 异步 IO
  - Libev/Nginx, node.js
- 轻量级进程
  - Erlang, Go, Akka

异步 IO 框架

- Libev/Nginx
  - 运行效率高
  - 开发成本非常高
  - 适用于业务逻辑简单，外部依赖少
- Node.js
  - 运行效率一般
  - 开发成本低
  - 承受不了很大并发

异步 IO 框架的问题

- CPU 时间分配公平性不能保障
  - 部分繁忙的任务造成其他任务饥饿
    - 损失软实时性
  - 同一个 process 运行，错误隔离很弱

Go/Akka

- Go/Akka
  - 与 Erlang 类似的轻量级进程模型
  - Akka 可以直接使用 Java 库
  - 调度公平性
  - 全局 GC
- 缺少类似 OTP 的框架

Erlang 特性

- 函数式语言
- 轻量级进程
  - 独立身份、独立状态
- 抢占式调度
  - Reduction
- 进程独立 GC
- 分布式支持

函数式语言

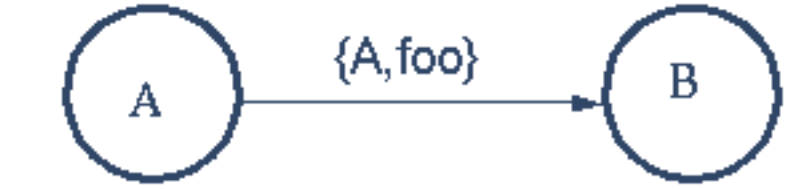
- 关注于逻辑描述
- 模式匹配(Pattern matching)
- 代码可读性好?

```
-module(fact).
-export([fac/1]).

fac(N) when N > 1 ->
    N * fac(N-1);
fac(1) ->
    1.
```

轻量级进程

- 单机可以维持百万级进程
- COP (Concurrency Oriented Programming)
- 通过消息通讯
- 不共享数据
- Pid/Name




```
B ! {self(),foo}

receive
    {From,Msg} ->
        Actions
end
```

调度器

- 公平调度，软实时系统关键之一
  - 部分任务占用过多资源
  - 其他任务饥饿，实时系统失效
- Reduction budget, 2000
  - 函数调用、BIF调用
  - ETS、发消息、表达式匹配
- 异步 IO 框架的问题

调度器

- 进程数怎么分配?
  - 快的模块进程数少
  - 慢的模块进程数多
- 监控 mailbox length
  - 哪些进程的 mailbox 过长?
- Profiling
  - fprof
  - systemtap，用源码编译

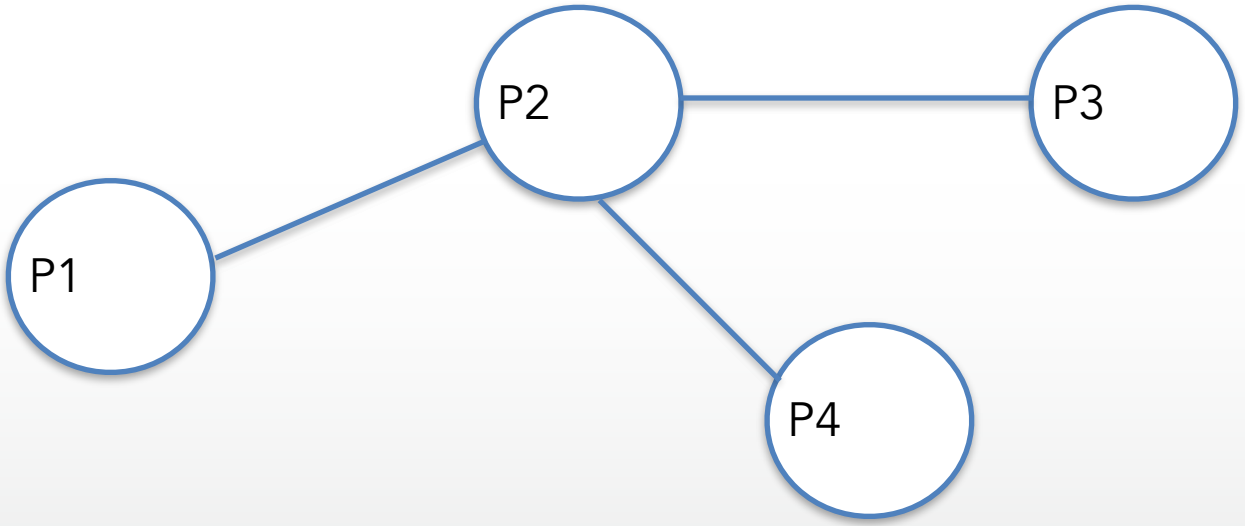
内存垃圾回收 GC

- 进程独立垃圾回收
- 垃圾回收时只有一个进程阻塞
  - Go/Akka/Java GC 抖动问题

- 软实时系统的另一个关键

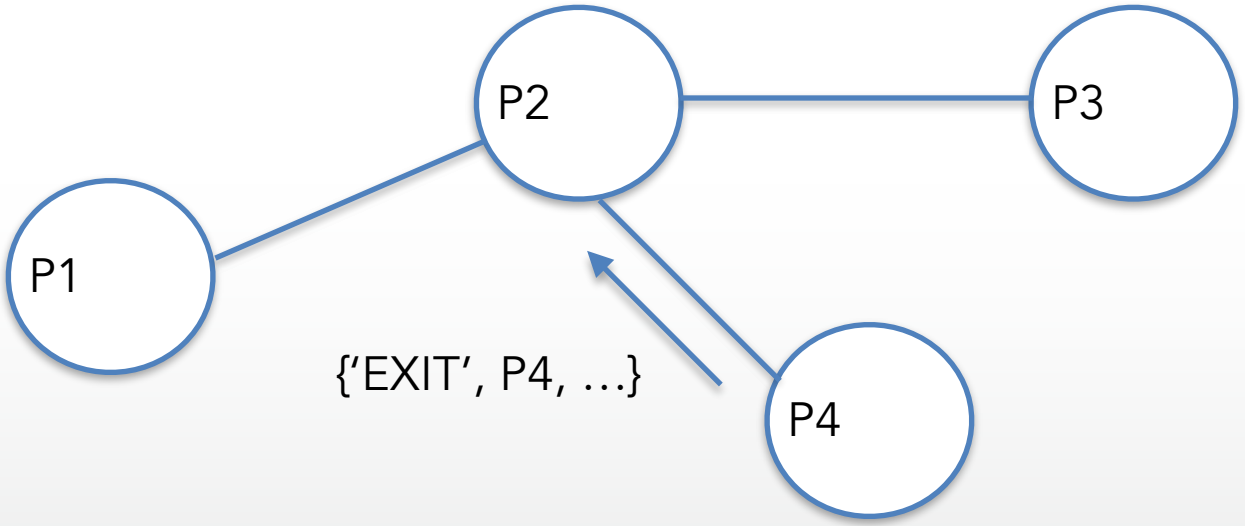
错误处理

- 错误无法避免
- 及时发现和恢复



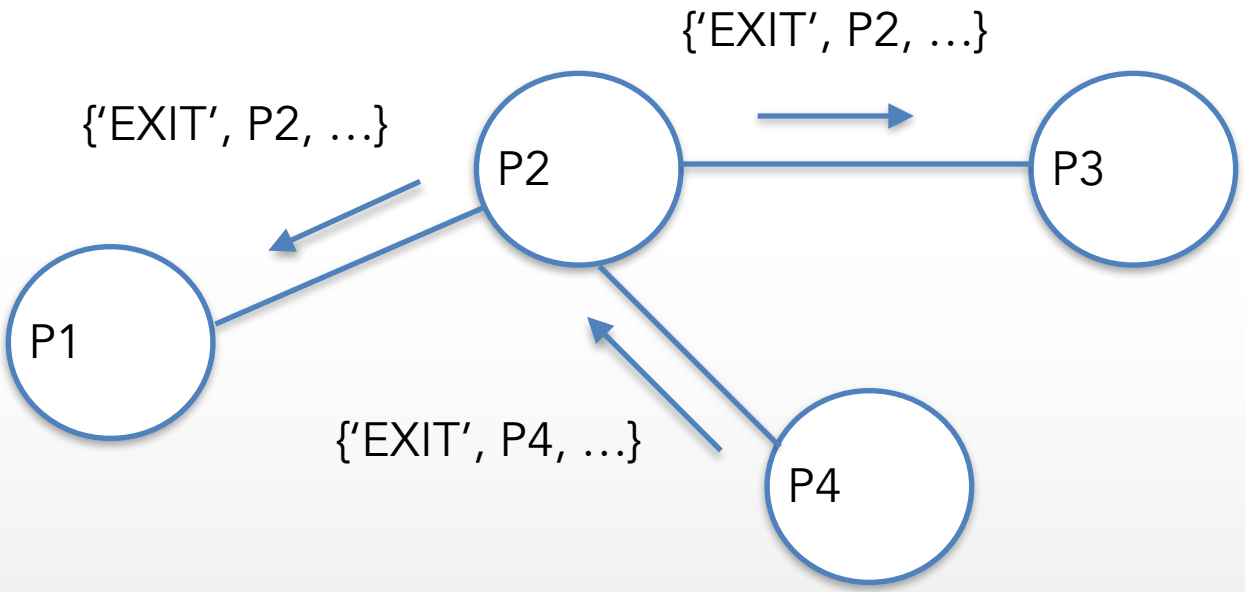
错误处理

- 错误无法避免
- 及时发现和恢复



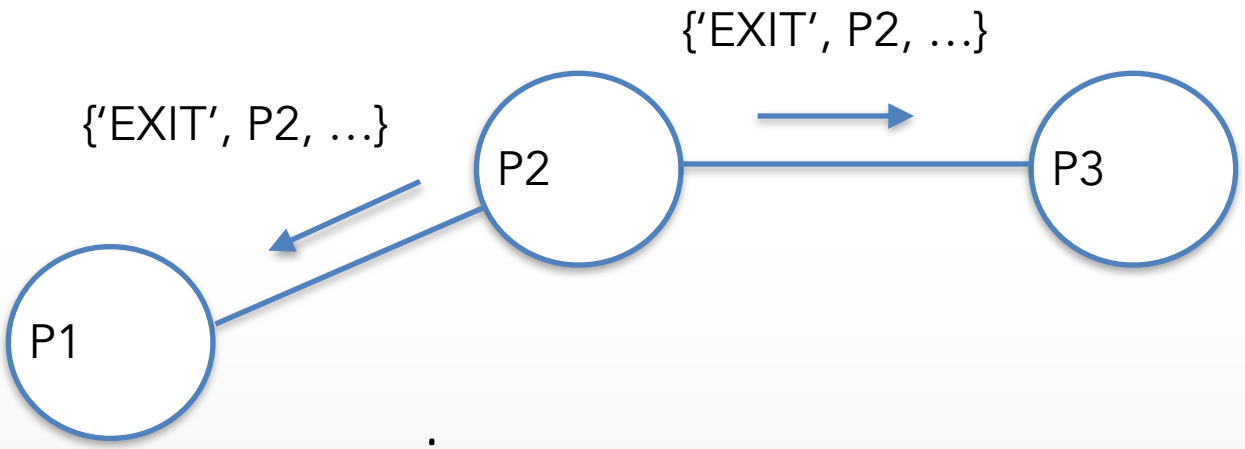
错误处理

- 错误无法避免
- 及时发现和恢复



错误处理

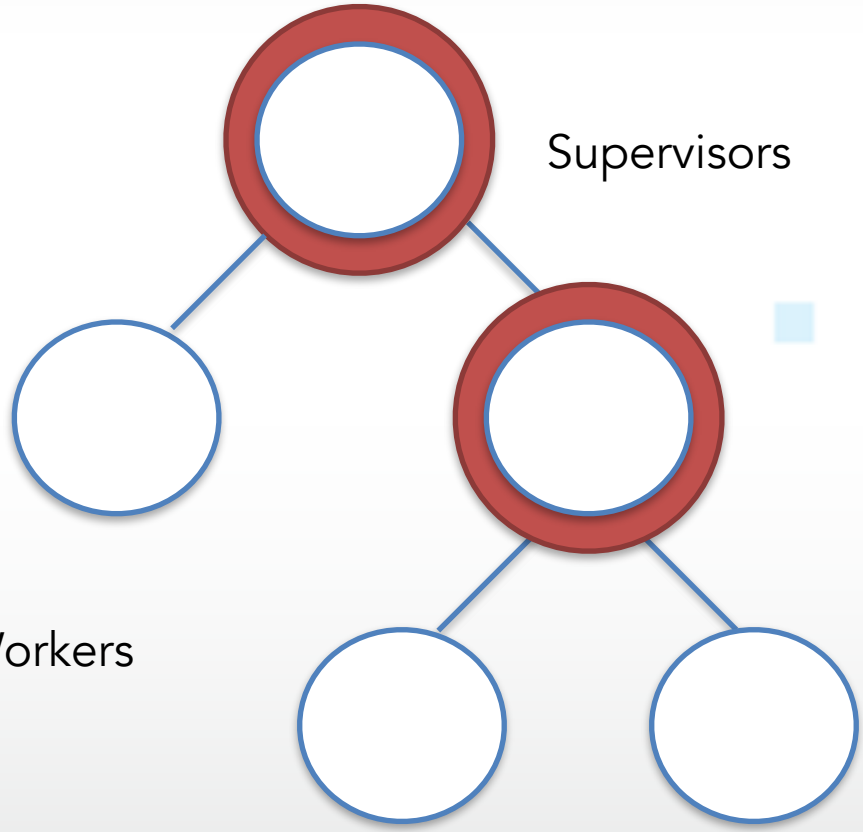
- 错误无法避免
- 及时发现和恢复




```
receive
     { 'EXIT' ,Pid,...}
-> ...
end
```

错误处理

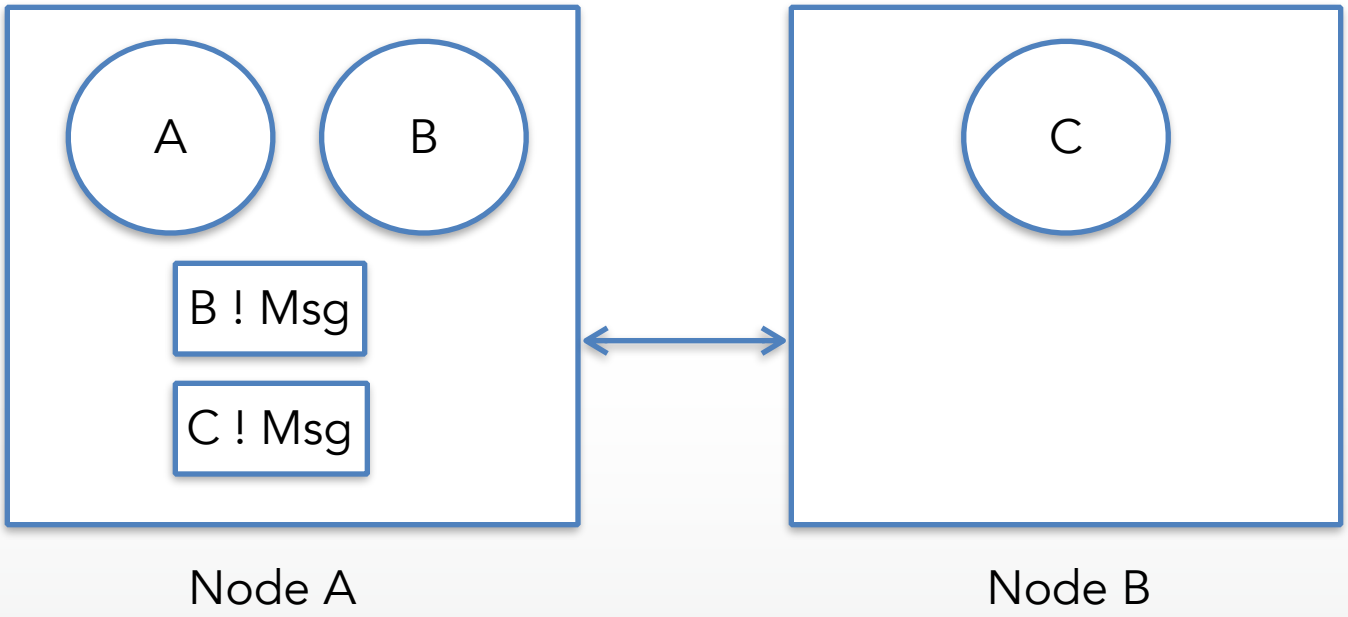
- 分层结构



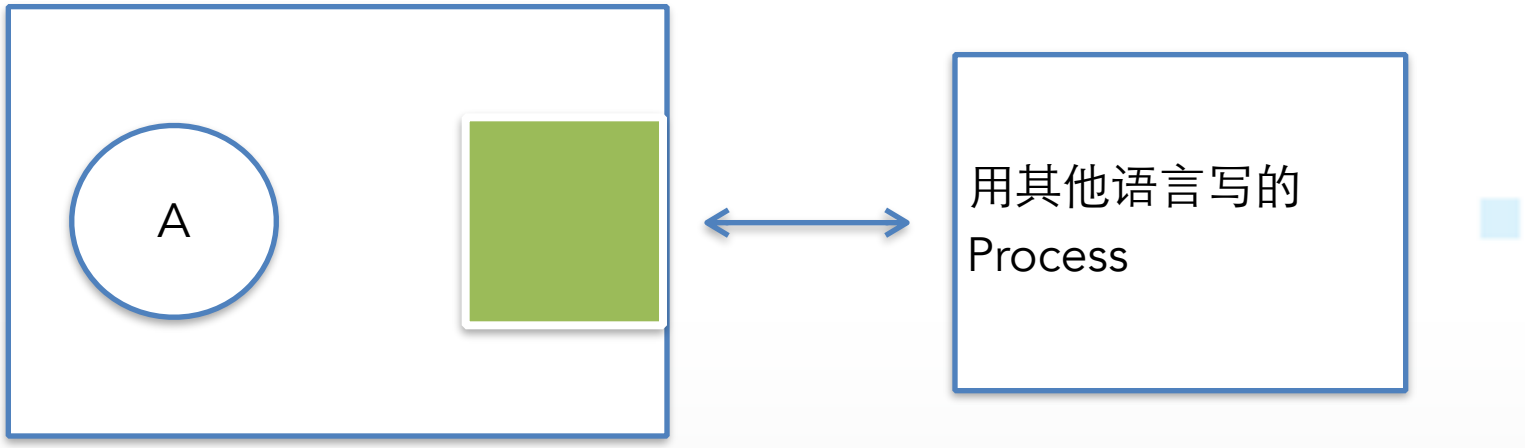
分布式支持

- 节点原生支持互联
  - 与其他节点的进程直接交互
  - Pg2/poolboy
- -setcookie

分布式支持



对外接口 Port



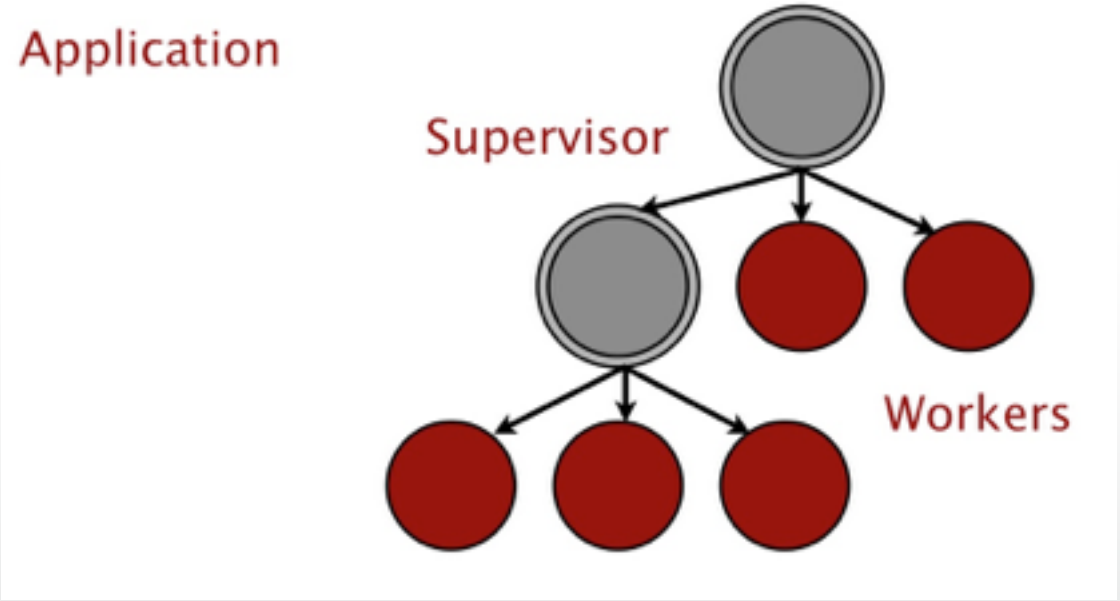
对外接口 Port

- Erlang
  - 结点管理
  - 路由调度
- C/C++
  - 算法
  - 存储

- Couchbase: Erlang + memcached

OTP

- Application
- Supervisor
- Worker
  - gen_server
  - gen_event
  - gen_fsm



OTP

- gen_server
  - RPC clint-server
  - call/cast/info
- gen_event
  - event/handler
- gen_fsm

gen_server 场景

- 维持长链接
- 管理外部依赖部件
  - rabbitmq/redis/couchbase/mysql…
- RESTful API
- Socket.io API

gen_event 场景

- 日志系统
  - file_handler
  - syslog_handler
- 实时统计系统
  - 一个事件多维度统计

gen_fsm 场景

- 处理步骤复杂的请求
  - 流程复杂，两次交互以后
  - 时间复杂，需要后台异步处理
- 海量事务
  - 一次发布千万级个对象
  - 分片，上万个 fsm 被创建
- fsm 本身被监控

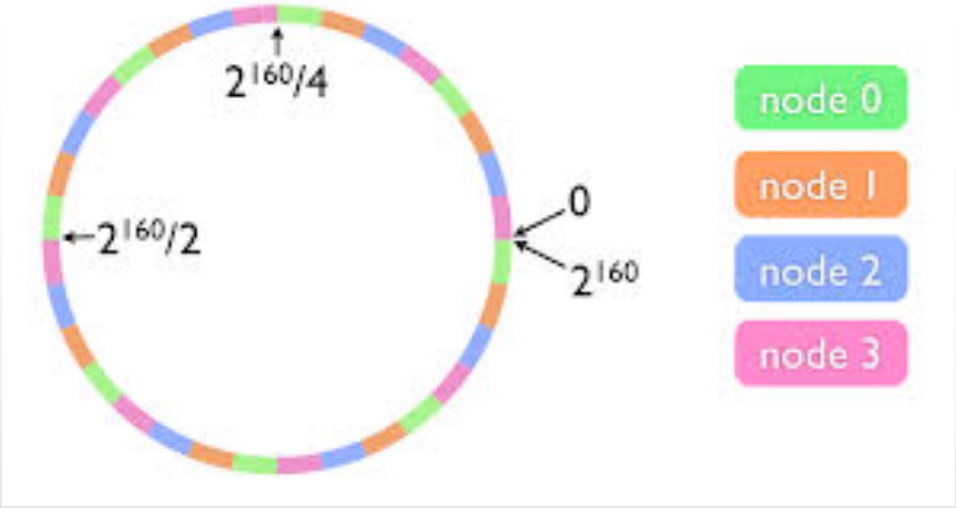
陷阱

- list_to_atom/1

- http://www.erlang.org/doc/efficiency_guide/commoncaveats.html

Riak Core下一个巨人?

- Riak Core：对等网，分布式 key 分片
- 基于 Amazon Dynamo Paper，已经出现一段时间
- Couchbase, Riak, Aerospike
  - 节点对等
  - 线性扩容
  - 持续运行
  - 最终一致



Riak Core

- 任何基于 Key 的系统
- Riak Core + X
  - Redis?
  - …

迁移到 Erlang

- 选择当前最合适的技术／平台
- 整个团队 Erlang 经验为零

迁移到 Erlang

- 语法的学习，容易
- 阅读开源项目代码，进阶
- 掌握调试、性能调优工具，进阶
- 理解 Erlang 虚拟机工作原理
  - 熟悉操作系统原理

迁移到 Erlang

- Code Review
  - 每一行、每一个命名
- eunit
- 集成测试
- Appmon/observer
- Mailbox 监控

迁移到 Erlang

- 招不到会 Erlang 的人而放弃 Erlang?
  - 团队 Leader 的认知
  - 不要只试图招会的人，而是招能学会的人

谢谢

Q & A

Yunba.io云巴

Brought by InfoQ

www.infoq.com/cn

InfoQueue

@InfoQ

infoqchina

软件
正在改变世界！