

# Reactive Programming with akka

吴雪峰@ThoughtWorks 6月22日

---

# “可用”

出错了还能用：软件、硬件、网络

高负荷， 高压力：秒杀 双11 v.s. 春节购票

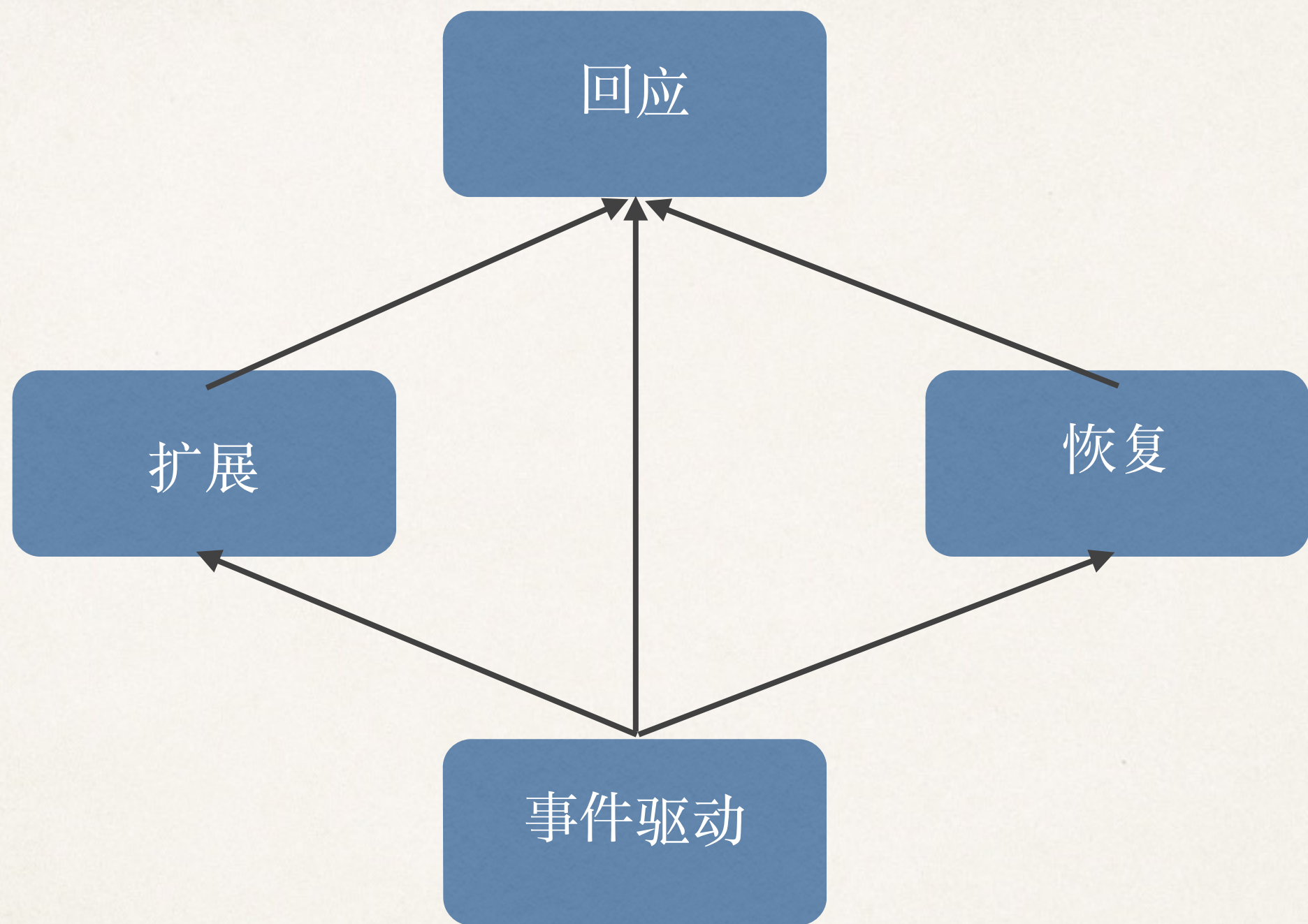
*-Johnny Appleseed*



# 单机单CPU时代过去了

---

- ❖ CPU摩尔定律不适合了，玩多核
- ❖ 云时代数据处理量大，单机扛不住了
- ❖ PC很便宜，大型服务器很贵





“Call back”

*–Johnny Appleseed*

```
$http({
    url: "http://api.weibo.com/u/123",
    method: "GET"
}).success(function (data, headers) {
    displayUser(data);
}).error(function (data, status) {
    var message = {data: data, status: status};
    EventService.fire(EventTopics.http_error, message);
});
```

```
$http({
    url: "http://api.wechat.com/user/123",
    method: "GET"
}).success(function (data, headers) {
    displayUserCombineWithWeibo(data, ???);
}).error(function (data, status) {
    var message = {data: data, status: status};
    EventService.fire(EventTopics.http_error, message);
});
```



# actor

---

- ❖ is an object with identity
- ❖ has a behaviour
- ❖ only interacts using a synchronous message passing
- ❖ an actor is single threaded



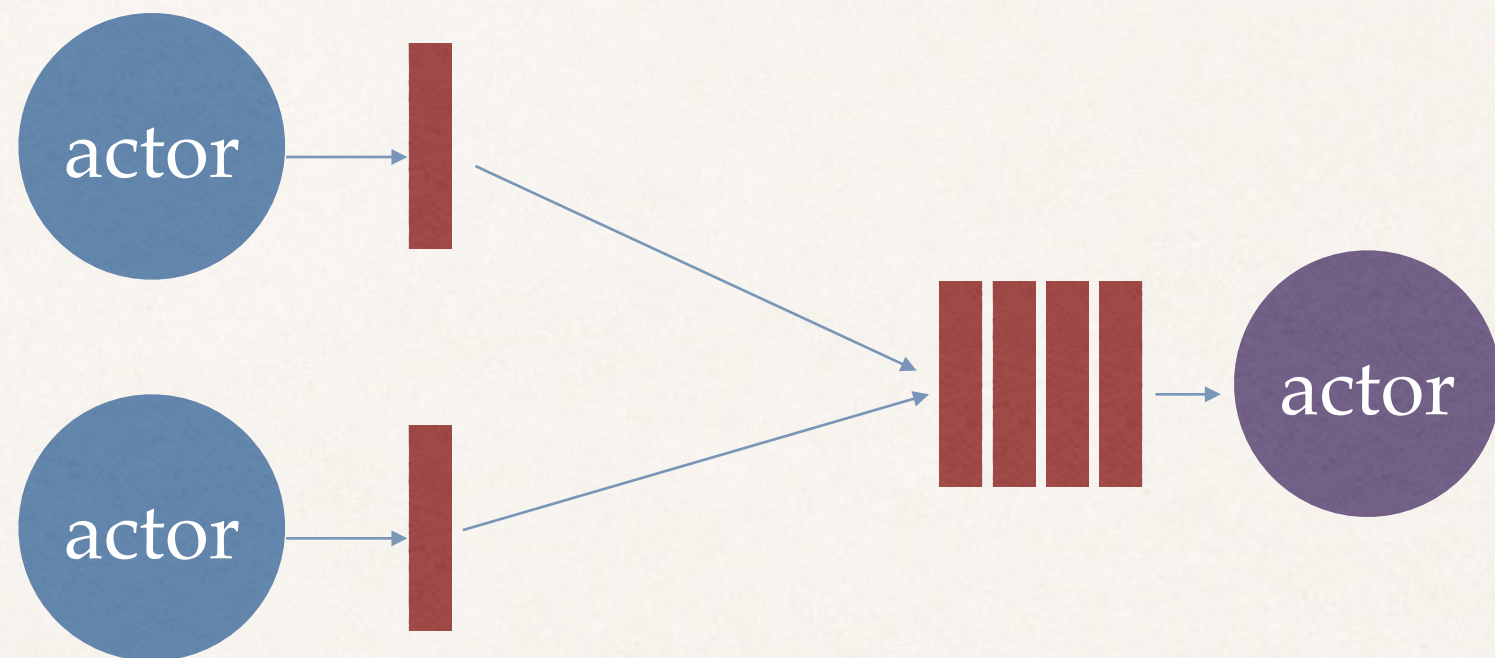


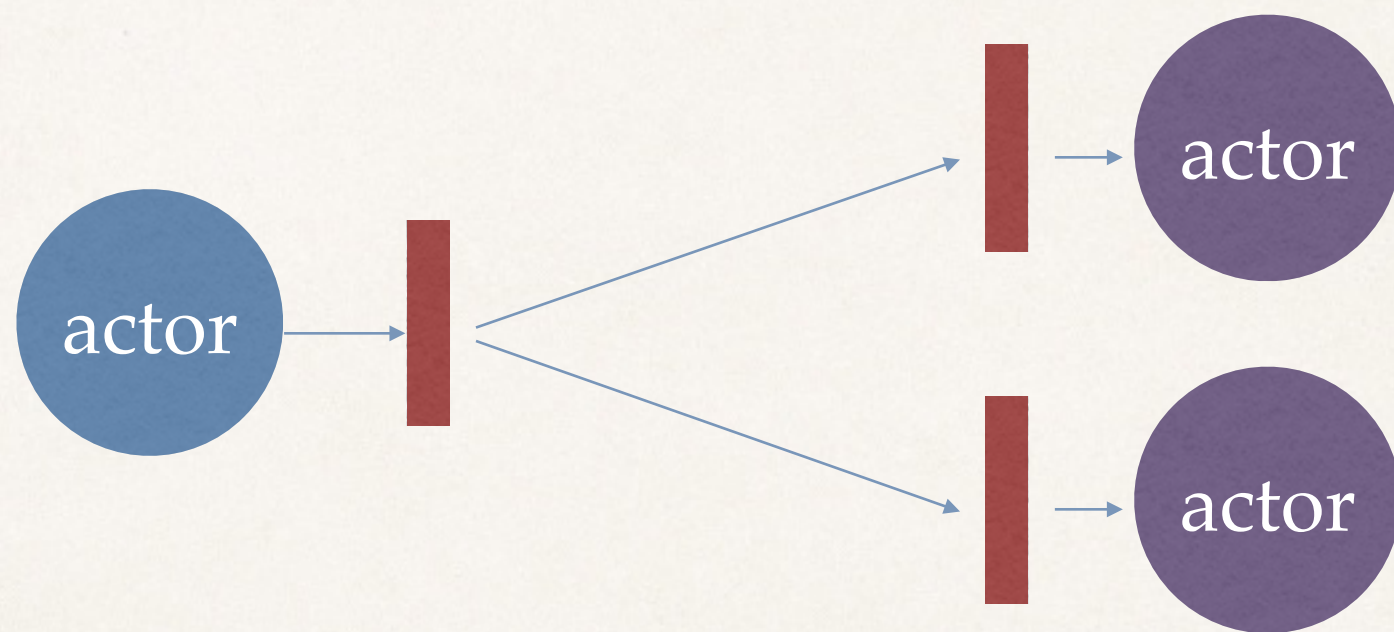
# Event Driven

---

- ❖ Events are first class
- ❖ Events are often represented as message
- ❖ Handlers of events are also first class
- ❖ Complex handlers can be composed from primitive ones









Show me the code





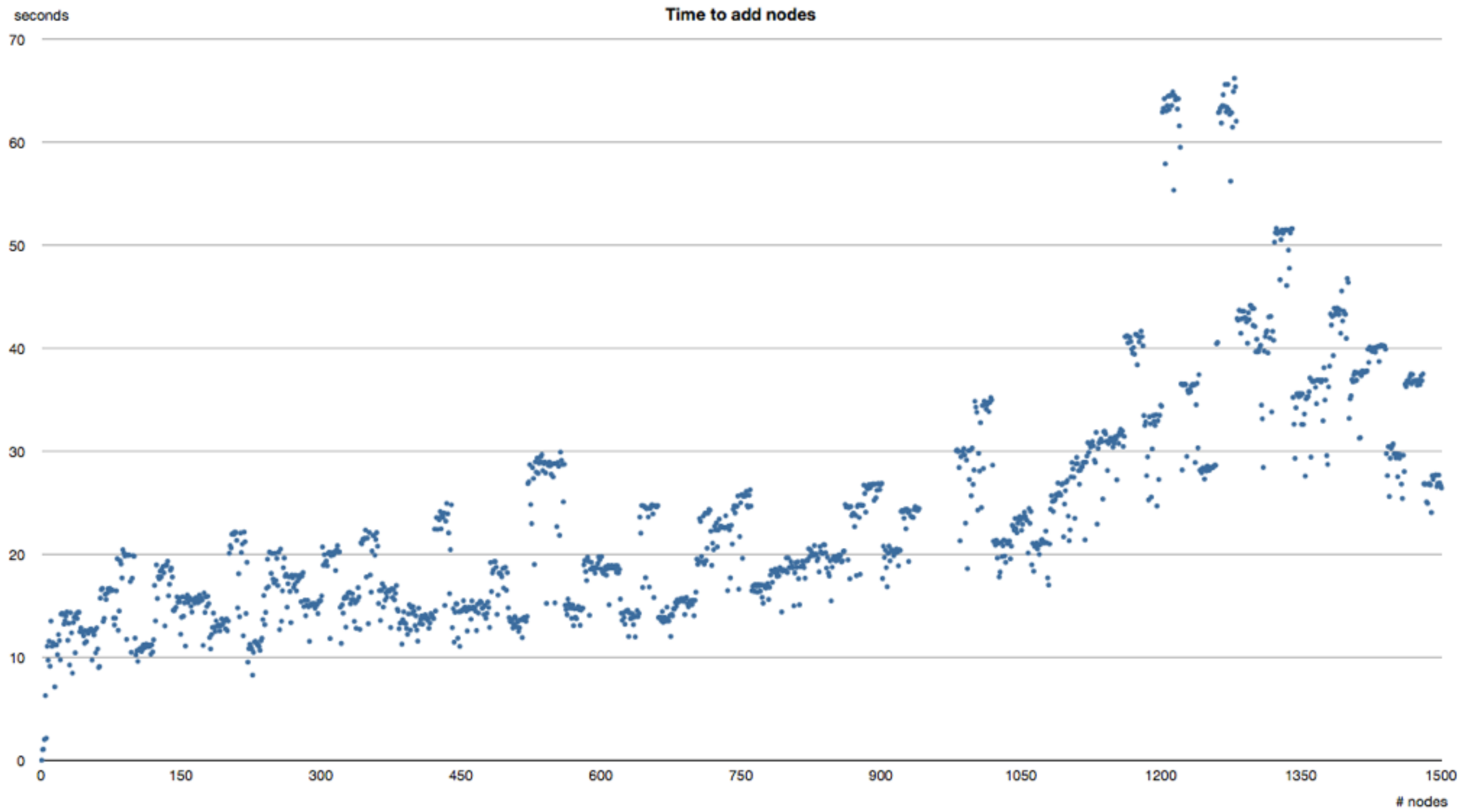
# Scalable

---

- ❖ Scale up: multi-core
- ❖ Scale out: multiple server

50 million messages per second - on  
a single machine











# Resilient

---

永不沉没



