

千万级消息系统构建实践

@刘浩

Worktile Web技术总监



Why

How

Future

Why

Why Erlang?

并发性

容错性

分布式

并行性

轻量级进程

Share nothing, copy everything

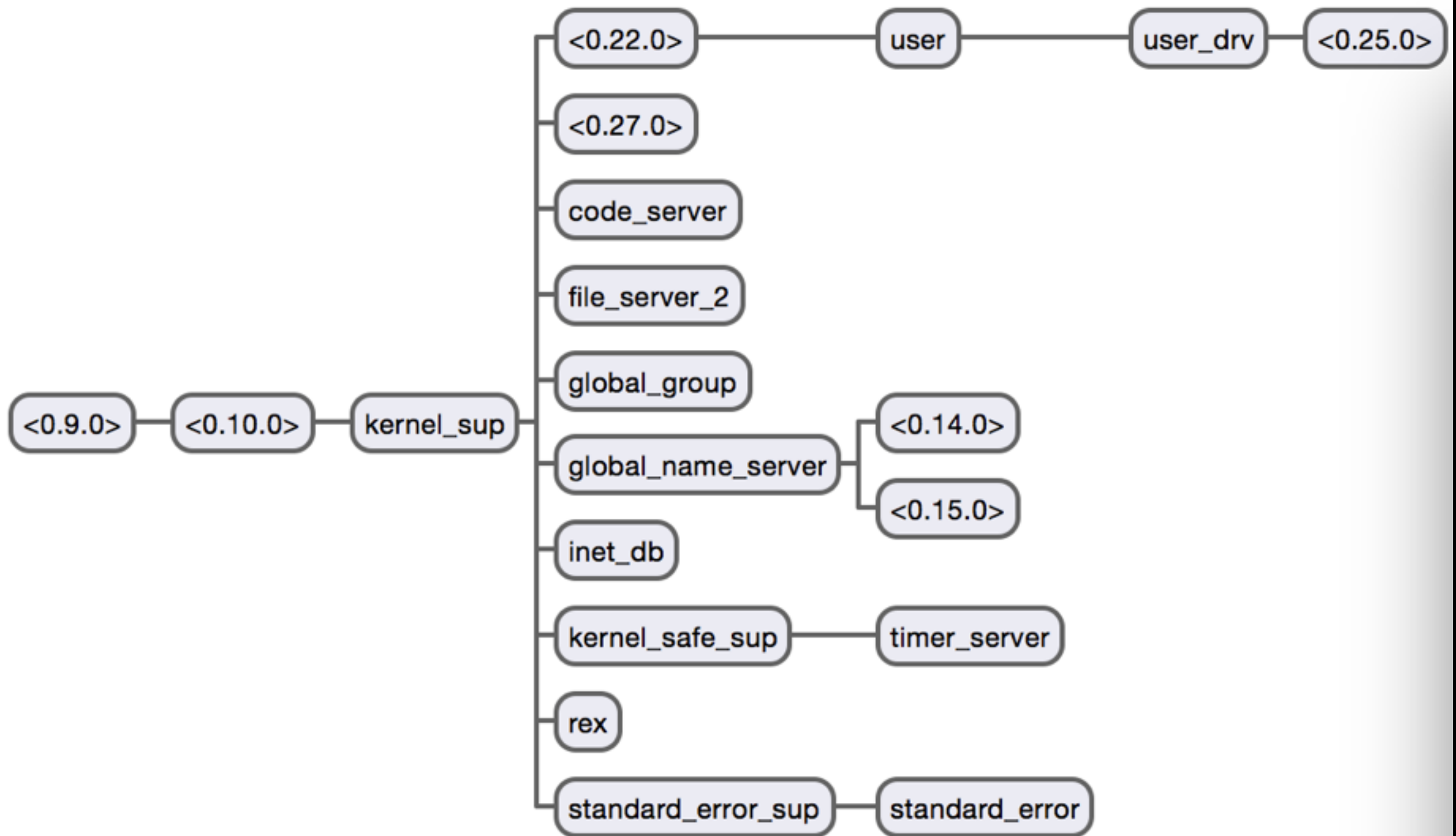
Soft real time

容错性

linked processes

generic behaviors

supervisor behaviors



分布式

位置透明

基于tcp/ip

STC.kent.ac.uk



dist2.erl

```
-module(dist2).  
-export([s/0]).  
s() -> register(server, self()), loop().  
loop() -> receive {M, Pid}  
               -> Pid ! M end,  
            loop().
```

FCC.erlang-consulting.com



```
erl -sname foo -setcookie cake
```

...

```
(foo@STC)1> spawn('bar@FCC', dist2, s, []).
```

```
<4824.44.0>
```

```
(foo@STC)2> {server, 'bar@FCC'} ! {hi, self()}.
```

```
{hi, <0.32.0>}
```

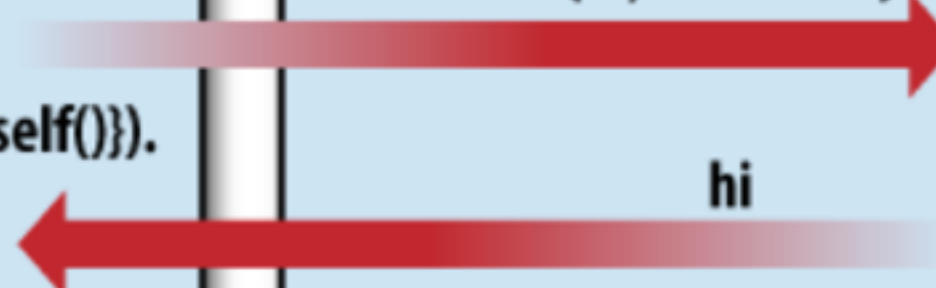
```
(foo@STC)3> flush().
```

```
Shell got hi
```

{hi, <0.32.0>}

server
=
loop()

hi



How

基于xmpp的ejabberd

xmpp优点与缺点

优点：

去中心化

开放协议

第三方支持

缺点：

数据量大

二进制数据的支持

ejabberd vs openfire

When it comes to sheer performance,
ejabberd is second to none!

Xeon E5-2650(2), 8G

100万同时在线

cpu 18%

memory 6G

Tsung

改进：

presence: 支持redis

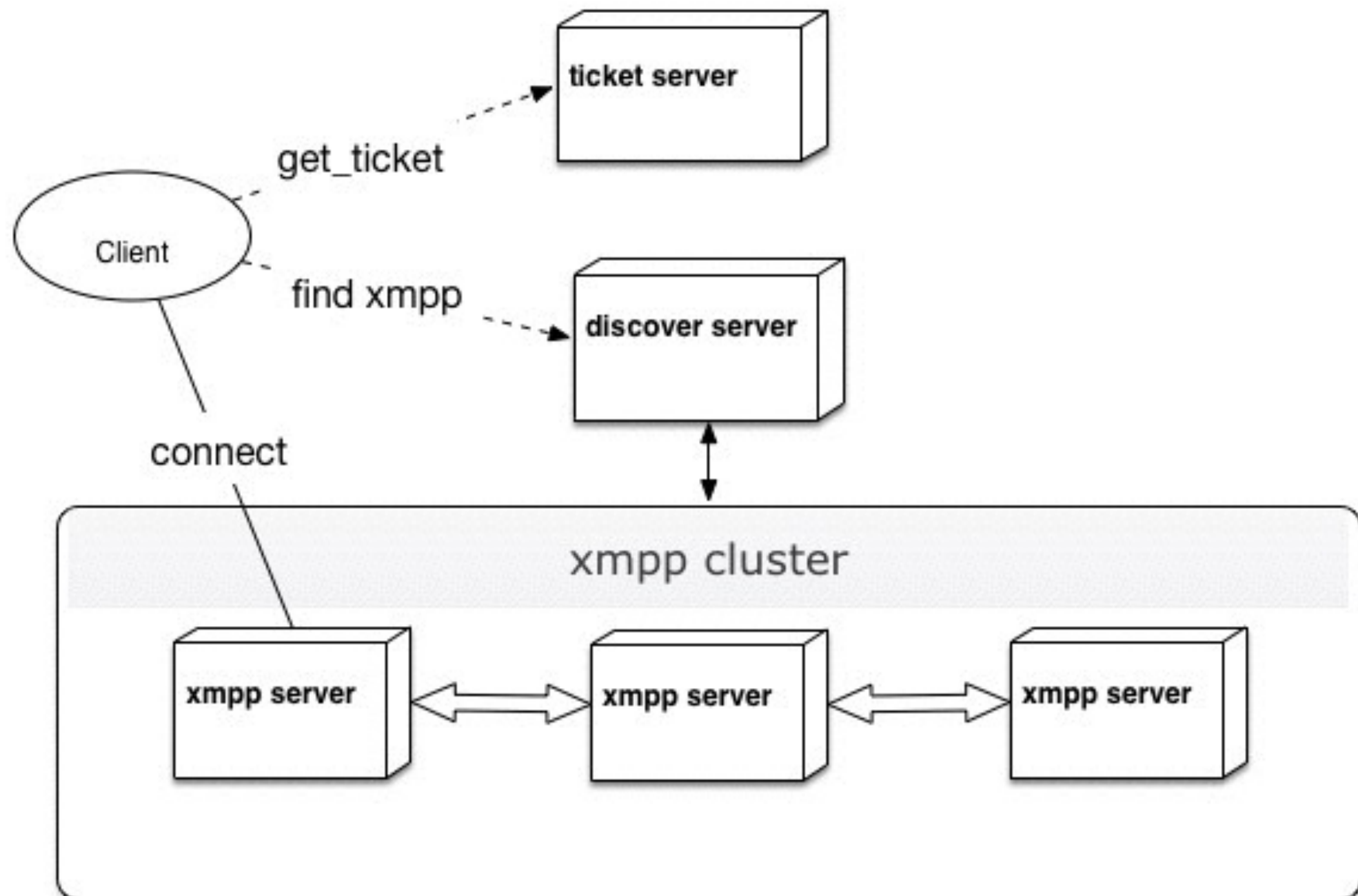
离线消息： 支持mongodb

连接池: poolboy

http 接口

mod_muc

worktile_xmpp_cluster



Demo

<https://github.com/ceshannon/shannonmet>

Future

二进制协议

管理平台

开源计划

Q & A



完全免费

你的网上办公室

worktile.com