用 Node.js 构建海量页面渲染服务

我是谁?

- 不四 @ tmall，Node.js 开发工程师
- dead-horse @ github，koa / cnpm 维护者

海量?


搜索商品/店铺
12.12 品牌盛典
1212抢先进场
爆款预定中
积分乐园
充值
闪购全球
聚划算
必抢秒杀
天猫热点
1分钱任性送！玛氏食品，满99减40！
0.01元起＋满99减40
重磅活动天天看
05 预售秒杀
06 预售秒杀
07 低至1元
12.12 品牌盛典
放价狂欢
19.9元封顶秒杀
数码抢先定
最高降千元
大牌臻选
预售加赠
建材会场
12.12元秒杀
居家厨房
全场买送
双12直播间 3秒后刷新
秒杀播报
更多


搜索商品/店铺
12.12 品牌盛典
1212抢先进场
爆款预定中
积分乐园
充值
闪购全球
聚划算
秒杀
必抢秒杀
天猫热点
1分钱任性送！玛氏食品，满99减40！
0.01元起＋满99减40
重磅活动天天看
05 预售秒杀
06 预售秒杀
07 低至1元
放价狂欢
19.9元封顶秒杀
数码抢先定
最高降千元
大牌臻选
预售加赠
建材会场
12.12元秒杀
居家厨房
全场买送
双12直播间 3秒后刷新
秒杀播报
更多


1212女装会场
女装会场
抢购必杀技
年终盛典 劲爆开抢
人气精品
大牌精选
抢先试穿
温暖寒冬
Levi's
折上满立减
asos
满699减100
Meters/bonwe
1000名免单
JEANSWEST
4000份免单
TONLION
免单不停止
paul frank
0点抢免单
C&A
满499减10
SEMIR 森馬
买100返5元
ESPRIT
满699减100
甘甘的世界
加购抢红包
TOPSHOP
全场2折起
千仞岗 CHERICOM
全场99元起
人气精品
主会场
换会场
必抢

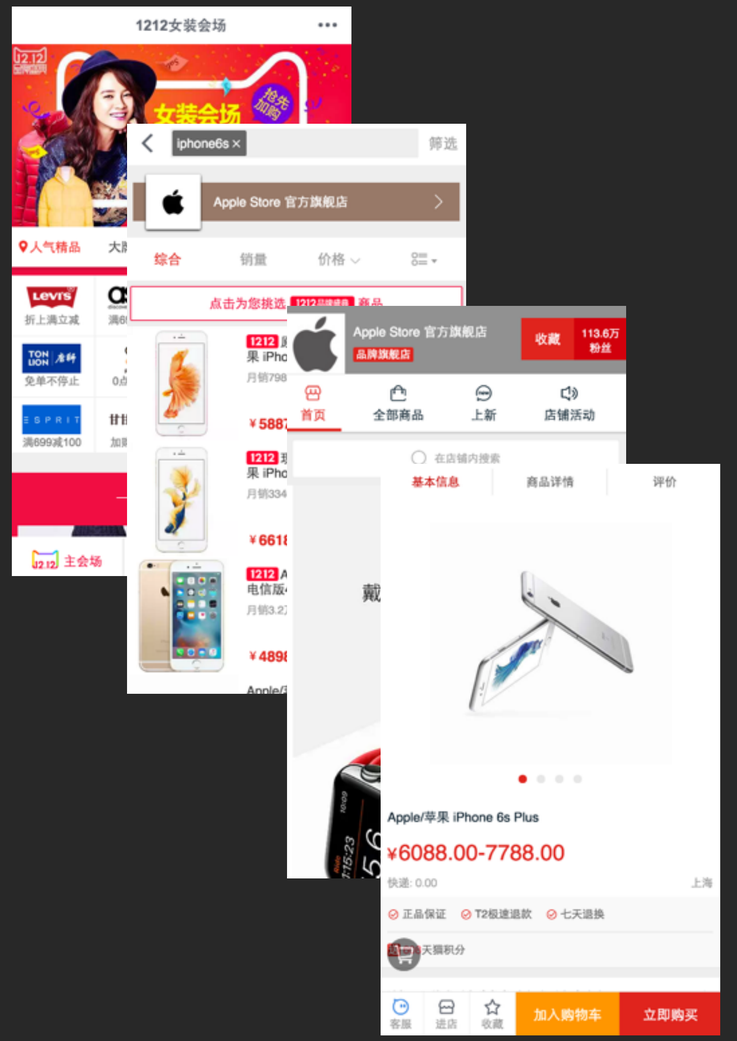

搜索商品/店铺
12.12 品牌盛典
1212抢先进场
爆款预定中
积分乐园
充值
闪购全球
聚划算
秒杀
必抢秒杀
天猫热点
1分钱任性送！玛氏食品，满99减40！
0.01元起＋满99减40
重磅活动天天看
05 预售秒杀
06 预售秒杀
07 低至1元
放价狂欢
19.9元封顶秒杀
数码抢先定
最高降千元
大牌臻选
预售加赠
建材会场
12.12元秒杀
居家厨房
全场买送
双12直播间 3秒后刷新
秒杀播报
更多


1212女装会场
女装会场
抢先加购
iphone6s
筛选
Apple Store 官方旗舰店
人气精品
综合
销量
价格
点击为您挑选 1212品牌提供 商品
1212 原封国行【分期免息】Apple/苹果 iPhone 6s 4.7英寸 全网通手机
月销7985笔 能良数码官方旗舰店
¥5887.00 免运费
1212 现货【送壳+钢化膜】Apple/苹果 iPhone 6s Plus 5.5英寸全网通
月销3340笔 能良数码官方旗舰店
¥6618.00 免运费
1212 Apple/苹果iPhone 6s 移动联通电信版4G手机 16G 苏宁国行全网通
月销3.2万笔 苏宁易购官方旗舰店
¥4898.00
主会场


搜索商品/店铺
12.12 品牌盛典
1212抢先进场
爆款预定中
积分乐园
充值
闪购全球
聚划算
必抢秒杀
天猫热点
1分钱任性送！玛氏食品，满99减40！
0.01元起＋满99减40
重磅活动天天看
05 预售秒杀
06 预售秒杀
07 低至1元
放价狂欢
19.9元封顶秒杀
数码抢先定
最高降千元
大牌臻选
预售加赠
建材会场
12.12元秒杀
居家厨房
全场买送
双12直播间 3秒后刷新
秒杀播报
更多


1212女装会场
女装会场
抢先加购
iphone6s
筛选
Apple Store 官方旗舰店
人气精品
综合
销量
价格
Levi's
折上满立减
点击为您挑选
主会场
Apple Store 官方旗舰店
品牌旗舰店
收藏
113.6万 粉丝
首页
全部商品
上新
店铺活动
在店铺内搜索
WATCH
戴着它，就会爱上它。
购买


搜索商品/店铺
12.12 品牌盛典
1212抢先进场
爆款预定中
积分乐园
充值
闪购全球
聚划算
必抢秒杀
天猫热点
1分钱任性送！玛氏食品，满99减40！
0.01元起＋满99减40
重磅活动天天看
05 预售秒杀
06 预售秒杀
07 低至1元
放价狂欢
19.9元封顶秒杀
数码抢先定
最高降千元
大牌臻选
预售加赠
建材会场
12.12元秒杀
居家厨房
全场买送
双12直播间 3秒后刷新
秒杀播报
更多


1212女装会场
女装会场
抢先加购
iphone6s
筛选
Apple Store 官方旗舰店
人气精品
综合
销量
价格
Levi's
折上满立减
免单不停止
满699减100
主会场
Apple Store 官方旗舰店
品牌旗舰店
收藏
113.6万 粉丝
首页
全部商品
上新
店铺活动
在店铺内搜索
基本信息
商品详情
评价
Apple/苹果 iPhone 6s Plus
¥6088.00-7788.00
快递: 0.00
上海
正品保证
T2极速退款
七天退换
天猫积分
客服
进店
收藏
加入购物车
立即购买

从两年前说起…

- 天猫的大量页面基于 php 渲染
- 数据模板打包后推送 CDN
- 每次请求都渲染页面

- 大量文件在网络情况不一的 CDN 节点同步非常困难
- 每次请求渲染导致大量的机器资源浪费
- php 太灵活，多业务共享难以升级，性能低

系统要重构，契机出现了

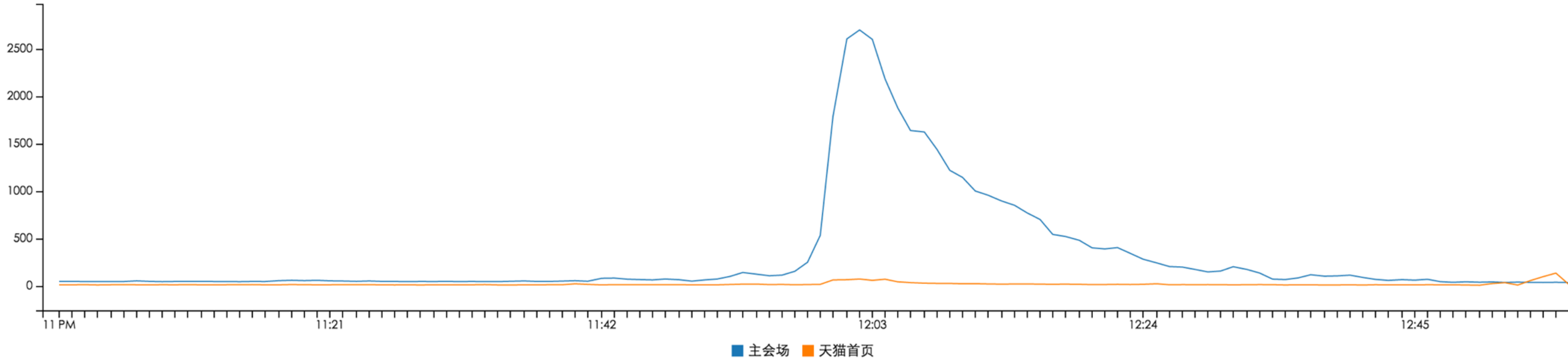
Why Node.js

“Any application that can be written in JavaScript, will eventually be written in JavaScript.”

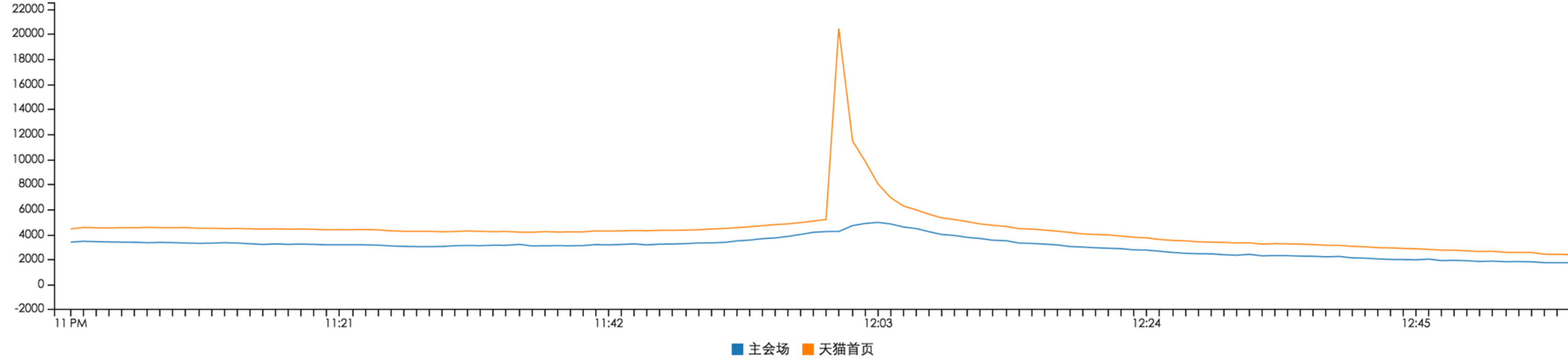
– Jeff Atwood

- 模板渲染层 js 的独特优势：前后端模板共享
- 阿里内部 Node.js 环境十分完善
- 轻量级、高性能

主会场（php）和天猫首页（node）RT 对比



主会场（php）和天猫首页（node）QPS 对比



回归渲染的本质

回归渲染的本质

- 前端开发编写维护的 模板

回归渲染的本质

- 前端开发编写维护的 模板
- 运营维护或者后端系统产出的 数据

回归渲染的本质

- 前端开发编写维护的 模板
- 运营维护或者后端系统产出的 数据
- 通过一个渲染容器进行合并产出 页面

```
// index.xtpl
{{#each ($data.items) }}
  <p><a href="{{this.link}}">{{this.name}}</a></p>
{{/each}}
```

```
// items.json
[{
  "link": "//detail.tmall.com/item.htm?id=10126492964",
  "name": "item one"
}, {
  "link": "//detail.tmall.com/item.htm?id=10126492964",
  "name": "item tow"
}]
```

```
// index.xtpl
{{#each ($data.items) }}
  <p><a href="{{this.link}}">{{this.name}}</a></p>
{{/each}}
```

```
// items.json
[{
  "link": "//detail.tmall.com/item.htm?id=10126492964",
  "name": "item one"
}, {
  "link": "//detail.tmall.com/item.htm?id=10126492964",
  "name": "item tow"
}]
```

```
// result
<p><a href="//detail.tmall.com/item.htm?id=10126492964">item one</a></p>
<p><a href="//detail.tmall.com/item.htm?id=10126492964">item tow</a></p>
```

扩展上下文

- 系统时间、系统环境
- http 请求上下文
- 通用工具方法，安全处理包

```
{{#if ($system.now.before('2015-11-11 00:00:00'))}}
  <h3>Comming soon...</h3>
{{else}}
  {{#each ($data.items) }}
    <p><a href="{{this.link}}">{{this.name}}</a></p>
  {{/each}}

  {{#if (!$http.detector.app.is)}}
    {{include ("downloadPage")}}
  {{/if}}
{{/if}}
```

模板中的静态资源管理

- 前后端统一的资源加载器
- 基于统一的 seed 控制依赖以及版本
- 输出 combo url 到页面
- 依赖自动去重，css / js 头尾加载

```
// 模板
<html>
  <head>
    {{placeholder('css')}}
  </head>
    {{require('mui/mod/a.js', 'mui/mod/b.js')}}
    {{require('mui/mod/a.css', 'mui/mod/b.css')}}
  <body>
  {{placeholder('js')}}
  </body>
</html>
```

```
// 渲染出的 html
<html>
  <head>
    <link rel="stylesheet" href="//g.alicdn.com/mui/mod/3.0.0/??a.css,b.css" />
  </head>
  <body>
    <h3>Contents</h3>
    <script src="//g.alicdn.com/mui/mod/3.0.0/??a.js,b.js"></script>
  </body>
</html>
```

模块化

```
{{!多页面共享的头尾}}
{{extend("solution://base/")}}

{{#block(“body”)}}
  {{!引入模块}}
  {{use ("mui://zebra-act-flashy/", $data.items)}}
  {{use ("mui://zebra-act-shop/", $data.shops)}}
{{/block}}
```

模块化

```
// 模块 mui/zebra-act-flashy
{{! load css and js }}
{{ require ("mui/zebra-act-flashy/index.css") }}
{{ require ("mui/zebra-act-flashy/index.js") }}

<div class="zebra-act-flashy">
  <ul class="mui-act-flashy-items js_flashy">
    {{#each(items)}}
      <li class="mui-act-flashy-item" data-mark="assist"  data-tag="item">
        <a class="mui-act-flashy-item-link" href="{{this.url}}" target="_blank">
          <div class="mui-act-flashy-item-text">{{this.content}}</div>
        </a>
      </li>
    {{/each}}
   </ul>
</div>
```

1212品牌盛典 •••

— 超级秒杀 14点整开始 —

抢券更实惠
¥50 优惠券
无门槛优惠券
1元秒杀

梦百合智能床垫

12.12元秒杀

高露洁明星款

0.01元秒杀

男装

男装会场

亿万优惠券 抢折上折

男装保暖

抢无门槛券

男装商场

抢先囤5折起

莫畏快轻奢

莫畏or1折起

拓路者品牌

满269减20元

酷衣购男装

满499减50元

ANGMANZO

10元优惠券

主会场 换会场 必抢

1212品牌盛典 •••

— 超级秒杀 14点整开始 —

抢券更实惠
¥50 优惠券
无门槛优惠券
1元秒杀

梦百合智能床垫

12.12元秒杀

高露洁明星款

0.01元秒杀

男装

男装会场

亿万优惠券 抢折上折

男装保暖

抢无门槛券

男装商场

抢先囤5折起

莫畏快轻奢

莫畏or1折起

拓路者品牌

满269减20元

酷衣购男装

满499减50元

ANGMANZO

10元优惠券

主会场 换会场 必抢

1212品牌盛典

资源位

— 超级秒杀 14点整开始 —

抢券更实惠
¥50 优惠券
无门槛优惠券
1元秒杀

梦百合智能床垫
12.12元秒杀

高露洁明星款
0.01元秒杀

男装

男装会场
亿万优惠券 抢折上折

男装保暖
抢无门槛券

男装商场
抢先囤5折起

莫畏快轻奢
莫畏or1折起

拓路者品牌
满269减20元

酷衣购男装
满499减50元

ANGMANZO
10元优惠券

主会场 换会场 必抢

1212品牌盛典 •••

— 超级秒杀 14点整开始 —

券更实惠
¥50 优惠券
无门槛优惠券
1元秒杀

梦百合智能床垫

12.12元秒杀

高露洁明星款

0.01元秒杀

男装

男装会场

亿万优惠券 抢折上折

男装保暖

抢无门槛券

男装商场

抢先囤5折起

莫畏快轻奢

莫畏or1折起

拓路者品牌

满269减20元

酷衣购男装

满499减50元

ANGMANZO

10元优惠券

主会场 换会场 必抢

1212品牌盛典 •••

— 超级秒杀 14点整开始 —

抢券更实惠
¥50 优惠券
无门槛优惠券
1元秒杀

梦百合智能床垫

12.12元秒杀

高露洁明星款

0.01元秒杀

楼层

男装

男装会场

亿万优惠券 抢折上折

男装保暖

抢无门槛券

男装商场

抢先囤5折起

莫畏快轻奢

莫畏or1折起

拓路者品牌

满269减20元

酷衣购男装

满499减50元

ANGMANZO

10元优惠券

主会场 换会场 必抢

1212品牌盛典

— 超级秒杀 14点整开始 —

抢券更实惠
¥50 优惠券
无门槛优惠券
1元秒杀

梦百合智能床垫

12.12元秒杀

高露洁明星款

0.01元秒杀

男装

男装会场

亿万优惠券 抢折上折

男装保暖

抢无门槛券

男装商场

抢先囤5折起

莫畏快轻奢

莫畏or1折起

拓路者品牌

满269减20元

酷衣购男装

满499减50元

ANGMANZO

10元优惠券

导航
主会场
换会场
必抢


```
// 引入跨页面共享的通用头尾
{{extend("solution://base/")}}

{{#block(“body")}}
  // 模块化构建页面
  {{use ("mui://headbanner/", $data.modules[1])}}
  {{use ("mui://banner/", $data.modules[2])}}
  {{use ("mui://multi-banner/", $data.modules[3])}}
  {{use ("mui://floor/", $data.modules[4])}}
  {{use ("mui://nav/", $data.modules[5])}}
{{/block}}
```

模块化 ->

模块化 -> 规模化

模块化 -> 规模化

- 跨业务共享模块

模块化 -> 规模化

- 跨业务共享模块
- 面向运营的可视化页面搭建平台

模块化 -> 规模化

- 跨业务共享模块
- 面向运营的可视化页面搭建平台
- Native 页面搭建

渲染服务

- 服务基于 koa 框架
- 扩展 xtemplate 模板
- 阿里云 oss 提供模板和数据的存储支持
- 阿里 CDN 提供缓存支持

koa, 不仅仅是 generator

Middlewares are Decorators

Middlewares are Decorators

- Express：中间件顺序执行

Middlewares are Decorators

- Express：中间件顺序执行
- Koa：包裹在后面所有中间件的装饰器

```
function caculate(times) {
  let i = 0;
  while (i < times) i++;
}
```

```
function caculate(times) {
  let i = 0;
  while (i < times) i++;
}

// decorator
function log(fn) {
  return function (...args) {
    const start = Date.now();
    fn(...args);
    const used = Date.now() - start;
    console.log(`call ${fn.name}(${args}) used ${used}ms`);
  }
}
```

```
function caculate(times) {
  let i = 0;
  while (i < times) i++;
}

// decorator
function log(fn) {
  return function (...args) {
    const start = Date.now();
    fn(...args);
    const used = Date.now() - start;
    console.log(`call ${fn.name}(${args}) used ${used}ms`);
  }
}

caculate = log(caculate);
caculate(100000);   // call caculate(100000) used 0ms
caculate(10000000); // call caculate(10000000) used 6ms
```

```
// 设置响应时间 header
function* responseTime(next) {
  // 所有的后续中间件之前执行
  const start = Date.now();
  yield next;
  // 所有的后续中间件之后执行
  const used = Date.now() - used;
  this.set('X-Response-Time', `${used}ms`);
}
```

```
// 错误处理
function* errorHandler(next) {
  try {
    yield next;
  } catch (err) {
    logger.error(err);
    this.status = 500;
    this.body = err.message;
  }
}
```

Abstract Context

Abstract Context

- express
  - 无法对已经发送的 body 修改
  - 不能在设置 body 后设置响应头

Abstract Context

- express
  - 无法对已经发送的 body 修改
  - 不能在设置 body 后设置响应头
- koa
  - 在所有中间件执行完成之后再发送响应
  - 中间件执行过程中可以对 body 和 header 任意修改

```
app.use(render); // 渲染页面

function* render() {
  const renderResult = yield _render(this.req.path)
  this.charset = renderResult.charset;
  this.body = renderResult.body;
}
```

```
app.use(gbk);  // 对渲染的结果做编码转换

app.use(render); // 渲染页面

function* render() {
  const renderResult = yield _render(this.req.path)
  this.charset = renderResult.charset;
  this.body = renderResult.body;
}

function* gbk(next) {
  yield next;
  if (this.charset !== 'gbk') return;
  this.body = iconv.encode(this.body, 'gbk');
}
```

```
app.use(gzip); // 对输出的 html 做压缩
app.use(gbk);  // 对渲染的结果做编码转换
app.use(render); // 渲染页面
function* render() {
  const renderResult = yield _render(this.req.path)
  this.charset = renderResult.charset;
  this.body = renderResult.body;
}
function* gbk(next) {
  yield next;
  if (this.charset !== 'gbk') return;
  this.body = iconv.encode(this.body, 'gbk');
}
function* gzip(next) {
  yield next;
  if (this.acceptsEncodings('gzip') !== gzip) return;
  this.body = yield _gzip(this.body);
}
```

```
const koa = require('koa');
const app = koa();
```

```
app.on('error', err => logger.error(err));
app.listen(port);
```

```
const koa = require('koa');
const app = koa();
```

```
app.use(render);          // 渲染

app.on('error', err => logger.error(err));
app.listen(port);
```

```
const koa = require('koa');
const app = koa();
```

```
app.use(renderLog);     // 记录渲染日志
app.use(render);        // 渲染

app.on('error', err => logger.error(err));
app.listen(port);
```

```
const koa = require('koa');
const app = koa();
```

```
app.use(backup);          // 备份容灾
app.use(renderLog);       // 记录渲染日志
app.use(render);          // 渲染

app.on('error', err => logger.error(err));
app.listen(port);
```

```
const koa = require('koa');
const app = koa();
```

```
app.use(detector);          // 判断终端类型
app.use(backup);            // 备份容灾
app.use(renderLog);         // 记录渲染日志
app.use(render);            // 渲染

app.on('error', err => logger.error(err));
app.listen(port);
```

```
const koa = require('koa');
const app = koa();
```

```
app.use(cacheControl); // 设置 CDN 缓存头
app.use(detector);     // 判断终端类型
app.use(backup);       // 备份容灾
app.use(renderLog);    // 记录渲染日志
app.use(render);       // 渲染

app.on('error', err => logger.error(err));
app.listen(port);
```

```
const koa = require('koa');
const app = koa();



app.use(meta);              // 记录时间等信息
app.use(cacheControl);      // 设置 CDN 缓存头
app.use(detector);          // 判断终端类型
app.use(backup);            // 备份容灾
app.use(renderLog);         // 记录渲染日志
app.use(render);            // 渲染

app.on('error', err => logger.error(err));
app.listen(port);
```

```
const koa = require('koa');
const app = koa();


app.use(trace);           // 打印 trace log
app.use(meta);            // 记录时间等信息
app.use(cacheControl);    // 设置 CDN 缓存头
app.use(detector);        // 判断终端类型
app.use(backup);          // 备份容灾
app.use(renderLog);       // 记录渲染日志
app.use(render);          // 渲染

app.on('error', err => logger.error(err));
app.listen(port);
```

```
const koa = require('koa');
const app = koa();

app.use(onerror);          // 异常处理
app.use(trace);            // 打印 trace log
app.use(meta);             // 记录时间等信息
app.use(cacheControl);     // 设置 CDN 缓存头
app.use(detector);         // 判断终端类型
app.use(backup);           // 备份容灾
app.use(renderLog);        // 记录渲染日志
app.use(render);           // 渲染

app.on('error', err => logger.error(err));
app.listen(port);
```

稳定性

首先，你要有单元测试

编写可测试的代码

```
app.use(onerror);          // 异常处理
app.use(trace);            // 打印 trace log
app.use(meta);             // 记录时间等信息
app.use(detector);         // 判断终端类型
app.use(cacheControl);     // 设置 CDN 缓存头
app.use(backup);           // 备份容灾
app.use(render);           // 渲染
```

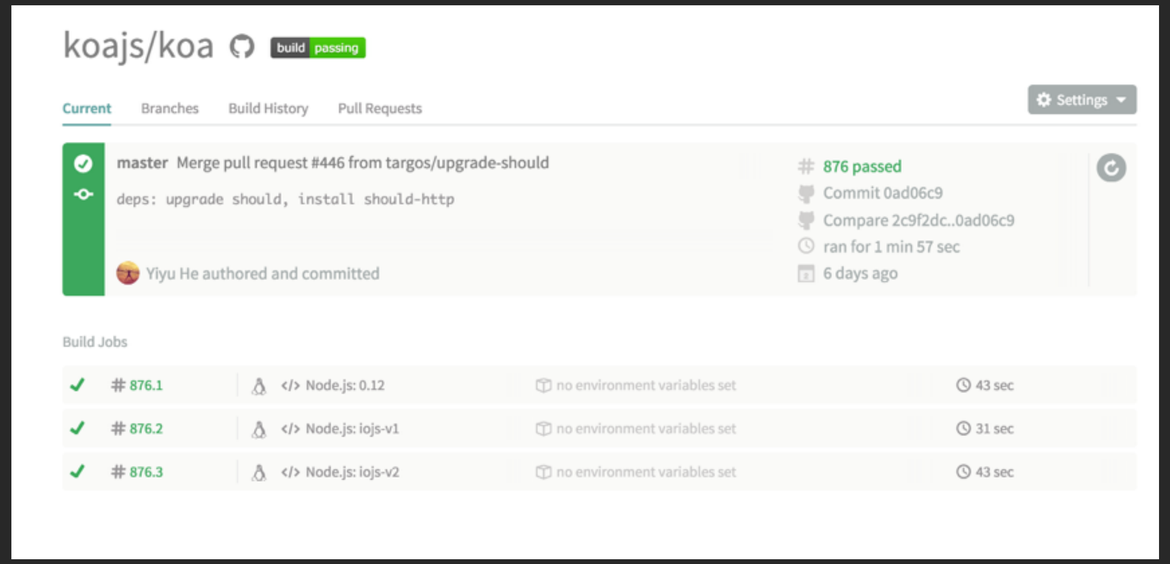
```
├── backup.test.js
├── cacheControl.test.js
├── detector.test.js
├── meta.test.js
├── onerror.test.js
├── render.test.js
└── trace.test.js
```

测试覆盖率与持续集成

- 让测试覆盖率告诉你哪里可能存在 bug

```
============================= Coverage summary =============================
Statements   : 98.7% ( 1445/1464 ), 12 ignored
Branches     : 93.21% ( 563/604 ), 7 ignored
Functions    : 100% ( 68/68 ), 1 ignored
Lines        : 99.15% ( 1405/1417 )
============================================================================
```

- 持续集成察觉每次代码变更引入的潜在风险



Cluster

- Master 只做进程管理
- worker 异常退出后自动重启（cfork)
- http 服务优雅退出 (graceful)

监控关键指标，对异常告警

日志

- 所有的请求都有 accesslog 追踪
- 所有的异常统一记录，并分类输出
- 依赖系统调用 tracelog，记录请求和响应情况

监控

- 系统状态：CPU / 内存 / Load / 磁盘空间 / …
- 服务状态：响应时间 / QPS / …
- 依赖服务状态

告警

- 系统 / 服务状态异常：RT 增高 / Load 增高 / …
- 依赖异常：调用失败 / 调用超时 / …
- 服务异常：异常日志

- 快速感知（告警）

- 快速感知（告警）
- 快速定位（监控、日志）

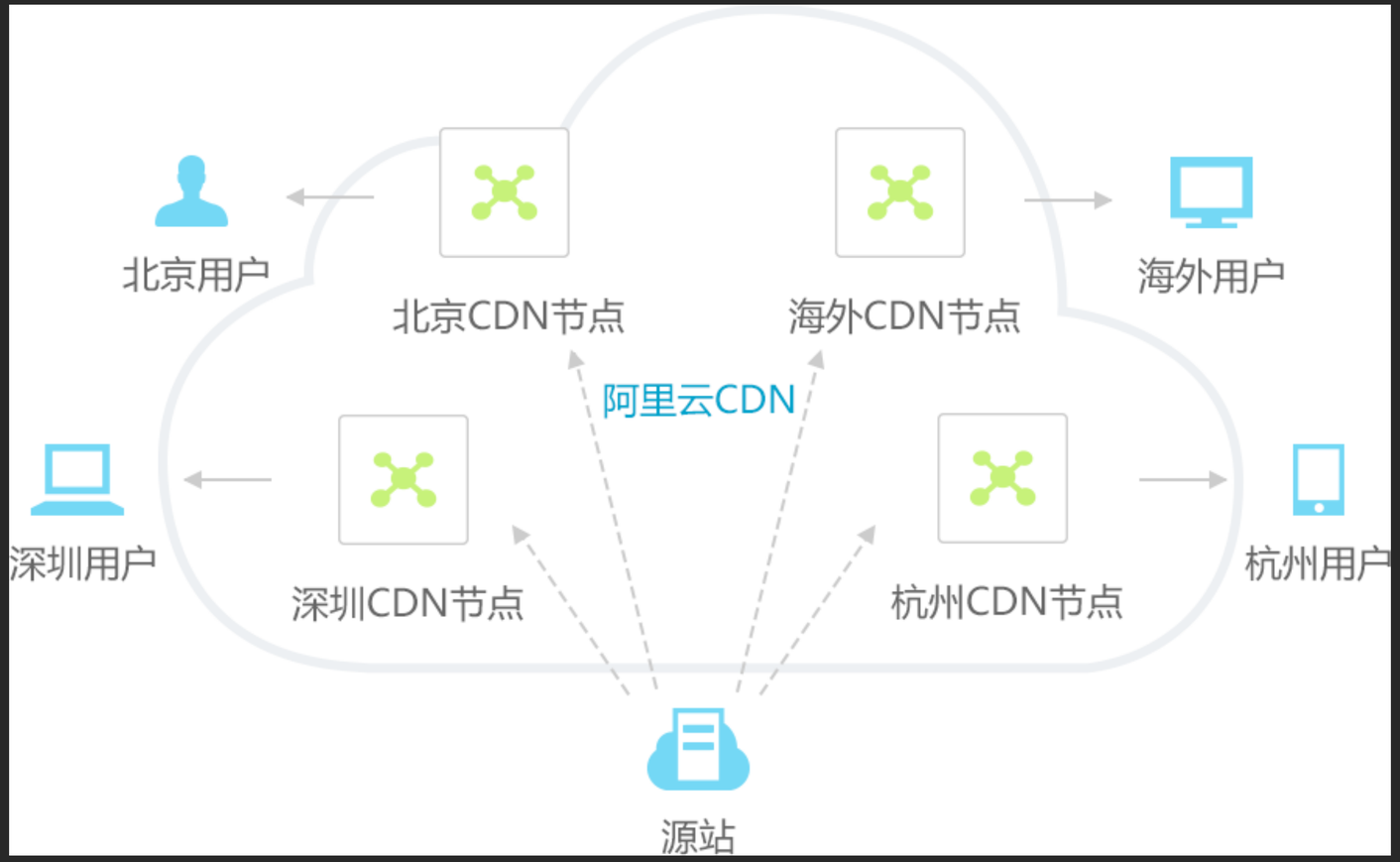
- 快速感知（告警）
- 快速定位（监控、日志）
- 目的是快速恢复

双十一 === DDos

合理利用 CDN

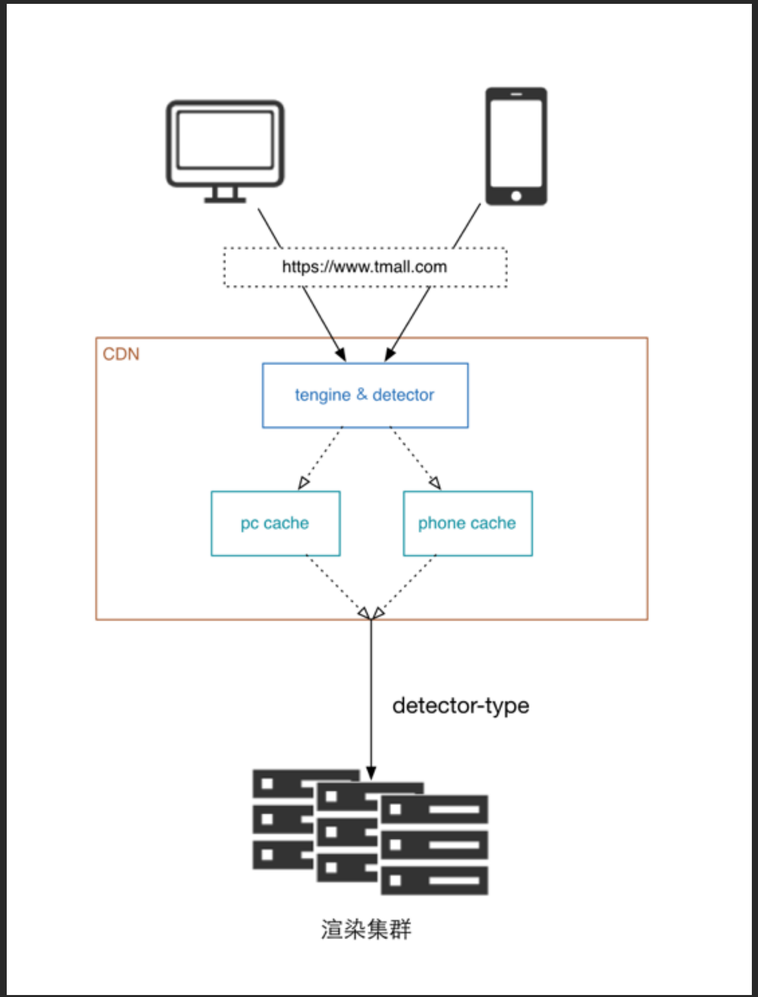
- 绝大部分的页面不需要每次请求都重新渲染
- 解决双十一流量高峰的高并发问题
- 大大节省机器成本

合理利用 CDN


北京用户
北京CDN节点
海外CDN节点
海外用户
阿里云CDN
深圳用户
深圳CDN节点
杭州CDN节点
杭州用户
源站

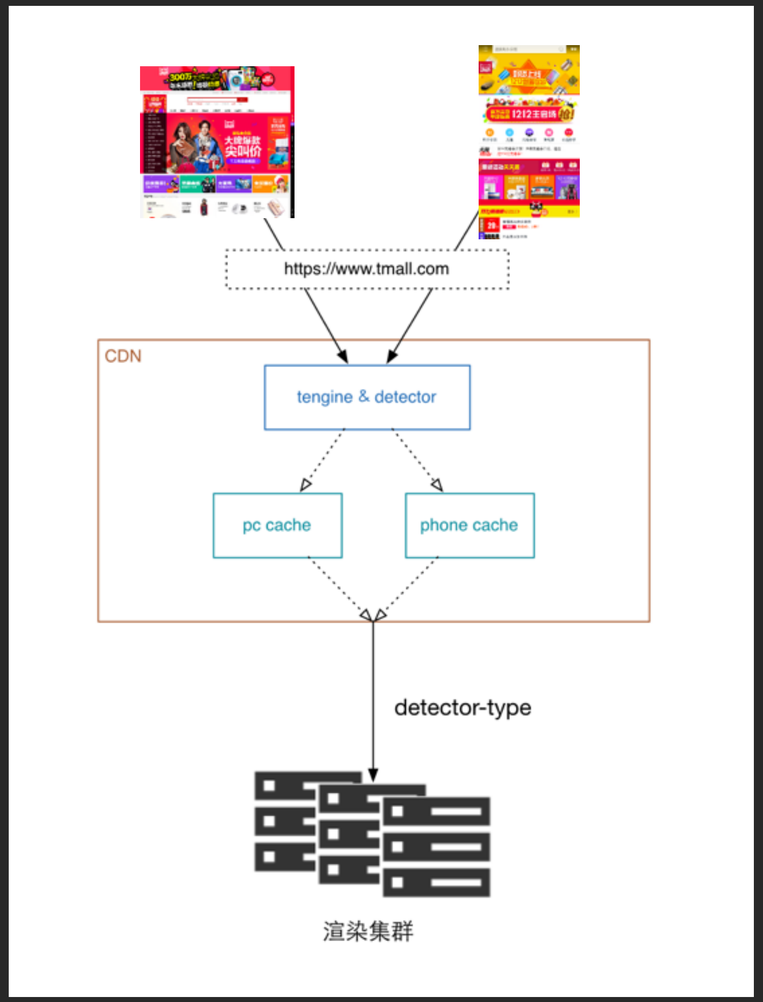
基于 CDN 的 URL 统一

- 不同终端请求相同 url
- CDN 识别设备类型
- 渲染服务返回不同页面



基于 CDN 的 URL 统一

- 不同终端请求相同 url
- CDN 识别设备类型
- 渲染服务返回不同页面



容灾与预案

Impossible is nothing

- 依赖服务宕机
- 机房断电、断网
- 光缆被挖断

消灭单点、弱化依赖

- 异地双机房部署
- OSS 双节点主备容灾
- 数据、模板磁盘文件容灾

预案自动化

- CDN 回源健康检查，异常自动切换
- OSS 健康检查，异常自动切换
- 系统 Load 过高自动切换静态备份
- 渲染异常自动切换静态备份


Q&A
time