## 冗余换性能-从Backbone的triggerEvents说开了去

Backbone是一个优秀的前端MVC库，它的代码质量必定可靠。阅读过程中发现了函数triggerEvents有点意思，初看会以为一些代码是多余的。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | var triggerEvents = function(events, args) {   var ev, i = -1, l = events.length, a1 = args[0], a2 = args[1], a3 = args[2];   switch (args.length) {     case 0: while (++i < l) (ev = events[i]).callback.call(ev.ctx); return;     case 1: while (++i < l) (ev = events[i]).callback.call(ev.ctx, a1); return;     case 2: while (++i < l) (ev = events[i]).callback.call(ev.ctx, a1, a2); return;     case 3: while (++i < l) (ev = events[i]).callback.call(ev.ctx, a1, a2, a3); return;     default: while (++i < l) (ev = events[i]).callback.apply(ev.ctx, args);   }  }; |

这段是Backbone.Events中的派发事件的核心函数。执行顺序大概是

1. 把事件处理函数依次取出执行
2. 根据第二个参数args的长度做分支处理
3. args长度为3及以下时使用call，长度3以上时使用apply

如果把call的都去掉，直接使用apply，那么程序的逻辑正确性没有问题。如下

|  |  |
| --- | --- |
| 1  2  3  4 | var triggerEvents = function(events, args) {      var ev, i = -1, l = events.length;      while (++i < l) (ev = events[i]).callback.apply(ev.ctx, args);  }; |

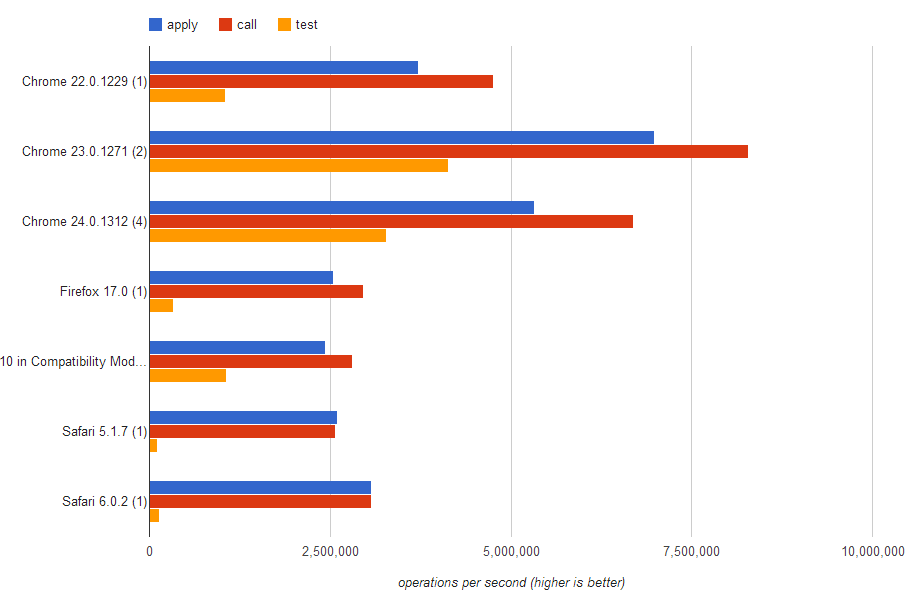
代码还精简了。Backbone却加上了以上使用call这些代码，来看注释

|  |  |
| --- | --- |
| 1  2  3 | // A difficult-to-believe, but optimized internal dispatch function for  // triggering events. Tries to keep the usual cases speedy (most internal  // Backbone events have 3 arguments). |

虽然难以相信，但却是为了性能考虑。多数Backbone内部方法触发事件时会传3个参数，即会使用call（而非apply）。

按此推理，call的性能较apply要高。搜关键字 "call apply performance"， 貌似印证了这个说法

这是[jsperf.com](http://jsperf.com/call-apply/3)上的一个测试



从测试结果可以看出，多数浏览器中call的性能都好于apply，只有Safari5和Safari6中两者性能差不多。因此，Backbone是用冗余的代码换取了运行时的性能。

注：版本Backbone 1.0

相关：

<http://jsperf.com/call-apply/3>

[what-is-the-difference-between-call-and-apply](http://stackoverflow.com/questions/1986896/what-is-the-difference-between-call-and-apply)

[why-are-call-and-apply-slower-than-a-direct-function-call-in-javascript](http://stackoverflow.com/questions/8184466/why-are-call-and-apply-slower-than-a-direct-function-call-in-javascript)