



AWS专线实战：高频量化，不是拼算力，而是拼延迟

别再拼算力了！

高频交易的真正瓶颈在这里

这几年帮高频团队做系统优化，我最常听到的一句话就是——

“是不是我们服务器算力不够？”

但真相往往相反，高频交易拼的，从来不是算力，而是——延迟的极限。

香港团队的实盘困局

前阵子一个香港团队找我，策略模型写得很漂亮，回测收益也稳。



可一上实盘，成交率始终提不上去。

延迟一查：撮合来回 300 多微妙，一遇行情放量，最高飙到 700 微秒。

他们一开始还以为是算法太慢，

结果一看架构，我直接说——

“不是策略问题，是路径走错了。”

最大的问题：走了公网链路

他们的系统部署在 AWS 上，却没注意网络配置。

默认的 AWS 网络走公网，是共享链路。

报单从 AWS 到交易所，要经过五六跳路由，抖动动辄几十微妙。

那天我们抓包时，发现每笔订单往返的丢失时间在 200 μs 左右，

光 retransmit 重传就占了 10%。



我当场建议：

“别再调代码了，先上专线。”

第一步：Direct Connect 专线直连

我们帮他们开通 AWS Direct Connect，

从东京区直连交易所的 Equinix 机房。

BGP 路由固定 AS，关闭ECMP，并启用BFD，让所有流量走同一条物理路径，彻底消除乱跳。

这一步完成后，延迟从 $310\mu s \rightarrow 150\mu s$ 。

客户看着延迟图几乎砍半的变化，当场沉默了几秒，只说了一句：

“原来差距真的能量化。”

但真正的提升，
在细节里



专线只是起点，后面的优化才是决定性。

我们又做了几项关键调优：

实例选型：从 C7i.metal 换成 i4.metal 裸金属，彻底去掉虚拟化层干扰；

中断优化：禁用 irqbalance，把网卡中断绑到固定核心；

TCP 栈：启用 SO_BUSY_POLL，让线程主动轮询，不再等中断；

内核调优：关闭省电、锁主频、开启 HugePages；

PTP 时间同步：硬件级时间戳，误差控制在 $\pm 2\mu s$ 内。

整套调完，重新跑端到端延迟测试：

平均 $98\mu s$ ，波动不超过 $\pm 5\mu s$ 。

那天他们在 Zoom 上看着延迟曲线变成一条“直线”，

团队一片安静。

那不是兴奋，而是一种“终于知道真瓶颈在哪”的释



然。

延迟优化的底层逻辑

做多了这种项目，我越发觉得：

延迟，不是堆配置，而是理解物理。

CPU再强、内存再大，

如果你的链路穿公网、时钟不同步、NUMA跨节点
——延迟照样飘。

很多团队一年花几十万租顶配机器，

却忽略了路由与内核。

你真正要的，不是最贵的机器，而是最干净的路径。

数据不会骗人

这个团队后来自己做了个对比测试：

公网模式下：10万笔撮合，延迟均值 $320\mu s$ ；

优化后架构：均值 $101\mu s$ ；



成交成功率提升 13%，

单月收益多了 6%，成本几乎没变。

高频的世界很残酷——

慢 50 微秒，你就看不到价差。

真正的竞争力，不在策略逻辑，而在信号落地的每一个微秒。

你的系统，现在延迟在哪个区间？

跑在 AWS 的，是 Direct Connect 还是公网？

时间同步靠 NTP 还是 PTP？

NUMA 跨节点了吗？

如果你也在做高频量化，

不确定瓶颈到底在哪，

可以来私信聊聊。

我专注做高频量化低延迟优化已经七年，帮过不少团队从“看起来很快”，变成“真快”。有时候，改对一条链路，就能换来整段盈利曲线。