



0 行业观察 bservation

12 网络传输状态数学建模和实证检验

13 浅谈非易失性存储器在现代交易系统中的应用

14 雪球期权价格计算的 FPGA 实现

网络传输状态数学建模和实证检验

郑凡 / 方正证券股份有限公司 / zhengfan@foundersc.com
李鑫 / 中华女子学院金融系 / differeasy@hotmail.com



对复杂网络条件下的网络数据传输建立可检验的数学模型，根据模型做出预测，并通过实际观测得到的网络监控数据对预测进行检验。最终根据模型形成对网络数据传输能力和状态的量化定义。

一、概述

从分析数据成因入手，结合工程和物理设备对应的概念，建立传输状态模型，并根据实际采样数据进行实证检验，最终建立一个理论可信、操作可行的网络数据传输能力和状态的量化定义及其评价标准。

二、提出问题

通过网络进行数据传输，是信息、通信等工程技术领域常见的应用场景。是否可以，以及如何量化的、确定性的描述网络数据传输能力和状

态，是工程技术研究非常关心的问题，对工程技术实践具有指导意义。可以认为，量化的、确定性的、可检验的网络数据传输能力和状态，为信息、通信工程领域问题建立了讨论的基础和比较的原点。

前序论文《行情数据专线传输的分析和调优》为行文方便使用了一种对网络传输状况的定义，即：“理想数据传输状态”；然而限于篇幅，上述论文并没有说明“理想连接状态”的具体定义内容、检验标准以及在工程实践上的意义。回顾可知：定义“理想数据传输状态”的目的是描述一种网络传输状况，在这种状态下网络传输可以保持稳定且比较高的数据吞吐量。所以本质上

“理想数据传输状态”就是本文希望提供的关于网络传输问题讨论的原点。

通常解决此类复杂问题，需要结合可观测的数据，形成可检验的数学模型。并最终根据这个模型来预测和指导解决问题。本文希望循此思路，对复杂网络条件下的网络数据传输建立可检验的数学模型，结合可观测的网络监控数据，对网络数据传输能力和状态做出明确定义。

三、分析问题

当前需要解决的问题是是否可以、以及如何建立“网络传输能力和状态”和在传输端点上收集到的网络监控信号数据之间的对应关系。步骤如下：

1. 分析问题对应的物理根源；
2. 找到可用的网络监控信号数据；
3. 建立数据模型；

（一）问题的物理原型

如图 1 所示，长距离、跨广域网的数据传输很难做到直接的、端到端的通信，而是需要通过物理链路上的若干中继设备，如交换机、路由器、网关等等进行消息中转。虽然这些中继设备的设计目的不同，但是其工作模式类似：

1. 从上游数据连接入口接收数据报文；
2. 根据数据报文中的特定数据字段或者内置固化配置，通过算法找到下游数据连接出口；
3. 向下游数据连接出口发送数据报文；

显而易见的是，这些工作步骤具有时间成本。进一步考虑到端点和中继节点之间、以及中继节点之间的网络传输能力差异，可知这些中继设备上极可能出现消息堆积问题。为了提高中继设备的工作弹性，设备供应商和网络运营商为这些中继设备建立了消息缓存能力，以应对传输数据量的剧烈波动和爆发性增长。这样的设计和实现能够以增加一定传输时间为代价，有效地降低传输数据量剧烈波动情况下的丢包可能性。但是这样的物理链路实现也给技术人员测量网络传输能力、监测网络连接状态带来了困难。

我们根据上述网络物理设备工作原理建立模型。为了建立模型的简单起见，我们将整条物理链路想象为一个带有缓存能力的中继设备。其传输能力边界状态包括：

- 中继设备不需要启用消息缓存可支持的最大数据传输量 V_L ；
- 中继设备启用消息缓存可支持的最大传输量 V_H ；

“网络传输能力和状态”的量化定义，本质上是要找到上述 V_L 和 V_H 。

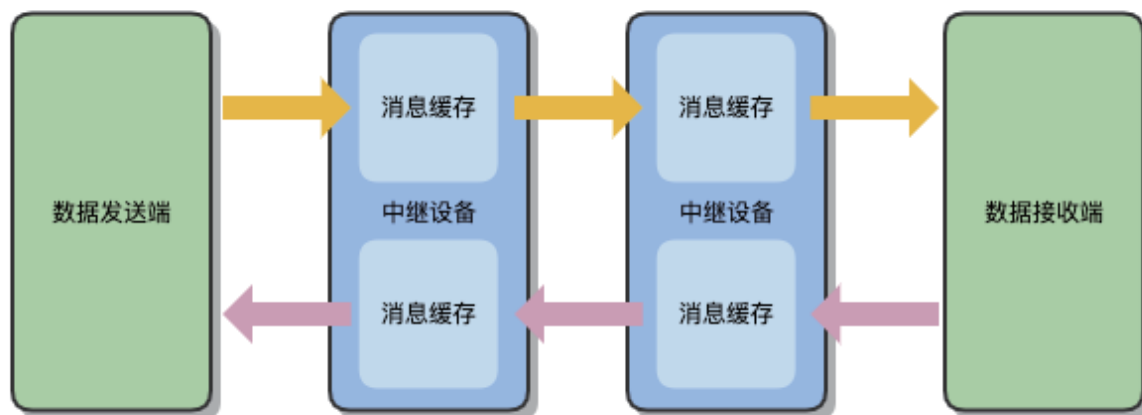


图 1：物理传输链路示意图

（二）选取分析对象数据

传输端点可以收集到的网络监控数据可以分为三种类型：

1. 客观记录值，通常是对外部网络状况的实际数据记录，包括 `rtt/rttvar`、`segs_out/segs_in` 等；
2. 协商或默认配置，通常是根据临时或默认约定形成的数据，包括 `mtu`、`mss` 等；
3. 算法生成值，通常是上述两种类型数据通过一定算法得出的结果，包括 `cwnd`、`ssthresh` 等；

很直观的是，第二种类型数据是相对静态的配置数据，不能对网络传输状态产生动态反应；第三种类型数据是其他类型数据的计算结果，具有一定的主观因素；只有第一类客观记录数据满足可观测信号条件，可以作为候选。

进一步分析第一类数据可以发现，其中的大部分数据如 `segs_out/segs_in` 是对传输端点协议栈执行、数据接收和发送总量的记录，可以作为数学模型中数据传输量的参考。`rtt/rttvar` 作为时间间隔记录，其数据经历了从一个传输端点出发经过网络传输到达彼端，彼端应答再次通过网络传输回到本端的全过程。

Linux 命令手册中对 `rtt/rttvar` 的介绍是：“`rtt` 是往复通信延迟 (Round-Trip Time, RTT) 的记录均值，`rttvar` 是 `rtt` 值的平均偏差 (Mean Deviation)。”

可知，`rtt/rttvar` 数据具备如下性质：

1. 独立于端点逻辑和状态；
2. 数据采样之间独立；
3. 可以动态体现网络状况及其变化；
4. 与单位时间内数据传输量强相关；

综合考虑上述因素，决定选取 `rtt/rttvar` 作为分析对象数据。

同时需要考虑到，数据传输量可能会受到传输端点算法的制约，比较典型的有拥塞退避算法等；这些算法往往是根据 `rtt` 数据对传输量产生调整的。所以在建模过程中需要针对此类算法调整的边界状态进行必要过滤和修正。

（三）建立数学模型

考察 RTT 数据记录所代表的物理意义。假设在此数据传输过程中共记录 n 个实际 RTT 数值，则第 i 次 RTT 记录应可以表达如下：

$$RTT_i = FixedEffort_i + FloatEffort_i + \delta_i; i \in [1, n]$$

其中 δ_i 表示第 i 次 RTT 记录中包含的随机时间开销记录部分。由于已经将其他可归因影响因素单列，不妨假设 δ 是在整个网络传输过程中的独立同分布、具备相同期望和方差的随机变量。

$FloatEffort_i$ 是 i 次 RTT 记录中包含的设备缓存造成的浮动时间开销记录部分。

$FixedEffort_i$ 是 i 次 RTT 记录中包含的传输固定时间开销；其表示链路中线路、设备传输的固定时间开销。所以此项数值应为常量。即满足如下恒等式：

$$E(FixedEffort) \equiv FixedEffort_i \equiv FixedEffort_j; \forall i, j \in [1, n]$$

通过运维手段获得的网络数据传输时间采样 `rtt`，实际上是将上述 RTT 记录值按照采样时间间隔分组，形成一系列子序列 $\{RTT\}_1, \{RTT\}_2, \dots$ ；一个 `rtt` 采样值表示其所在采样间隔对应的 RTT 记录值子序列的均值。第 m 次 `rtt` 采样数据可以表示为：

$$rtt_m = \overline{RTT}_m = \overline{FixedEffort}_m + \overline{FloatEffort}_m + \overline{\delta}_m$$

而 `rttvar` 采样值表示其所在采样间隔对应的 RTT 记录值子序列的平均偏差。假设第 m 个采样时间间隔对应 $\{RTT\}_m$ 子序列中共有 k_m 个记录值，则 $\{RTT\}_m$ 子序列对应的 `rttvar` 采样数据所代表的平均偏差为：

$$\begin{aligned} rttvar_m = MD(rtt_m) &= \frac{\sum_{i=1}^{k_m} |RTT_i - \overline{RTT}_m|}{k_m} \\ &= \frac{\sum_{i=1}^{k_m} |FloatEffort_i + \delta_i - (\overline{FloatEffort}_m + \overline{\delta}_m)|}{k_m} \\ &= \frac{\sum_{i=1}^{k_m} |FloatEffort_i - \overline{FloatEffort}_m|}{k_m} + \frac{\sum_{i=1}^{k_m} |\delta_i - \overline{\delta}_m|}{k_m} \\ &= MD(\{FloatEffort\}_m) + MD(\{\delta\}_m) \end{aligned}$$

换一个角度来观察 $FloatEffort$ 。根据定义可知 $FloatEffort$ 是假设模型中链路中继设备消息缓

存带来的传输延迟，不妨假设其是关于中继设备消息缓存积压状态 $Accumulation_m$ 的线性函数：

$$\{FloatEffort\}_m = D \times Accumulation_m$$

其中 D 是处理单位缓存积压的时间成本。 $E(\{FloatEffort\}_m)$ 反映的是，在第 m 次 rtt 采样时间间隔中，链路中继设备缓存使用状况、消息堆积程度的期望值：

$$E(\{FloatEffort\}_m) = D \times E(Accumulation_m)$$

$MD(\{FloatEffort\}_m)$ 反映的是第 m 次 rtt 采样时间间隔中，链路中继设备的缓存使用状况、消息堆积程度的波动范围：

$$MD(\{FloatEffort\}_m) = D \times \sum (Accumulation_m - E(Accumulation_m))$$

进一步考虑链路中继设备缓存积压状态与数据流量的关系。假设第 m 个采样间隔中的单位数据传输量为 $\{Volume\}_m$ ，有：

$$\{Volume\}_m = segs_out_m - segs_out_{m-1} = d_m(segs_out)$$

很直观的是，消息堆积状况 $Accumulation_m$ 可以表达为单位数据传输量的函数：

$$Accumulation_m = g(d_m(segs_out))$$

结合前文对物理网络和中继设备的假设，分别考虑 $Accumulation_m$ 和 $\{FloatEffort\}_m$ 在 $d_m(segs_out)$ 小于 VL 、介于 VL 和 VH 之间，以及大于 VH 三种传输状态下的变化。

1、 $d_m(segs_out)$ 小于 VL ：

在 $d_m(segs_out)$ 小于 VL 状态下，不需要启用中继设备消息缓存；则此时消息堆积状况为：

$$Accumulation_m = g(d_m(segs_out)) = 0, d_m(segs_out) \leq VL$$

缓存时间开销的期望值 $E(FloatEffort)$ 较低：

$$E(\{FloatEffort\}_m) = D \times 0 = 0, d_m(segs_out) \leq VL$$

由此可以推论出，在此传输状态下的 RTT 采样均值较小：

$$rtt_m = \overline{\{RTT\}_m} = \overline{\{FixedEffort\}_m} + \overline{\{\delta\}_m}$$

RTT 采样均值的平均偏差比较小、稳定：

$$rttvar_m = MD(\overline{\{RTT\}_m}) = MD(\{\delta\}_m)$$

需要注意的是，TCP 慢启动 (Slow Start) 控制策略会导致在低数据流量区间内单位时间内发

送数据的快速增长（通常是指指数级增长），这有可能导致在平均传输量小于 VL 区间内 $segs_out$ 的采样数据稀疏。

2、 $d_m(segs_out)$ 介于 VL 和 VH 之间：

在 $d_m(segs_out)$ 介于 VL 和 VH 之间状态下，需要启用中继设备消息缓存。并且缓存占用情况会随着平均传输量上升而增长：

$$Accumulation_m = d_m(segs_out) - VL, VL < d_m(segs_out) \leq VH$$

此状况下， $Accumulation_m$ 正比于 $d_m(segs_out)$ ；缓存造成时间开销 $\{FloatEffort\}_m$ 及其期望值：

$$\{FloatEffort\}_m = D \times (d_m(segs_out) - VL)$$

由此可以推论出，在此传输状态下的 RTT 采样均值为：

$$rtt_m = \overline{\{RTT\}_m} = FixedEffort + D \times (d_m(segs_out) - VL) + \overline{\{\delta\}_m}$$

可知在此传输状态下 rtt 采样值随 $segs_out$ 采样值线性增长，伴随随机波动。这符合一般对数据传输状态的直观想象。

另一方面，缓存空间的剩余部分随着消息缓存占用的增长而缩小。而事实上缓存空间的剩余部分的大小决定了缓存积压偏差的范围的上限。单位传输量 $d_m(segs_out)$ 对应缓存积压偏差的范围是 $[0 \sim (VH - d_m(segs_out))]$ 之间。据此得到：

$$MD(Accumulation_m) = \min(MD(d_m(segs_out) - VL), VH - d_m(segs_out))$$

缓存造成时间开销平均偏差为：

$$MD(\{FloatEffort\}_m) = D \times \min(MD(d_m(segs_out) - VL), VH - d_m(segs_out))$$

$$= D \times \min(d_m(segs_out) - E(d_m(segs_out)), VH - d_m(segs_out))$$

最终可以得到：

$$rttvar_m = MD(\overline{\{RTT\}_m}) = MD(\{FloatEffort\}_m) + MD(\{\delta\}_m) = D \times \min(d_m(segs_out) - E(d_m(segs_out)), VH - d_m(segs_out)) + MD(\{\delta\}_m)$$

在 $VL < d_m(segs_out) \leq VH$ 条件下一般假设：

$$E(d_m(segs_out)) = \frac{VH + VL}{2}$$

可推论出， $rttvar_m$ 在 $VL \leq d_m(segs_out) < \frac{3 \times VH + VL}{4}$ 随 $d_m(segs_out)$ 线性增长；在 $\frac{3 \times VH + VL}{4} \leq d_m(segs_out) < VH$ 随 $d_m(segs_out)$ 线性减小；即 $rttvar_m$ 在 $d_m(segs_out) = \frac{3 \times VH + VL}{4}$ 时存在最大值。这与一般对数据传输状态的直观想象相悖。

可以进一步推论，随着数据流量逐步逼近链路平稳可靠传输的上限，链路设备缓存被占用殆尽， $MD(\{\text{FloatEffort}\}_m)$ 将趋近于 0。则此时的 $rttvar$ 采样值：

$$\lim_{d_m(segs_out) \rightarrow VH} (rttvar_m) = MD(\{\delta\}_m)$$

3、 $d_m(segs_out)$ 大于 VH：

在 $d_m(segs_out)$ 大于 VH 状态下，需要启用中继设备消息缓存，并且消息缓存满负荷工作。有非常高比例的传输消息因为超出设备缓存的能力而被丢弃导致丢包，可以认为此时的传输时间成本接近 $+\infty$ ：

$$\text{FloatEffort}(d_m(segs_out)) = +\infty, VH < d_m(segs_out)$$

这将最终导致数据发送端启动拥塞退避算法降低传输量。所以应不可能稳定持续地观测到平均传输量大于 VH 状态。有可能出现偶发非持续的平均传输量大于 VH 状态，其对应 $rtt/rttvar$ 采样值应非常分散、不可估计。

四、根据模型提出预测

根据已建立的模型、计算和推论，给出不同传输状态下 $d_m(segs_out)$ 和 $rtt/rttvar$ 的相关预测如下：

1. 数据平均流量存在上限阈值，大于此阈值的

采样值分布呈快速衰减，对应 $rttvar$ 采样分布非常分散；

2. 在数据平均流量较小时，存在单位时间流量采样数据相对稀疏区间；

3. 在数据平均流量较大时 rtt 采样均值与平均流量增加呈正比关系；

4. 数据平均流量较小阶段 rtt 采样均值小于数据流量较大时段 rtt 采样均值；

5. 在数据平均流量较大时 $rttvar$ 采样值在存在峰值区域；

五、预测检验

(一) 选取检验数据：

待考察的“网络数据传输能力和状态”可能与所传输数据对象的形态特征有关。为此决定使用真实工作数据而非模拟测试数据作为网络传输数据的选取来源。检验数据使用 2021 年 3 月 3 日 - 2021 年 3 月 8 日（均为交易日）间深圳交易所全日行情数据到湖南省长沙市证券大厦机房的专线传输的网络状态监控数据。数据采样间隔为 1 秒；采样数据包包含采样时间、 $rtt/rttvar$ 、 $segs_out/segs_in$ 、 $cwnd$ 等。

每日采样数据时间范围和采样点总数如表 1 所列。

观察表 1 可以发现，在不同日期内数据采样的时间范围、采样总数等都比较接近；表明采样环境质量良好，具备进一步分析的基础。

(二) 检验数据分组、均值计算和过滤：

- 1、检验数据分组的原因：

表 1

| | 2021-03-03 | 2021-03-04 | 2021-03-05 | 2021-03-08 |
|-------|------------|------------|------------|------------|
| 首记录时间 | 08:45:23 | 08:45:22 | 08:45:27 | 08:45:24 |
| 末记录时间 | 17:00:09 | 17:00:10 | 17:00:10 | 17:00:10 |
| 采样点总数 | 26464 | 26324 | 26536 | 26284 |

上述模型预测中涉及到对不同 d_m (segs_out) 采样值 $rtt/rttvar$ 均值的比较和计算。考虑到 d_m (segs_out) 采样数据取值范围很广且分布极度不均匀，直接对单个 d_m (segs_out) 对应的 $rtt/rttvar$ 求均值可能不具备统计意义。最终决定采用 d_m (segs_out) 采样值分组方式处理数据。在此约定 d_m (segs_out) 采样分组的选取方法和对应分组内 $rtt/rttvar$ 均值的计算方法如下：

以 d_m (segs_out) 步进 10 作为一个区间，求其中 $rtt/rttvar$ 的算数平均作为均值；

2、数据过滤的原因：

1. 拥塞退避发生时刻的采样；拥塞控制算法对 $rtt/rttvar$ 采样数据敏感，并会据此调整数据发送速率，从而导致某些采样点数据被传输端算法“污染”。从 TCP 慢启动的工作原理中可以发现，其传输控制临界状态跳变（拥塞退避）时对 $segs_out$ 的影响最大；有必要对此部分数据进行过滤；

2. 单位时间内 d_m (segs_out) 为 0 的采样；单位传输量为 0 的采样点不具备数据观测意义；

3. d_m (segs_out) 分组中采样值少于 10 个的分组；步进 10 的分组区间中采样少于 10 个不具备统计意义；

3、数据过滤的方法：

对于数据样本的调整包括：

1. 对所有样本，剔除掉所有 d_m (cwnd) 为负值的前一个采样数据点（TCP 拥塞退避算法触发点）；

2. 对所有样本，剔除掉所有吞吐量（ $segs_out/segs_in$ ）为 0 的采样点；

3. 对分组均值采样，剔除掉所有 d_m (segs_out) 分组中采样值少于 10 个的分组；

4、数据过滤的影响

经过对全样本进行上述方法 1 和方法 2 过滤后，样本点数量情况如表 2 所列。

观察表 2 可以发现，在不同日期内：符合过滤条件的采样点体量基本一致，总体采样占比基本一致；过滤后保留采样数据点数量约为采样点总数量 75%；

经过对上述过滤后数据进行分组、以及方法 3 过滤情况如表 3。

观察表 3 可以发现，在不同日期内：采样点单位时间传输量分组数量基本一致；符合过滤条件的采样点体量基本一致，总体采样占比基本一致；过滤后保留采样数据点数量约为采样点总数量 70%；

（三）单位时间传输量 d_m (segs_out) 统计分布形态：

对过滤后每日监控数据中单位时间传输量

表 2

| 样本数 | 2021-03-03 | 2021-03-04 | 2021-03-05 | 2021-03-08 |
|-----|------------|------------|------------|------------|
| 过滤前 | 26464 | 26324 | 26536 | 26284 |
| 过滤后 | 19838 | 19582 | 19597 | 19615 |
| 剔除 | 6626 | 6742 | 6939 | 6669 |
| 保留率 | 74.96% | 74.39% | 73.85% | 74.63% |

表 3

| 样本数/分组数 | 2021-03-03 | 2021-03-04 | 2021-03-05 | 2021-03-08 |
|---------|--------------|--------------|---------------|---------------|
| 过滤前 | 19838/703 | 19582/694 | 19597/703 | 19615/690 |
| 过滤后 | 18727/411 | 18644/422 | 18562/425 | 18620/425 |
| 剔除 | 1111/292 | 938/272 | 1035/278 | 995/265 |
| 保留率 | 94.4%/58.46% | 95.2%/60.81% | 94.72%/60.46% | 94.93%/61.59% |
| 总保留率 | 70.76%/- | 70.82%/- | 69.95%/- | 70.47%/- |

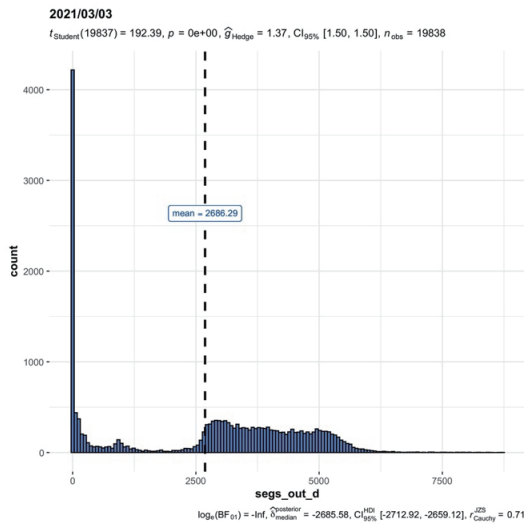


图 2：2021 年 03 月 03 日统计

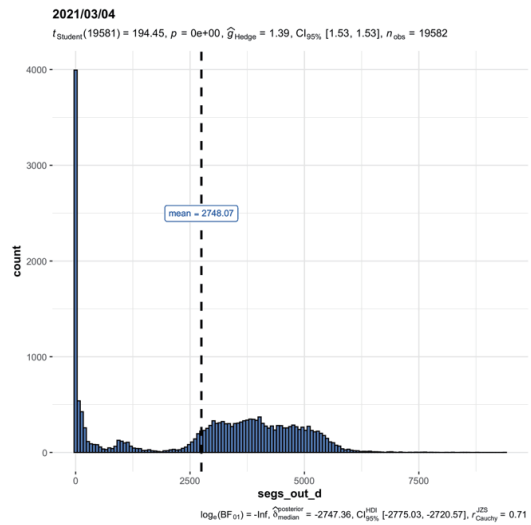


图 3：2021 年 03 月 04 日统计

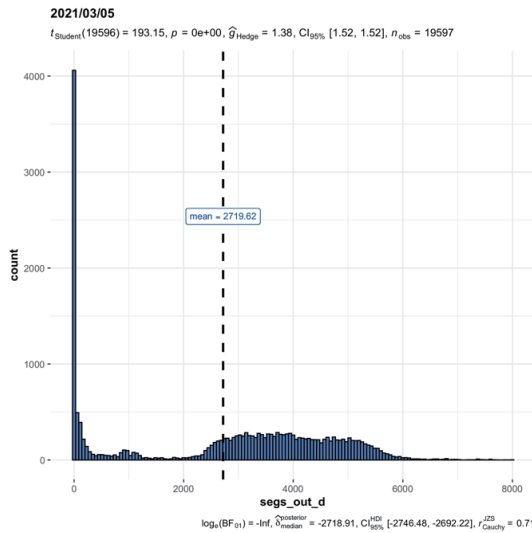


图 4：2021 年 03 月 05 日统计

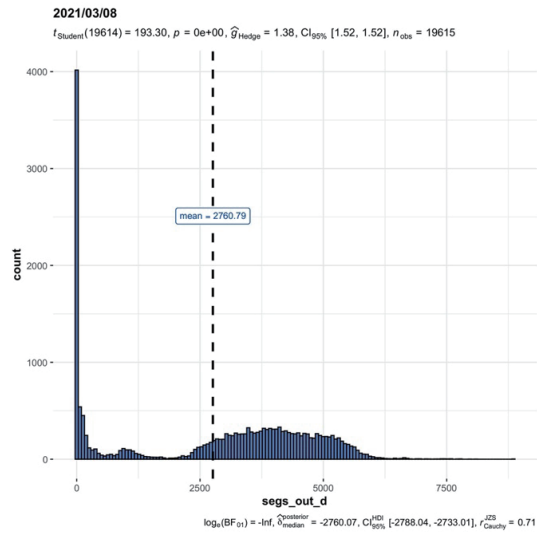


图 5：2021 年 03 月 08 日统计

d_m (segs_out) 观测值进行统计，得到一系列柱状图如图 2—图 5。

归纳上述日期单位时间传输量观测统计值分布情况如表 4。

观察上述分布情况可知，不同日期内单位时

间传输量观测值：

1. 分布范围基本一致，跨度较大；
2. 统计值范围基本一致，跨度较大；
3. 统计值分布极度不均匀；
4. 统计值分布形态极为接近：

表 4：单位时间传输量观测统计值分布情况

| 日期 | 抽样值范围 | | 抽样值统计 | | | |
|------------|-------|------|-------|-------------|------|---------|
| | 最小值 | 最大值 | 最小值 | 对应区间 | 最大值 | 对应区间 |
| 2021-03-03 | 0 | 8719 | 2 | [8500-8750] | 5341 | [0-250] |
| 2021-03-04 | 0 | 9375 | 0 | [9100-9360] | 5262 | [0-260] |
| 2021-03-05 | 0 | 7996 | 3 | [7800-8060] | 5306 | [0-260] |
| 2021-03-08 | 0 | 7591 | 0 | [8380-8580] | 5345 | [0-260] |

- a) 在首个区间 [0-250/260] 之间有统计极大值；
 - b) 在 [250/260-2000/2080] 区间统计值较低；
 - c) 在 [2000/2080-2500/2600] 区间经历快速增加；
 - d) 在 [2500/2600-5720/5750] 区间有连续稳定高统计值；
 - e) 在 [5720/5750-6240/6250] 区间经历快速衰减，其后区间数据统计极为稀疏；
5. 稳定高统计值区间 [2000/2080-5720/5750] 内分布形态有较微小差别；

经分析可得到如下结论：

1. 不同日期内单位时间传输量统计统计具有一致性、稳定性；具备进一步分析总结的基础；
2. 稳定高统计值区间 [2500/2600-5720/5750] 内分布形态有较微小差别，或来源于市场参与度的弹性变化和热点事件；
3. 某些区间的数据采样点非常稀疏，有必要采用分组采样或者采样值过滤手段保证数据的统计意义；
4. 在 [250/260-2000/2080] 区间, 存在 d_m (segs_out) 采样数据稀疏区间，符合模型预测 2；

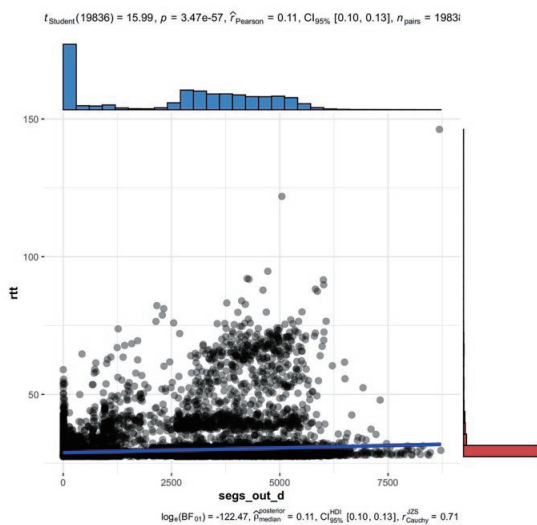


图 6: 20210303 segs_out-rtt

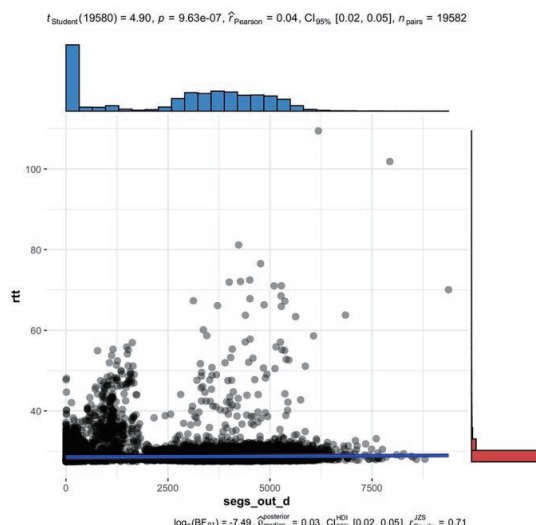


图 7 : 20210304 segs_out-rtt

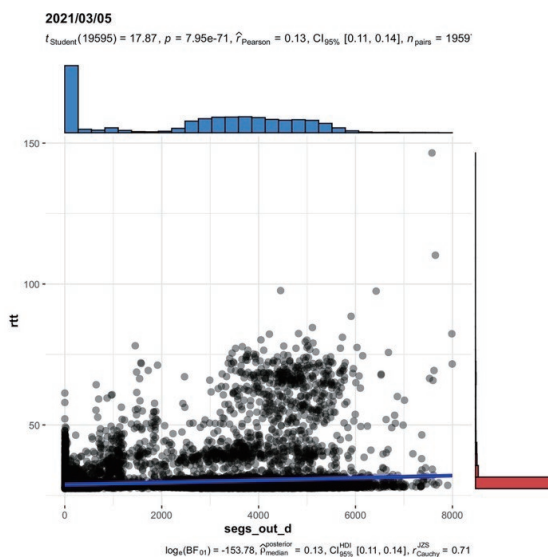


图 8: 20210305 segs_out-rtt

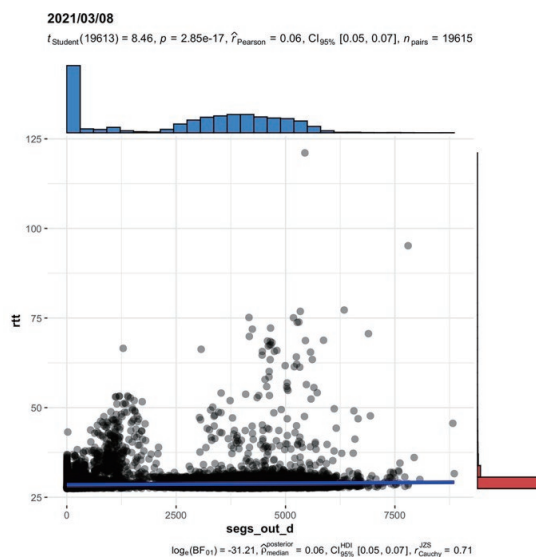


图 9 : 20210308 segs_out-rtt

5. 在 [5720/5750-6240/6250] 区间呈快速衰减，其后区间数据统计极为稀疏，符合模型预测 1；

(四) d_m (segs_out) 采样与 rtt_m 采样间的关系：

过滤后每日监控数据中单位时间传输量 d_m (segs_out) 和对应的 rtt_m 观测值对应关系如散点图 6—图 9 所示。

依据前文约定使用以 d_m (segs_out) 步进 10 作为一个分组，求其中 rtt 的算数平均作为均值，

并剔除掉所有 d_m (segs_out) 分组中采样值少于 10 个的分组。获得 d_m (segs_out) 分组与组内 rtt_m 均值散点图如图 10—图 13 所示。

观察可知经过上述分组、过滤后，所有 d_m (segs_out) 在 600-2000 之间以及大于 5750 的采样值作为稀疏区间值被去掉。可以表明模型描述中 VH 在测试数据链路中反映值约为 5750 左右；在 2000-3000 区间 rtt_m 采样值相对平稳，在 3000-5750 区间 rtt_m 采样值随单位时间传输量 d_m (segs_out) 增长而增长。可以表明模型描述中 VL 在测试数据链路中反映值约为 3000 左右；

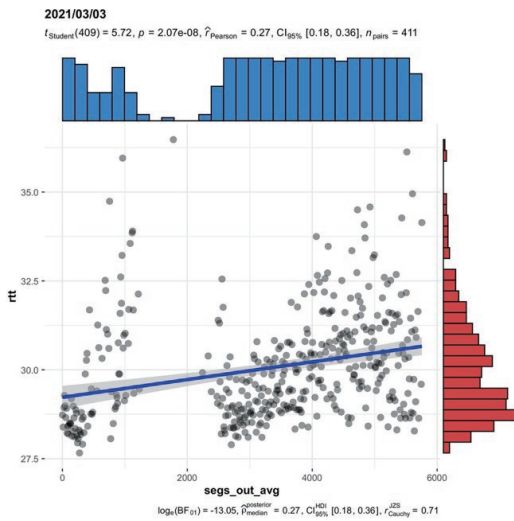


图 10：20210303 - segs_out 分组 /rtt 均值

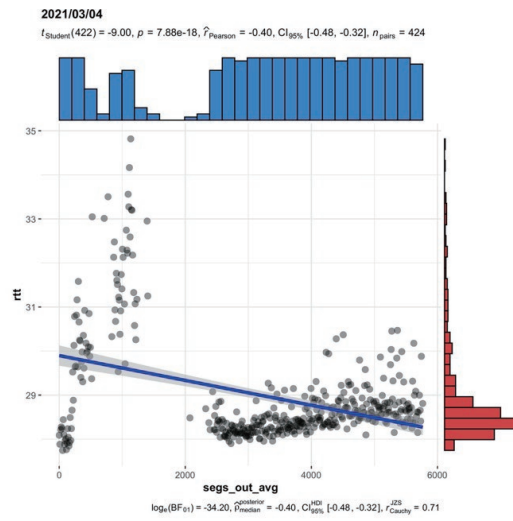


图 11：20210304 - segs_out 分组 /rtt 均值

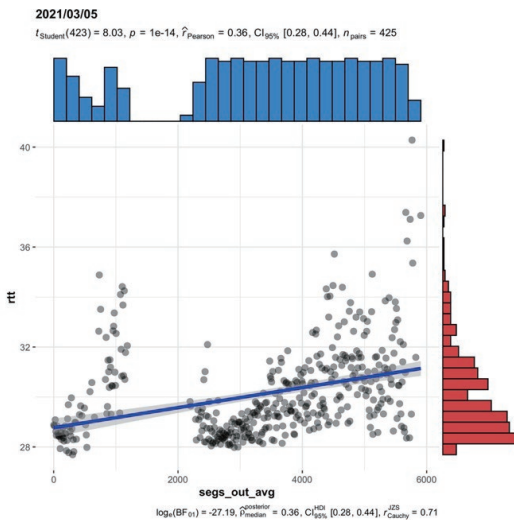


图 12：20210305 - segs_out 分组 /rtt 均值

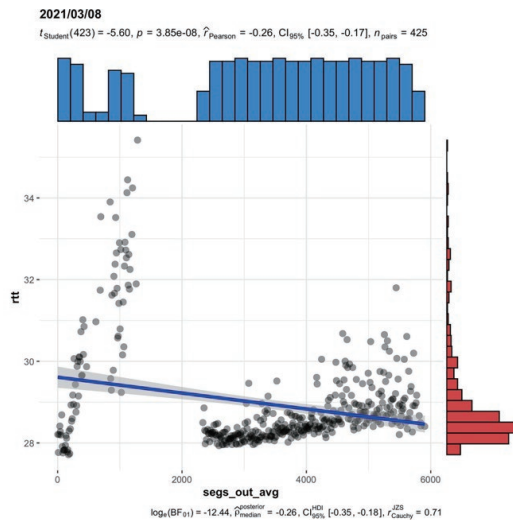


图 13：20210308 - segs_out 分组 /rtt 均值

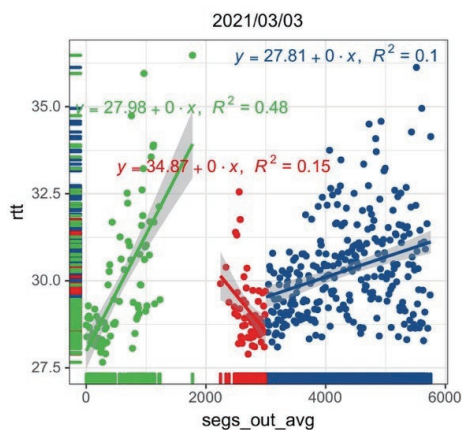


图 14 : 20210303 rtt 分段线性回归

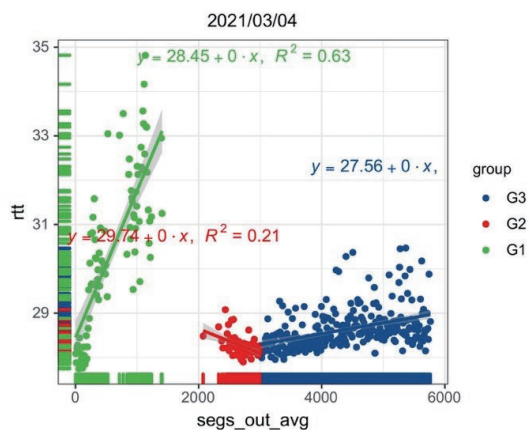


图 15 : 20210304 rtt 分段线性回归

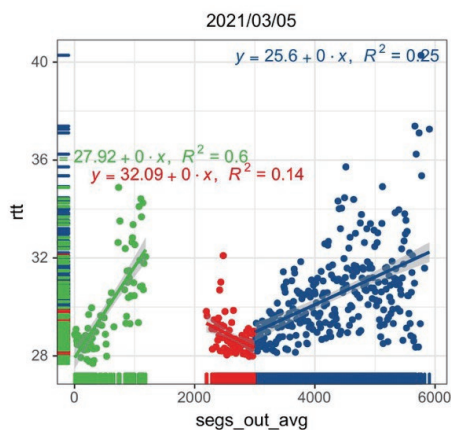


图 16 : 20210305 rtt 分段线性回归

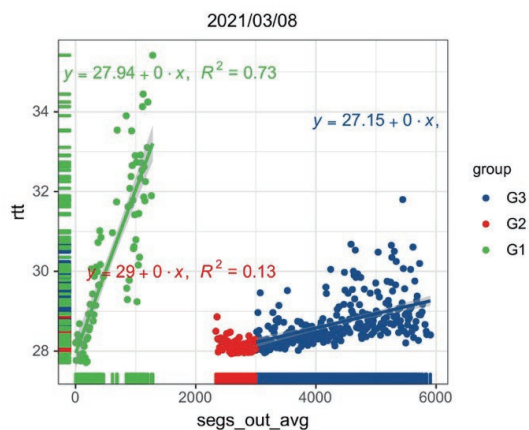


图 17 : 20210308 rtt 分段线性回归

根据上述实际 VL 和 VH 对 d_m (segs_out)、 rtt_m 散点分段, 进行线性回归拟合如图 14—图 17 散点图所示。

据此可得到如下结论:

1. 在数据平均流量较大时 rtt 采样均值与平均流量增加呈正比关系; 符合模型预测 4;
2. 数据平均流量较小阶段 rtt 采样均值小于数据流量较大时段 rtt 采样均值, 符合模型预测 3;
3. 测试链路 VL 值约为 3000, VH 值约为 5750;

(五) d_m (segs_out) 采样与 $rttvar_m$ 采样间的关系:

过滤后每日监控数据中单位时间传输量 d_m (segs_out) 和对应的 $rttvar_m$ 观测值对应关系如图

18—图 21 散点图所示。

依据前文约定使用以 d_m (segs_out) 步进 10 作为一个分组, 求其中 $rttvar$ 的算术平均作为均值, 并剔除掉所有 d_m (segs_out) 分组中采样值少于 10 个的分组, 获得 d_m (segs_out) 分组与组内 $rttvar_m$ 均值散点图如图 22—图 25。

根据上文得到测试链路 VL 值 3000 和 VH 值 5750 可以推算出 $\frac{3 \times VH + VL}{4}$ 约为 5063 左右。对上述散点, 将 d_m (segs_out) 按照 3000、5063、5750 三个关键值进行分段; 可以发现 d_m (segs_out) 在 [3000-5063] 区间时, $rttvar_m$ 采样值随 d_m (segs_out) 增长而增长; d_m (segs_out) 在 [5063-5750] 区间时, $rttvar_m$ 采样值随 d_m (segs_out) 增长而下降; 进而线性回归拟合, 结果如图 26—图 29 所示。

据此可得到如下结论:

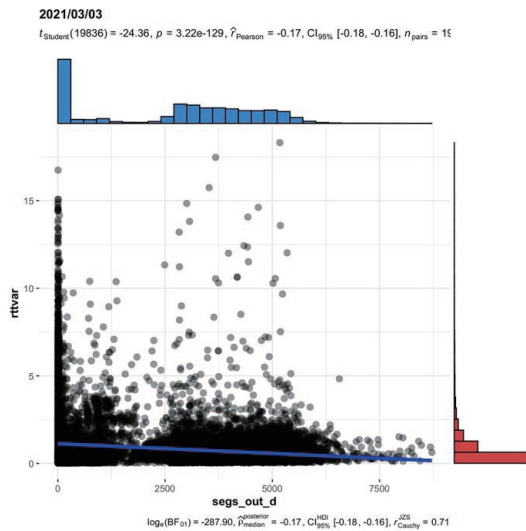


图 18 : 20210303 segs_out-rttvar

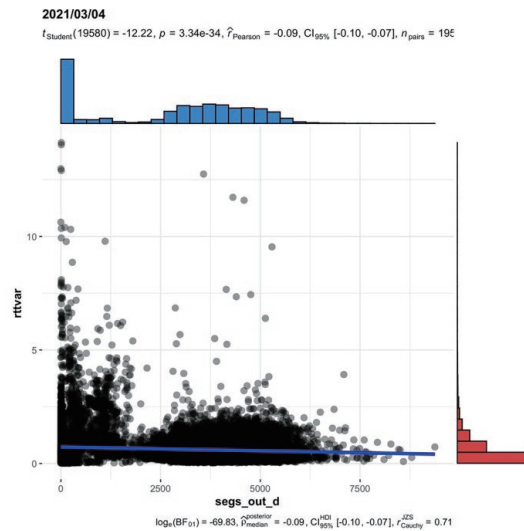


图 19 : 20210304 segs_out-rttvar

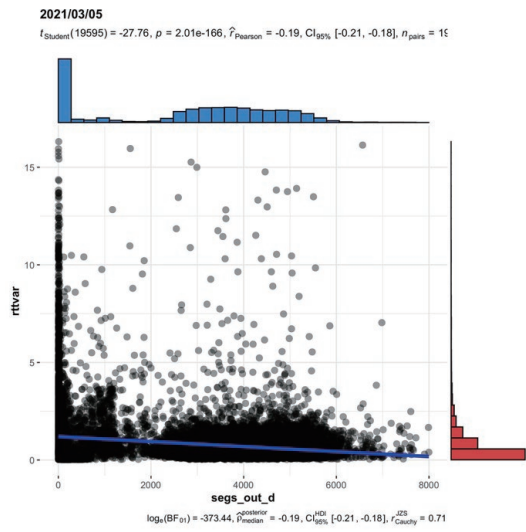


图 20: 20210305 segs_out-rttvar

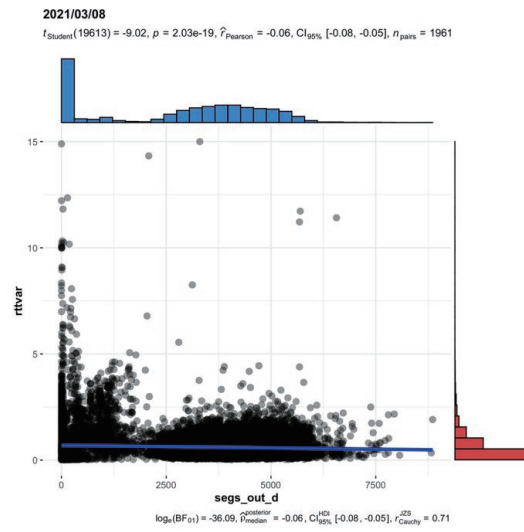


图 21 : 20210308 segs_out-rttvar

1. 在数据平均流量 d_m (segs_out)=5063 左右时 $rttvar_m$ 采样值在存在峰值区域；符合模型预测 5；
2. 根据数据形态可以比较直观获得数据传输状态的优劣：20210304 和 20210308 要优于 20210303 和 20210305；

(六) 检验结果总结

通过上述检验过程可以看到，实际网络传输过程得到的监控数据，基本符合本文对网络数据传输过程建立模型的预测：前文模型预测

部分给出 5 个定性、定比和定量预测均被验证为真。据此得出检验结论：本文建立的网络传输能力和状态模型对真实网络传输过程具备一定的解释能力，模型中引入的网络能力关键指标 VL 和 VH 对评价网络传输能力和状态具备一定指导意义。

同时，本文对网络传输能力和状态建立的模型，以及对模型预测检验所使用的观测和计算方法存在如下不足：

1. 单位时间传输量 d_m (segs_out) 较低区间 (0-500) 内 rtt_m 和 $rttvar_m$ 出现较高离散度采样值，

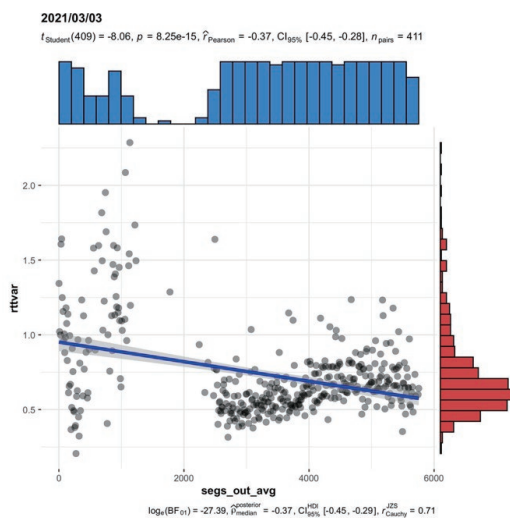


图 22 : 20210303/segs_out 分组 /rttvar 均值

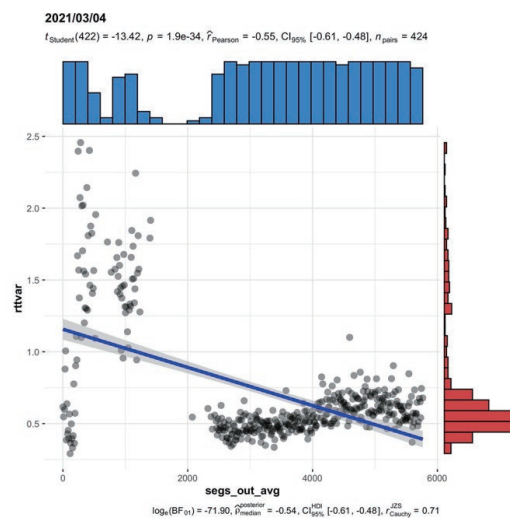


图 23 : 20210304/segs_out 分组 /rttvar 均值

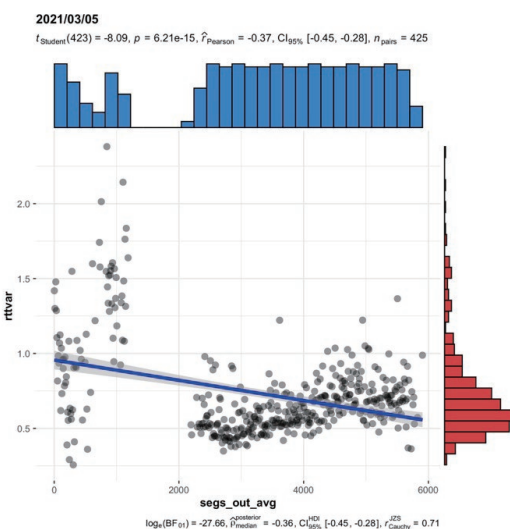


图 24 : 20210305/segs_out 分组 /rttvar 均值

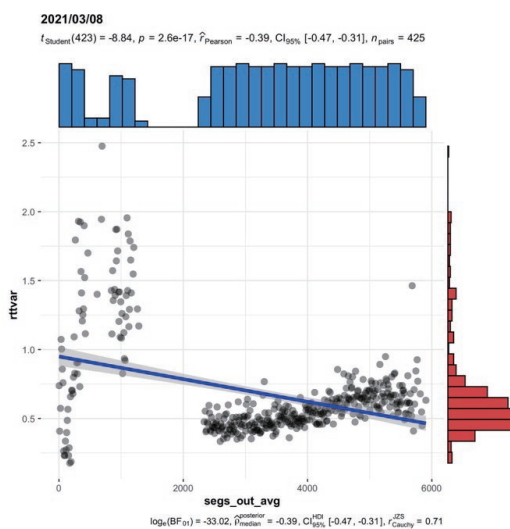


图 25 : 20210308/segs_out 分组 /rttvar 均值

采样区间集中且稳定。本文建立模型对此现象解释能力不足；

2. 在一些测试数据单元（20210303、20210305）中，单位时间传输量 d_m (segs_out) 较高区间（3000–5750）内 rtt_m 采样数据有比较明显的非连续分布的“分层现象”。本文建立模型对此现象解释能力不足；

六、模型的工程价值

本文对网络传输能力和状态建立的模型，以

及对模型预测检验所使用的观测和计算方法，对网络工程具有如下价值：

1. 建立网络传输能力和状态的量化分析能力；
2. 提供识别网络状况的简单直观方法；
3. 解决物理设备细节多样性对能力预测和估计带来的困难；
4. 提供一套方法，针对具体网络连接，利用在端点易得的监控数据，推测网络能力和状态；
5. 为网络开发、调优和问题排查提供数据分析支持；

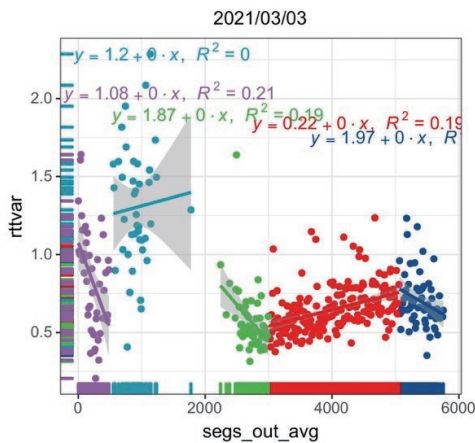


图 26 : 20210303 rttvar 分段线性回归

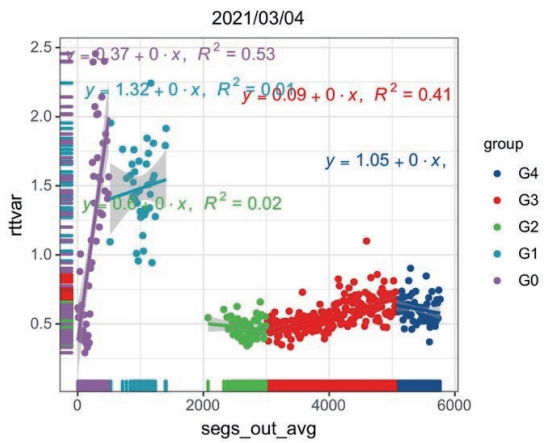


图 27 : 20210304 rttvar 分段线性回归

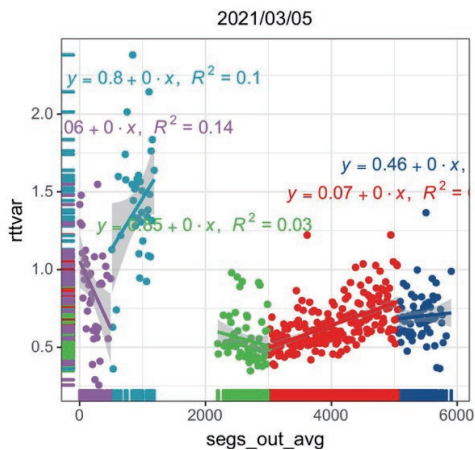


图 28 : 20210305 rttvar 分段线性回归

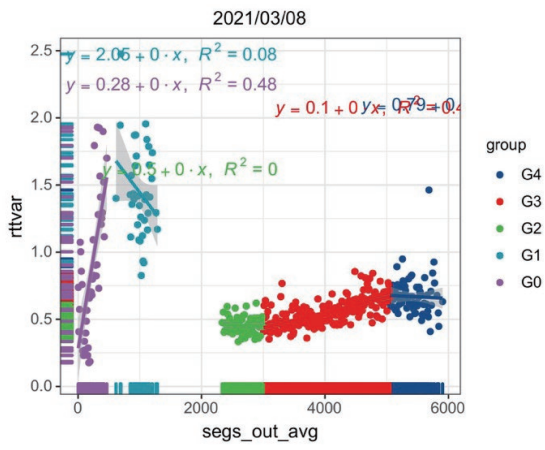


图 29 : 20210308 rttvar 分段线性回归

七、未来的工作方向

针对上文中已发现的、模型存在的不足，可在如下工作方向做进一步探索：

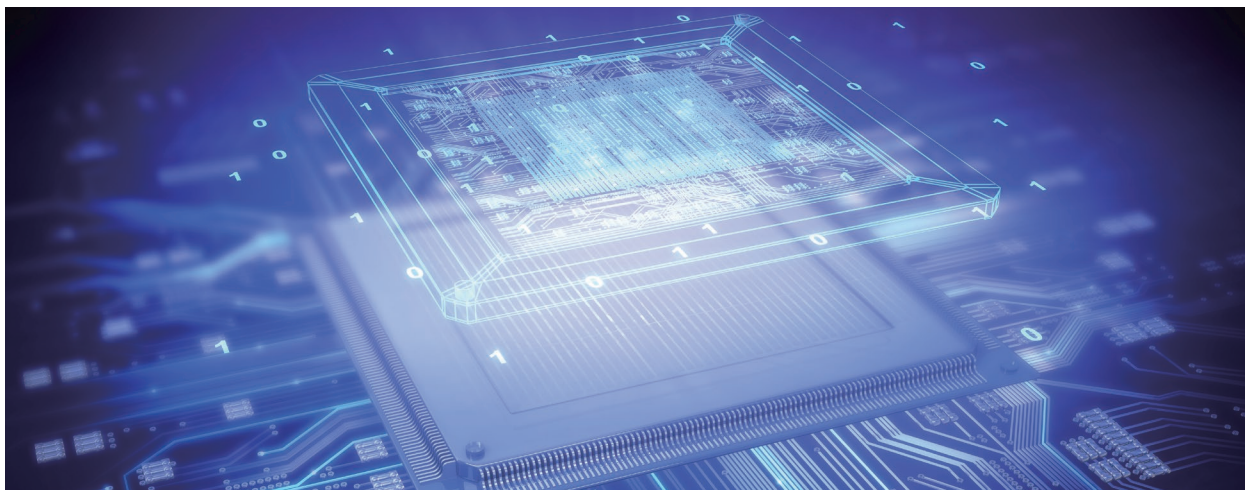
1. 对于“单位时间传输量 d_m (segs_out) 较低区间 (0-500) 内 rtt_m 和 $rttvar_m$ 出现较高离散度采样值”问题：鉴于其采样数据离散程度很高，

同时发生区间稳定且集中的特性，可以考虑是否存在某些模型未涉及因素造成此影响，进一步完善建模过程，细化模型设计；

2. 对于“测试数据单元中采样数据非连续分布”问题，可以考虑从实际存在不同路由、连续高数据负载叠加等方面进行研究；也可以考虑借助频谱分析等工具对数据进行更深入的分析。

浅谈非易失性存储器 在现代交易系统中的应用

储佳佳、刘凯 / 上交所技术有限责任公司 jjchu@sse.com.cn



交易系统作为证券市场的关键核心系统，伴随着新技术的出现不断发展完善，为证券市场注入了活力。交易系统同时关注低时延和可用性，因此既需要对数据进行持久化又需要尽量降低持久化带来的时延损耗。因此新兴存储硬件（例如非易失性存储器）的发展也为交易系统软件的研发带来了新的机遇和挑战，它与传统存储硬件截然不同的新特性促使我们重新思考现代交易系统的存储架构。本文以商用的非易失性存储器产品傲腾 DIMM 为例，分析了将非易失性存储器应用到现代交易系统时面临的关键问题，探讨了面向新硬件特性和交易系统构建的表存储引擎架构。

1 背景

交易系统是资本市场中最核心、最重要的信息系统之一，无论证券交易所的交易系统还是证券公司的交易系统都是市场中最关键的环节。随着计算机技术的进步和证券市场的逐步发展，交易系统也在发生着演变，不断提高处理能力、降低处理延迟、丰富业务模型、提供更高的可用性。交易所交易系统一直以来都基于内存进行交易处理，而近期证券公司部分交易类系统也逐渐从基于数据库的交易处理向分布式基于内存的交

易处理模式改变，这将大大提高证券市场的运行效率。

1.1 传统存储结构的局限性

传统的基于内存的交易系统一般采用基于“内存 + 外部存储（固态硬盘 / 磁盘）”的存储体系结构。这样做能够兼顾性能和成本，充分利用内存访问的高效性和磁盘的低成本、大容量。传统的存储体系结构至少在三类场景下发挥着重要作用，但也存在着问题：

其一，交易日常处理的数据优先放在内存中，

当内存空间不足时，数据置换到外部存储中。数据在内存和外存之间频繁的换入换出操作会影响性能。当数据集大小超过内存容量时，处理性能会下降约 66%。性能下降是因为当数据集大于内存容量时，会触发数据的换入换出，这种交换无论是操作系统控制还是应用控制，都受外部存储性能制约，降低了整体处理能力。

其二，为了实现高可用性，数据在进行实时处理之前需要进行前置持久化，以此保证数据可恢复。前置持久化可以通过记录操作日志、刷新快照等方式来实现，但无论是哪一种方式，其都位于数据处理的关键路径上，做法也均是将内存数据写入外部存储。考虑到内存和外存之间的写性能差异，借助外存进行前置持久化必然会影响数据处理的效率。

其三，在系统的数据恢复阶段，需要从外存中加载数据到内存进行数据结构的重建。数据恢复的总时延取决于外存读操作的效率以及数据文件的大小。从外存中加载数据至内存，从某种意义上也是一种冗余操作，这是内存的易失性导致的。

综上，传统的存储体系架构充分利用了不同存储硬件的优点，但已逐渐无法满足现代交易系统对高吞吐、高可用和低延迟的极致追求。随着硬件的发展，单个内存条的容量可达到 TB 级别，因此可以考虑将交易系统完全运行在内存中。然而，随着数据量的增加，内存容量及性价比的增

速远远不足，并且内存的易失性也使得以上的问题无法解决。

1.2 非易失性存储器的积极意义

存储硬件的发展为解决此类问题带来了新的手段，例如，非易失性存储器（Non-Volatile Memory，简称 NVM）。非易失性存储器是一种持久化存储设备，它保证在断电情况下数据不丢失。NVM 的读写性能远高于外存，在保证数据持久性的同时也能兼顾执行效率。NVM 拥有接近内存的硬件延迟，成本比内存低，可以将其用于扩充 DRAM 的容量。由于 NVM 在性能、容量和成本方面均位于内存与外存之间，NVM 对传统的三层金字塔存储体系结构产生了冲击。传统的三层模型包括高速缓存、内存和磁盘，自上而下容量越来越大、延迟越来越高、成本越来越低。引入 NVM 后，三层金字塔的层级得以扩充，内存和外存的边界变得模糊，如图 1 所示。

不同的 NVM 产品，在硬件工艺、容量、成本和延迟方面也存在差异。得益于存储金字塔层级的丰富和 NVM 产品的发展，我们设计存储系统时的硬件选择变得更多。

NVM 按照寻址方式的不同主要分为两类，一种是按块寻址，它的硬件特性更接近于磁盘和固态硬盘，访问延迟在几十微秒级别。支持 NVMe 协议的固态硬盘属于此类，它的读写延迟

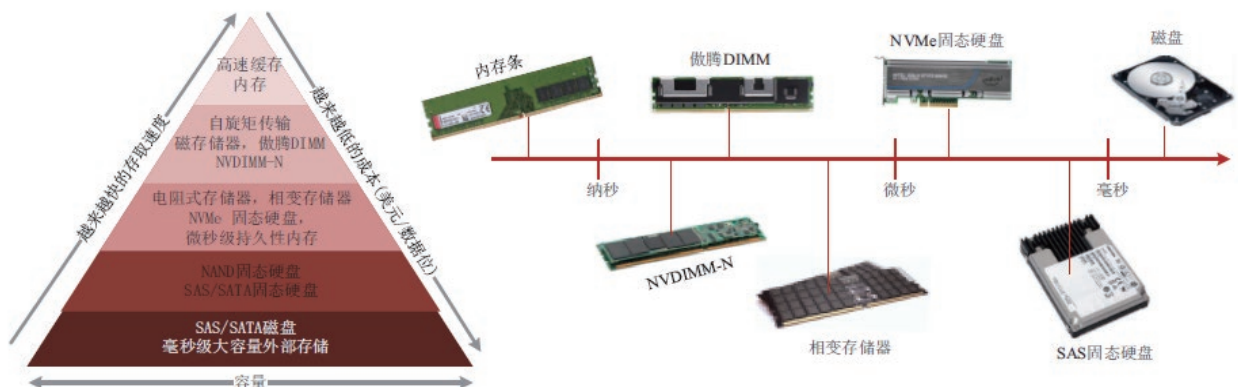


图 1：存储体系结构和设备

为 10-20 微秒。另一种 NVM 按照字节寻址，它的硬件特性更接近于内存，硬件延迟在百纳秒级别。按字节寻址的 NVM 目前已经有商用产品，例如英特尔傲腾（Optane）DIMM。考虑到现代交易系统对低时延的追求，本文重点关注按字节寻址的 NVM，并选用 OptaneDIMM 进行具体分析。如无特殊说明，后文中的 NVM 均特指 OptaneDIMM。

2 关键问题

考虑到 NVM 与 DRAM、闪存、磁盘有着截然不同的硬件特性，将 NVM 应用于现代交易系统时，需要结合新特性进行软件层面的优化，包含以下关键问题：

2.1 单一存储与混合存储

NVM 由于其大容量，高性能，高性价比的特点，为解决 DRAM 的容量不足问题带来了新的思路。不同的 NVM 产品，拥有不同的性质，组合可构建的存储体系结构包括“OptaneD-IMM-only”、“OptaneDIMM+DRAM”和“OptaneD-IMM+DRAM+SSD”三种架构。根据 OptaneDIMM 设备的测试分析，其读写性能和 DRAM 存在了一定差距，尤其是写性能要差 5-10 倍，如图 2 所示。而且，傲腾 DIMM 的耐久性不好，需要尽可能避免写入放大。所以，一般的设计是将 OptaneD-IMM 放在 DRAM 之后，将 SSD 放在 OptaneDIMM 之后，前者通常作为后者的缓存。

2.2 数据迁移策略

考虑到异构存储设备拥有不同的访问延迟和容量，为了最大化吞吐量、最小化时延同时兼顾成本，系统需要根据数据的冷热程度将其放到合适的存储设备中。而且，数据的冷热程度会根据应用负载情况持续变化。因此，设计数据迁移策略时需要解决三个问题，分别是如何选出冷数据、如何将冷数据换出内存、如何将热数据换入内存。可以使用在线维护 tuple tombstone 和布隆过滤器等数据结构来标记和计数，过滤出访问频次最少的为冷数据，或者使用离线的日志进行分析。筛选冷数据的方式包括 LRU、LFU 和 Exponential Smoothing 等。将冷数据换出内存的时机可以设置内存阈值或者使用后台线程来控制。将热数据换入内存，数据的合并方式包括同步或者异步，合并粒度可分为全合并或者增量合并，按照合并时机的不同可分为读入数据页后立即合并、有更新时触发合并和自适应的合并等。

2.3 数据一致性问题

非易失性内存的一致性问题源于缓存到内存的乱序写，以及 NVM 中数据的部分写问题。由于缓存中的数据被刷入内存是乱序执行的，在故障重启后的 NVM 中，可能存在只写了一半的数据或非连续的数据。得益于诸如 CLWB、MFENCE 等指令，我们可以控制将缓存中的数据刷入 NVM，避免数据不一致的问题。还可以借助持久化操作日志的方式来保证数据一致性。

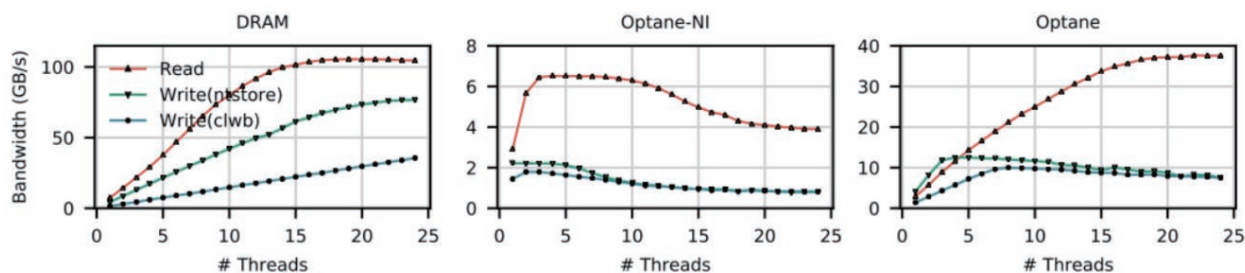


图 2：DRAM 和傲腾 DIMM 的性能对比

3 研究现状

将非易失性内存应用到底层存储系统中，学术界和工业界已经有了一些尝试和实践。SOFORT 是借助 NVM 实现的列存储引擎，它不使用前置的日志持久化同时保证数据的快速恢复。SOFORT 直接在 NVM 中存储数据、更新数据，在系统重启阶段无需将数据从 NVM 中加载到 DRAM 中，因此有着极高的恢复速度。同时，SOFORT 将索引放置在 DRAM 中来加速查找。FOEDUS 是基于混合内存 DRAM 和 NVM 实现的数据库引擎，数据可以驻留在 NVM 或 DRAM 中。混合内存被划分为大小固定的页面，这些页面从 NVM 中加载到 DRAM 中便于后续的读写操作。为了更新 NVM 中的持久化数据，FOEDUS 定期运行一个异步进程，该进程使用日志更新 NVM 中的快照。SAP 公司尝试将 NVM 用于 HANA 数据库，并进行了初步的实践。借助于 NVM，SAP 公司将 HANA 数据库中的增量数据和基线数据进行了存储分离，将不常变化的基线数据存储存在 NVM 中，而将最近修改的增量数据保存在 DRAM 中。由于 NVM 拥有和 DRAM 相近的读性能，混合方式下可以实现对 DRAM 的扩容同时保证查询效率。Peloton 是一个自驱动的存储系统，基于“DRAM+NVM”的存储架构，Peloton

通过一个深度学习组件来实现自动的参数调优。Arulraj 等人面向 NVM、DRAM 和 SSD 实现了近似最优的缓存管理策略，他们考虑了变化的负载，并基于负载构建了模型，寻找最优的存取路径。上述工作均结合 NVM 优化了底层存储的设计，并实现了独特的数据迁移策略，这为我们应用 NVM 优化现代交易系统的底层存储提供了借鉴和参考。然而，这些工作都没有面向现代交易系统的特点、业务场景进行分析和实践，结合 NVM 和交易系统的相关研究目前仍是缺乏的。

4 面向现代交易系统的存储方案探索

NVM 的优良特性和实践可行性促使我们思考如何将 NVM 应用到现代交易系统中，来实现高吞吐、低延迟和快速恢复的性能目标。在存储体系架构的选择方面，可以考虑使用“NVMe SSD+OptaneDIMM+DRAM”替代传统的“SATA HDD/SSD+DRAM”的存储架构，为现代交易系统提供轻量的表存储引擎，如图 3 所示。

表存储引擎使用数据表来组织和管理数据，统一管理不同存储设备。数据表支持非结构化数据。DRAM、OptaneDIMM 和 NVMe SSD 中分别存储着热数据（服务于更新操作）、热数据（服务于查询操作）和冷数据。表存储引擎使用一个

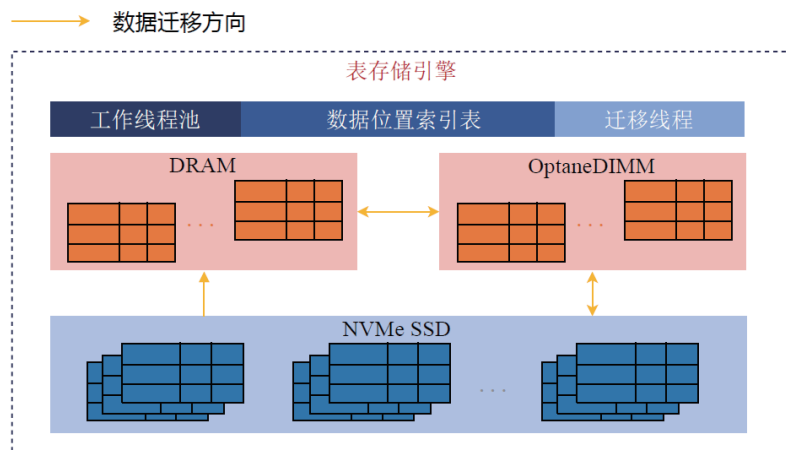


图 3 : 面向现代交易系统的表存储引擎

迁移线程，异步地将 OptaneDIMM 中的冷数据迁移到 NVMe SSD 中。冷热数据的迁移规则基于对应数据的访问频次和访问类型，经常被修改的数据暂存在 DRAM 中，经常被查询的数据存储在 OptaneDIMM 中，一段时间未被更新 / 读取的冷数据存储在 NVMe SSD 中。表存储引擎维护数据的位置索引，来确定待请求数据的设备、偏移量等信息。当表存储引擎接收到数据更新请求时，数据保存在 DRAM 中，并将对应的事务日志持久化写入 OptaneDIMM 中。当接收到数据查询的请求时，表存储引擎通过查询位置索引表，可获得对应数据的位置信息，进而从对应的存储设备上读取相应的数据。因为频繁访问的热数据总是位于 DRAM 或 OptaneDIMM 中，而这两个存储设备的查询效率无明显差别，均可以快速返回待查询的数据结果。使用表存储引擎来维护交易系统的数据，更利于数据共享。全部数据都以数据表的形式保存在存储引擎中，表存储引擎提供一系列的接口供其它组件调用，包括增 (insert)、删 (delete)、改 (update)、查 (select) 和范围查询 (scan) 操作等。借助于这些接口，交易系统的其它组件可以直接和表存储引擎交互，进行底层数据的更新和读取，进而完成订单处理、状态更新以及行情数据揭示等业务层处理逻辑。

对于传统架构之下内存不足引发的数据换入换出问题，我们使用 OptaneDIMM 对 DRAM 进行了容量扩充，在系统启动后全部的交易数据都会

位于 DRAM 或 OptaneDIMM 中，避免了数据在内存和外存之间的迁移。在特殊情况下，当数据集大小超过 DRAM 和 OptaneDIMM 的容量阈值时，表存储引擎也会通过冷热数据的迁移策略保障性能。得益于 OptaneDIMM 自身具有非易失性并且读写性能远远优于外部存储，表存储引擎使用 OptaneDIMM 记录前置持久化的日志可以提高写入日志的效率，同时保证高可用性。表存储引擎还通过异步的快照机制减少数据重启阶段的恢复时间。在系统运行过程中，会有异步线程周期性地 OptaneDIMM 中生成内存数据快照。在系统重启阶段，只需要将最近一次数据快照的标志位设置为可使用状态，再重演少量日志即可实现快速的数据恢复。

5 总结

随着硬件技术和证券市场的不断发展，交易系统也需要与时俱进，结合新技术、新兴存储设备突破性能瓶颈。本文探究了将非易失性存储器应用到现代交易系统的关键问题，以及如何围绕非易失性存储器构建高性能的底层存储。为了实现高性能、高可用的现代交易系统，未来我们将结合本文的思路进行原型验证与系统实现，进一步探索 NVM 的优良特性，以及基于 NVM 构建表存储引擎的实践难点和优化技巧。

雪球期权价格计算的 FPGA 实现

李士昱 / 中信建投证券股份有限公司

孙冬凯 / 中信建投证券股份有限公司

梁程远 / 中信建投证券股份有限公司



雪球期权是一种新兴的结构较为复杂的期权产品，雪球期权的定价的准确性和速度直接影响交易双方的收益和风险水平。目前我司雪球期权定价采用的是基于 C++ 程序实现蒙特卡罗模拟的方法。本文为解决基于 C++ 的传统定价程序带来的处理时间长、延迟高、处理速率低的问题，提出并实现了一种基于 FPGA 的并行流水线计算处理设计，能够完成对雪球期权的定价功能，并使用 HLS 开发模式对设计进行了实现。通过对比测试，相对于通用处理器与 C++ 软件实现的定价方式可获得约 17.83 倍的性能提升。

引言

普通欧式期权通常可以使用 Black-Scholes 模型进行定价，雪球期权在欧式期权的基础上引入的观察日和敲入敲出的概念，使得其定价无法运用类似模型获得一个确定的期望价格，因此需要使用蒙特卡罗方法进行模拟。蒙特卡罗方法是一种对过程随机抽样的程序，我们可以通过随机

采样得到近似的结果，其拥有采样越多，越近似最优解的特点。蒙特卡罗方法广泛应用于科学计算的各种领域。传统思路是使用通用微处理器（CPU）进行蒙特卡罗模拟，在这种情况下通常需要通过提高处理器单核性能或者多线程方式提高模拟速度，也有使用 GPU 等专用计算处理器的尝试。然而近年来，随着摩尔定律渐渐失效，处理器单核性能提升逐渐放缓，而人们对计算量

的需求随着大数据、人工智能等技术的发展却越来越大，因此对传统体系结构的变革需求变得越来越迫切。

随着 FPGA 技术的不断发展，利用 FPGA 进行硬件加速已经逐步成为一个新的趋势。一方面相比通用架构，FPGA 能够根据功能需求进行定制，获得更好的性能和功耗；另一方面相比开发专用硬件，FPGA 的开发周期更短，成本更低，其可重构特性使得能够根据算法变化快速进行更新迭代。此外，Xilinx 公司推出的 HLS 和苹果等公司推出的 OpenCL 等开发标准使得开发者能够使用高级语言如 C 或 C++ 进行硬件开发工作，进一步降低了开发门槛，提升了开发效率。

本文设计并使用 Xilinx 公司提供的 HLS 开发方式，利用 C++ 语言实现了定价实现了基于 FPGA 的雪球期权定价程序，并作为动态链接库，通过 Java 本地接口 JNI 提供调用，作为 dubbo 服务整合到了场外衍生品 OMS 订单管理系统中。

1 雪球期权的定价方法

雪球期权的本质是一种卖出带触发条件的看跌期权。雪球期权属于奇异期权，因此带有奇异

期权的障碍概念，“障碍”是指当标的资产价格在特定时间内穿越某一水平该期权才会生效或者失效。“障碍”一般分为敲出和敲入两类。敲出即当标的资产价格达到一个特定的障碍水平时，该期权了结并兑现赔付，若规定时间内标的资产价格没有触及障碍水平，则为一个普通期权。敲入与敲出期权相反，当标的资产价格达到一个特定障碍水平时，该期权才能触发约定赔付机制，若规定时间内标的资产价格没有触及障碍水平，则不触发约定赔付机制。

一个雪球期权的要素表示如表 1。

雪球期权存续或结束时可能出现三种情形：

(1) 产品发出敲出事件，此时无论是否发生敲入事件，产品将提前结束，客户可获得票息收益，按照产品实际存续期计算。

(2) 产品存续期未发生敲出或敲入事件，客户可在产品到期时获得票息收益，按产品期限计算。

(3) 产品存续期发生了敲入事件但未发生敲出事件，则客户无票息收益，且可能承担亏损， $\text{亏损金额} = \text{名义本金} \times \text{标的证券期末价格相对于期初价格的跌幅}$ ，如无跌幅则无亏损。

由上述条件可得到雪球期权收益率如图 1 所

表 1：雪球期权要素表

| 雪球期权 | |
|--------|--------------------------|
| 标的证券 | 股票 A |
| 期限 | 12 个月 |
| 参考价格 | 标的期初价格 |
| 敲出水平 K | 参考价格×120%，每月观察 |
| 敲入水平 B | 参考价格×80% |
| 票息 | 年化 30% |
| 敲出事件 | 观察日当日标的证券的收盘价大于或等于敲出水平 K |
| 敲入事件 | 任一交易日标的证券的收盘价小于敲入价格 B |

示：

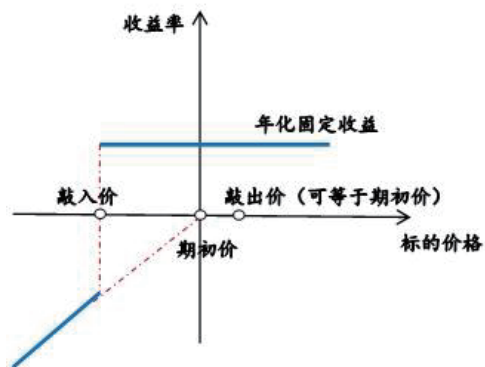


图 1：雪球期权收益

由于雪球期权障碍的存在，我们难以直接给出雪球期权的价格模型，因此考虑采用蒙特卡罗方法对作为标的的股票价格进行抽样，根据股票价格和障碍条件确定雪球期权最终收益，将收益折现后得到雪球期权的当前价格。

2 雪球期权定价程序设计

2.1 雪球期权蒙特卡罗定价程序设计

我们以几何布朗运动模型描述风险中性情况下股票价格随时间变化的离散形式如下：

$$S_{t+\Delta t} = S_t e^{[(r - \frac{\sigma^2}{2})\Delta t + \sigma z \sqrt{\Delta t}]}$$

其中 r 为无风险利率， σ 为股票波动率， z 服从标准正态分布。我们需要对该随机过程进行蒙特卡罗模拟，获得标的股票期权存续期内的价格路径。蒙特卡罗模拟程序主要有两个模块，一是生成服从标准正态分布的随机数，二是完成价格计算，进行股票价格路径模拟，并根据模拟结果计算雪球期权价格，由此可得到定价程序流程图如图 2 所示。

在通用处理器上程序都是串行顺序执行，执行效率不高。本文提出了基于 FPGA 的解决方案。在处理方式上采用流水线执行，并在路径模拟部分根据硬件资源使用进行一定程度的并发，进一步提高了路径模拟的吞吐量。

2.2 定价程序开发模式

传统 FPGA 开发方式通常采用 RTL 级描述，如使用 Verilog 或 VHDL 语言。RTL 级开发虽然直接描述硬件的行为，能够获得更加准确、性能更好的电路设计，但是对于软件工程师来说门槛较高，调试优化时间也更长，就好像在用汇编语言进行软件开发一样。为了解决这些问题，Xilinx 公司推出了支持高层次综合（High-Level Synthesis）的编译工具，使开发人员能够使用 C/C++ 语言进行 FPGA 开发，把精力更多集中在算

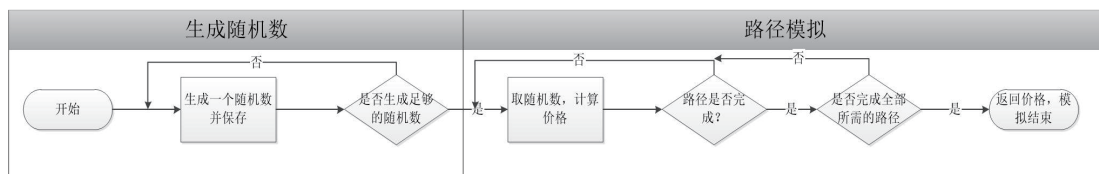


图 2：雪球期权定价程序流程图



图 3：雪球期权定价在 FPGA 多并发流水线的执行流程示例

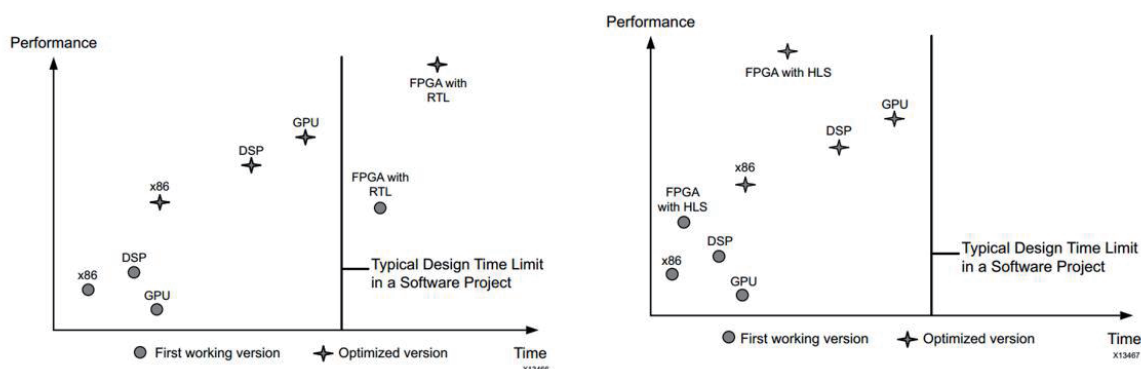


图 4：不同平台开发方式时间和理想性能对比

法实现上面，不需要亲自动手进行底层细节的实现，极大提升了开发效率。图 4 展示了使用 RTL 设计方法的传统 FPGA 开发方式、HLS 开发方式与其他平台主流设计开发方式的开发时间和理想性能对比。

总体来说，HLS 可以自动完成以下曾经需要手动完成的工作，包括：

- (1) HLS 自动分析并利用一个算法中潜在的并行性；
- (2) HLS 自动在需要的路径上插入寄存器，并自动选择最理想的时钟；
- (3) HLS 自动产生控制数据在一个路径上出入方向的逻辑；
- (4) HLS 自动完成设计的部分与系统中其他部分的接口；
- (5) HLS 自动映射数据到储存单位以平衡资源使用与带宽；
- (6) HLS 自动将程序中计算的部分对应到逻辑单位，在实现等效计算的前提下自动选取最有效的实施方式。

本文采用 HLS 方式进行雪球期权定价程序的开发，一方面可以降低开发难度，将幂计算、开方计算、浮点数处理等硬件开发难点交由编译器进行处理，提升开发速度，另一方面能够专注于算法实现，从更高层级优化程序的执行效率，提升程序优化效率，在更短时间内获得更高性能提升。

3 定价程序实现与优化

3.1 总体设计与实现

雪球定价程序参照 Xilinx 推荐的层次结构，总体自下至上可以分为三个层次：最下层 L1 层提供了用于构建内核的低级原语，包括随机数生成器、路径模拟、累加器等模块；中间层由 L1 层各模块构建，并辅以更高层次层次逻辑控制和计算模块，并以 FPGA 内核形式提供定价引擎的接口；L3 层则将数据传输、与内核相关的资源配置和任务调度的低层细节抽象化，提供了供高级语言或软件可直接调用的接口，通过调用即可完成定价引擎内核的部署和运行。

此外，我们将定价用 JNI 包装后编译成动态链接库，使用 Java 微服务进行调用，从而将定价服务云化，进一步为交易人员提供使用便利。图 5 展示了定价功能的整体架构。

3.2 核心模块设计实现与优化

蒙特卡罗模拟主要功能在 L1 层实现，L1 层共分为随机数生成、路径模拟和累加器三个模块。

3.2.1 随机数生成

随机数生成模块要求能够生成符合标准正态分布的随机数，这里我们复用了 Xilinx 提供的随机数发生器。Xilinx 共提供了三种正态分布随机数发生器，分别为 MT19937IcnRng、MT2003IcnRng 和 MT19937BoxMullerNomralRng。

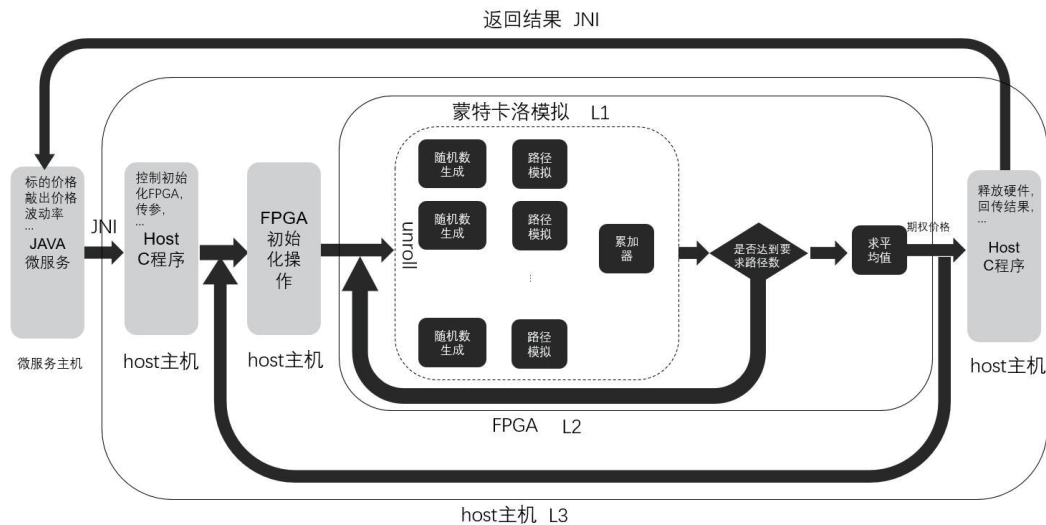


图 5 : FPGA 雪球定价程序总体架构设计

第一种使用 MT19937 随机数，通过逆累计函数变换获得最终结果；第二种使用 MT2203 随机数，同样为逆累计函数变换获得结果，与第一种不同的是 MT2203 的周期略短，同时理论上多实例的相关性更弱，随机性也更好一些；第三种是使用 MT19937 随机数，通过 Box-Muller 变换得到正态随机数，其优点是性能和资源消耗略小。本文采用 MT19937BoxMullerNomralRng 随机数发生器来生成随机数。生成的随机数会按序放入一个队列结构中，供路径模拟模块取用，并通过这一队列将两个模块衔接起来组成一个长流水线。在 HLS 中这种队列使用 hls 命名空间的 stream 类型来表示。

3.2.2 路径模拟

路径模拟是定价程序最重要的模块，也是逻辑最复杂、对总体性能影响最大的模块。在该模块中需要对一条完整的随机过程路径进行模拟，也即模拟标的股票在期权存续期内的价格，并根据价格判断是否触发雪球期权的敲入敲出。

路径中每节点开始首先判断期权是否已经敲出，如敲出则该路径直接结束，如未敲出，则将上一节点价格和随机数计算带入定价公式，算出当前节点股票价格，随后判断是否是观察日和敲入敲出等障碍条件是否满足，并根据结果调整敲

入敲出标志，最后判断该路径模拟是否已经结束，如未结束则开始模拟路径中下一个节点，处理流程如图 6 所示。

FPGA 的性能优势主要体现在流水线处理和并行两个方面，在 HLS 开发过程中主要是通过循环进行流水线处理和展开实现，可以通过加入 Pragma 提示编译器进行优化工作，如：

```
for(int i = 0; i < maxObdays; i++){
    #pragma HLS pipeline II = 1
    ...
}
```

其中 #pragma HLS pipeline 即告诉编译器将该循环按照流水线展开，II 为循环起始间隔，即本次循环到下一次循环开始的时钟周期数，本例 II=1 即告诉编译器我们期望每周期能够启动新的循环迭代，相应的即流水线填满后每周期能够产生一个结果，编译器会根据期望结果进行尝试，实际优化效果未必能够完全实现。

编译器对于能够流水线化循环的结构也有两点要求：一是循环尽量为单层循环且有固定边界，二是如果无法满足单层循环，则尽量在两层循环之间没有逻辑操作且让内层循环拥有固定边界。

回顾路径模拟的处理流程图我们可以发现，循环初始对敲入敲出的判断在 CPU 顺序执行的

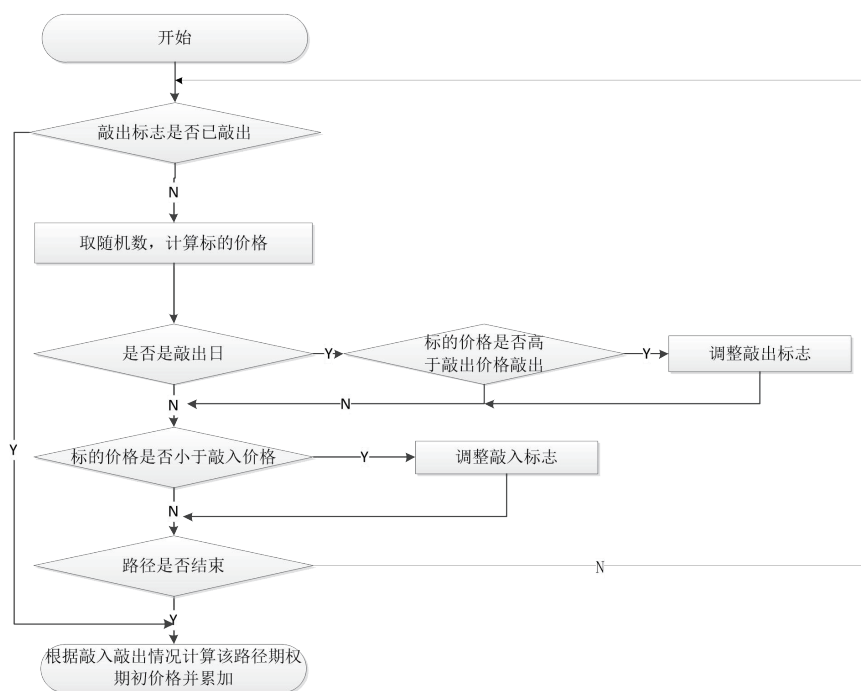


图 6 : 路径模拟模块初始处理流程

情况下能够避免后续无用的循环，但是在 FPGA 平台则会导致循环无法顺利流水化，因此需要对处理流程进行调整，将敲出结果保存下来留到循环结束再进行处理，从而固定循环边界，使循环能够顺利展开并减少分支损耗。该优化使得循环

迭代周期从 $II=17$ 降至 $II=5$ ，优化后的处理流程如图 7 所示。

3.2.3 累加器

累加器模块一方面将各个路径的结果累加并求平均得到最终结果，另一方面也负责随机数生

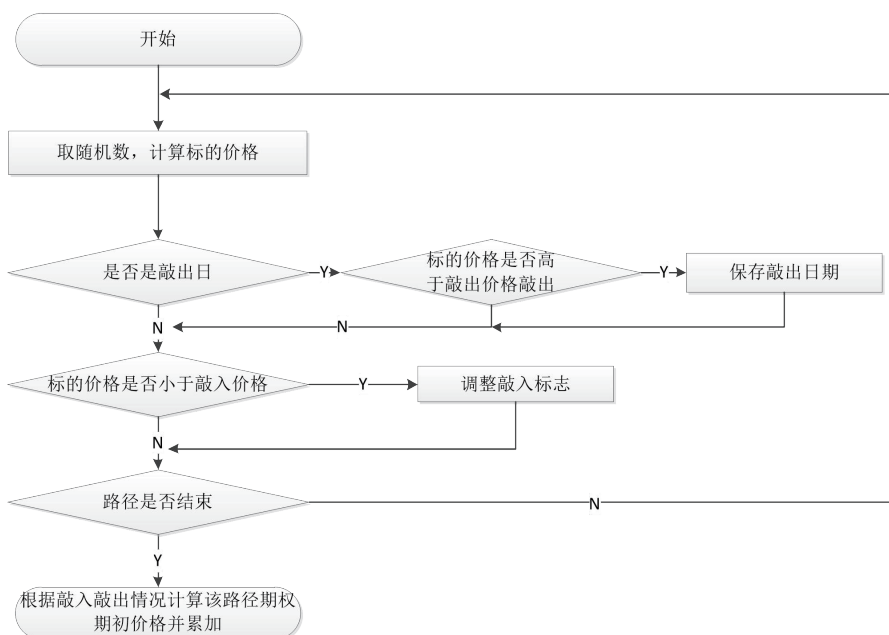


图 7 : 优化后的路径模拟处理流程

成和路径模拟两个模块的衔接，以及路径模拟的并行展开控制。

由于蒙特卡罗模拟路径各路径间的结果相互独立，因此非常适合并行执行，在 HLS 中的实现方式也非常简单，如下所示：

```
for(int i = 0; i < unrollNum; i++){
    #pragma HLS unroll factor = unrollNum
    mcSimulation();// 蒙特卡罗模拟函数
}
```

通过 #pragma HLS unroll 告知编译器对该循环进行展开并行执行，factor = unrollNum 告知编译器并行度，该部分可以省略，如省略则编译器会将循环完全展开。

各路径模拟结果将会暂存至与随机数类似的队列流中，该队列在实例化为硬件后将会是一个 RAM 结构，其读写口数量和大小编译器会根据代码自动确定，通常为 1 读口 1 写口。使用同样 RAM 结构的 C++ 数据结构还有数组等。当循环充分流水线化且并行展开后，对 RAM 的读写将会非常频繁，自动生成的 RAM 读写口数量可能会无法满足我们的需求，造成流水线阻塞。这时候就需要告知编译器所需要的 RAM 大小和读写口数量，一种方式是直接告知采用的 RAM 规格，另一种方式是通过将数组或队列拆分。以一个完全拆分即每一个数均采用寄存器存储，深度为 64

的队列为例，使用 HLS 的实现写法为：

```
#pragma HLS stream variable = sumStrm depth
= 64
#pragma HLS array_partition variable =
sunStrm dim = 0
```

4 测试与分析

4.1 测试环境

FPGA 测试平台：Xilinx 公司的 Alveo U200 FPGA 加速卡，该 FPGA 加速卡被划分为 3 个内核，本次测试仅测试单核下的资源利用达到极限的性能，XRT 运行环境版本为 u200_201830_2，在 Vitis 2019.2 开发环境下完成的开发、综合、验证工作。

对比程序为使用 QuantLib 库开发的基于蒙特卡罗模拟的雪球期权定价程序，开发语言为 C++，运行环境处理器为 Intel i7-8700，主频 3.2GHz，内存 16GB，操作系统为 64 位 Windows 10。

测试方法：在两个平台使用相同期权参数分别运行雪球期权的定价程序，蒙特卡罗路径数为 20 万条，路径节点数为 246 个，对应 246 个交易日。程序循环运行 100 次，每次对股票波动率进行调整，对比平均运行时间和定价结果误差。期权主要参数如表 2 所示。

表 2：雪球期权测试参数

| 雪球期权测试参数 | |
|----------|-----------------------|
| 期限 | 12 个月（246 个交易日） |
| 观察日数量 | 12 个 |
| 期初价格 | 100 |
| 参考价格 | 同期初价格 |
| 敲出水平 K | 参考价格×120%，每月观察 |
| 敲入水平 B | 参考价格×80% |
| 票息 | 年化 30% |
| 波动率 | 初始值为 0.25，每次循环增加 0.01 |

4.2 结果分析

FPGA 与 C++ 对比程序定价结果如图 8 所示，FPGA 定价结果相对 C++ 定价结果误差平均值为 2.6%，方差为 0.0245。与 FPGA 定价结果相比，C++ 程序的定价曲线更不平滑，考虑到两边随机数的种子与随机数生成方式均不相同，且蒙特卡罗为对随机过程的抽样近似，可以认为 FPGA 的定价结果准确，能够满足定价使用要求。

两者执行时间对比，C++ 程序平均每次定价时间为 1153.48ms，FPGA 程序平均定价时间为 62.75ms，性能提升约为 17.38 倍。资源利用率如表 3 所示，其中最高的 LUT 查找表单核使用率

为 98%，基本达到单核性能极限。

5 结束语

本文采用流水线并行的设计结构实现了对雪球期权的定价，相比于传统软件的处理方式，本设计在处理时间上显著提高，其性能提升约为 17.38 倍左右。基于本设计在现有的 Alveo U200 FPGA 加速卡上使用 HLS 开发模式构建了雪球期权定价功能，具有稳定性好、延迟低、效率高、处理速率快等特点，取得了良好的测试效果，具有较高的工程实用价值。

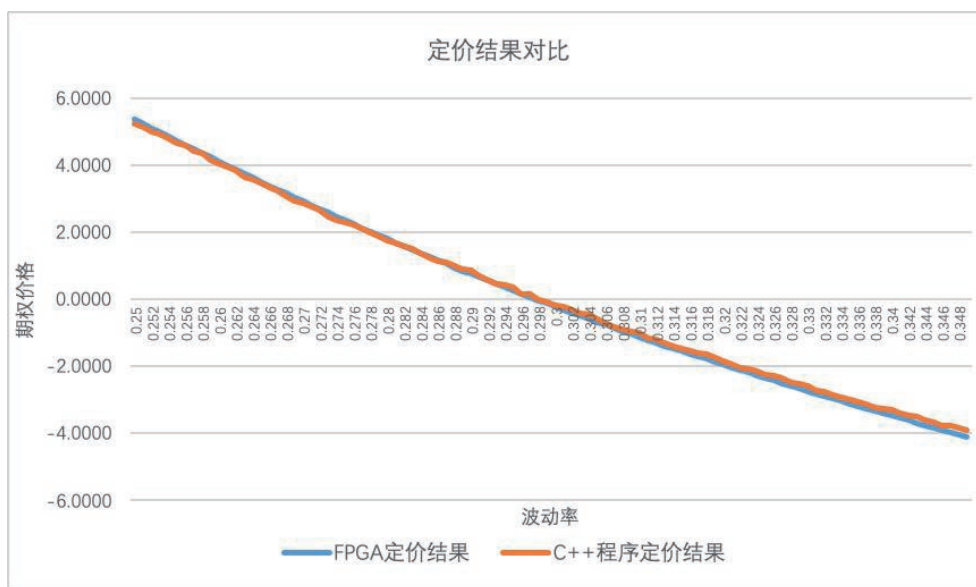


图 8：雪球期权 FPGA 定价结果与 C++ 程序定价结果对比

表 3：定价程序 FPGA 资源使用情况

| 资源 | 单核已使用数量 | 单核总数量 | 单核利用率(%) | 可用总数量 | 总量利用率(%) |
|----------|---------|--------|----------|---------|----------|
| BRAM_18K | 1037 | 1440 | 72 | 4320 | 24 |
| DSP48E | 1933 | 2280 | 84 | 6840 | 28 |
| FF | 438167 | 788160 | 55 | 2364480 | 18 |
| URAM | 0 | 320 | 0 | 960 | 0 |
| LUT | 386844 | 394080 | 98 | 1182240 | 32 |

I 信息资讯采撷 nformation

监管科技全球追踪



监管科技全球追踪

国际动态

国际清算银行协同英国与北欧四国央行推出创新中心

6月11日，国际清算银行（BIS）宣布与英格兰银行在伦敦合作设立BIS创新中心（BIS Innovation Hub）。BIS创新中心自2019年推出，目前已先后落户中国香港、新加坡和瑞士。据悉，BIS创新中心目前的工作主要涵盖六个方面：监管科技、下一代金融市场基础设施、央行数字货币、开放金融、网络安全和绿色金融。6月16日，国际清算银行和北欧地区四个中央银行（丹麦国家银行，冰岛中央银行，挪威中央银行和瑞典中央银行）于斯德哥尔摩成立BIS创新中心北欧中心，这是BIS在过去两年中开设的第五个创新中心。该创新中心的成立是扩大BIS创新中心全球影响力计划的一部分，该计划还包括与加拿大银行（多伦多）、英格兰银行（伦敦）和欧洲中央银行/欧元系统（法兰克福和巴黎）开设中心，BIS创新中心已经成为央行金融科技实验的关键驱动力。

国际清算行（BIS）发布调研报告：《关于数据美国消费者更加信任谁？》

5月27日，国际清算行（BIS）发布调研报告《关于数据美国消费者更加信任谁？》（Whom do consumers trust with their data? US survey evidence）。报告称在最近一项调查中，受访的美国家庭表示，在隐私数据的保护上，相比政府机构或金融科技公司，他们更信任传统的金融机构。这一研究结果在不同的群体中稍有差异：其中少数族裔对金融机构的信任度较低，

而年轻受访者对金融科技公司的信任度则相对较高；从调研对象整体看，女性、少数族裔和年轻受访者更关心数据共享对个人安全的影响。此外，四分之一的受访者还表示，新冠疫情降低了其分享数据的积极性，而其中近一半受访者因担心身份盗窃与数据滥用不愿与大型科技公司共享数据。随着数字化进程不断推进，金融服务领域会有更多的参与者，而强有力的数据保护政策也变得愈发重要。

谷歌推出 Datashare 提升金融服务数据管理效能

5月27日，Google在Google Cloud金融服务峰会上，推出了用于金融服务的Datashare，Datashare建立在BigQuery等谷歌云分析服务之上，并利用最近发布的Analytics Hub，旨在让数据发布者和使用者更轻松、更安全地共享市场数据。数据共享包括批处理数据传递，该功能可以帮助发布者通过BigQuery传递参考数据、历史标记数据、可选市场数据源等。基于事件的数据传递是Datashare的另一个特性，它实时处理工具价格、订单、新闻等，可以倒带到某个时间点，让团队重现之前的市场场景。此外，Datashare还提供数据货币化服务，让市场数据发布者将授权数据集带到Google Cloud，并通过Google Cloud Marketplace提供这些数据，从而提供新的销售渠道。

金融稳定协会（FSI）发布《针对反洗钱的加密资产监管》研究报告

4月9日，金融稳定协会（FSI）发布《针对

反洗钱的加密资产监管》(Supervising cryptoassets for anti-money laundering) 研究报告。报告指出, 尽管部分加密资产可以大大提升交易效率, 但其超高的交易速度、广泛的全球辐射能力、交易匿名性以及缺少金融中介等特性使得加密资产容易被滥用, 增加了洗钱与恐怖主义融资 (ML/TF) 的风险。实际上, 当前加密资产的非法使用已具有相当规模, 亟需相关反洗钱 / 反恐融资监管

及执法的跟进, 为此金融行动特别工作组 (The Financial Action Task Force) 已迅速采取行动。然而, 国际标准的有效性取决于各国当局的有效落地, 目前, 对加密资产服务供应商的监管在全球范围内仍处于起步阶段。报告旨在通过对新出台的监管方法与实践进行评估, 帮助各金融监管机构解决其面临的共同挑战并确定应对的政策重点, 为这一领域的国际讨论做出贡献。

欧美动态

美国证券交易委员会 (SEC) 宣布今年将不会处理加密货币的监管问题

6月14日, 美国证券交易委员会 (SEC) 公布了2021年的监管议程, 列出了行政机构计划采取的短期与长期监管行动。该委员会今年的监管重点清单中没有提及加密货币、区块链或比特币或以太坊等加密货币。这对于许多市场观察人士来说较为出乎意料, 因为SEC主席曾多次发表公开评论, 表示加密货币领域需要更多监管以保护投资者。

德国联邦金融监管局发布大数据与人工智能监管原则

6月15日, 德国联邦金融监管局 (BaFin) 发表了金融机构在决策过程中使用算法、大数据和人工智能的监管原则。这些原则构成了与人工智能使用相关的最低监管要求的初步构想。因此, 这些原则构成了与各种利益相关者进行讨论的基础, 这些利益相关者不仅包括市场参与者和学术界成员, 还包括其他国家和国际监管机构以及标准制定者。欧盟委员会在其数字金融战略中宣布, 打算最迟在2024年前与欧洲监管机构 (ESAs) 一起阐明现有金融市场监管是否以及如何适用于

大数据和人工智能 (BDAI) 的应用。

欧洲央行 (ECB) 在欧元国际角色年度报告中强调数字欧元的重要性

6月2日, 欧洲央行 (European Central Bank, ECB) 发布了欧元国际角色 (The International Role of The Euro) 年度报告。报告指出, 欧洲需要部署数字欧元来增强当前的跨境支付基础设施。数字欧元可以消除在国际交易中使用外币的必要性, 并降低与此相关的成本, 从而促进全球电子商务的扩张, 发行数字货币 (CBDC) 将有助于保持国内支付系统的自主权以及数字世界中货币的国际使用。该报告还指出, 央行数字货币的特定设计特征对其全球推广很重要, 并强调需要通过互操作性、用户匿名性和能够进行离线支付来激励数字欧元的使用。然而, 经济学家强调, 匿名性还必须与获得央行数字货币用户的足够信息需要相结合, 以“建立保障措施”并识别滥用资金用于资助恐怖主义、跨境犯罪活动和洗钱的情况。

欧盟委员会主席发表“引领数字十年”讲话

6月1日, 欧盟委员会主席 Ursula von der

Leyen 在 2021 年数字大会开幕式上发表主题为“引领数字十年”的讲话，提出数字时代下欧洲战略的四个基点——数字技能、基础设施、业务和电子政府。她强调，为了应对数字时代的挑战，欧盟必须在上述四个方面取得进展。其内容具体包括：技能方面，到 2030 年欧盟希望雇佣 2000

万 IT 专家；基础设施方面，到 2030 年欧盟希望所有家庭都能实现千兆网络连接，所有城市和村庄都能实现 5G 网络；企业数字化转型方面，到 2030 年四分之三的企业将使用云计算、大数据和人工智能；数字公共服务方面，到 2030 年所有欧盟公民都应该可以使用电子健康记录。

亚太动态

新加坡启动第六届“绿色金融主题”金融科技黑客加速器项目

5 月 6 日，新加坡金管局（MAS）宣布启动第六届“全球金融科技黑客加速器项目”（Global FinTech Hackcelerator）。本届活动由 Oliver Wyman 提供支持，以“强化科技力量，助力绿色金融”（Harnessing Technology to Power Green Finance）为主题，旨在推动新加坡和整个东南亚地区的绿色金融发展。据悉，本届活动通过向金融机构和绿色金融从业者的意见征询收集了 50 多个问题，并希望参与活动的全球各家机构就资本动员、承诺监测和影响力评估三个方面提交绿色金融发展创新解决方案。

亚马逊在迪拜建立金融科技实验室

亚马逊支付服务（Amazon Payment Services）是亚马逊公司在中东和北非（MENA）地区推出的一项支付处理服务。2021 年 6 月 15 日，该公司宣布在迪拜国际金融中心下属的创新中心（DIFC Innovation Hub）设立亚马逊金融科技实验室。作为落户 DIFC 创新中心的首家此类机构，亚马逊金融科技实验室将与 DIFC 创新中心的各类中小企业、金融机构、创新企业开展合作，共同推动该地区支付乃至整个金融科技产业的发展。

韩国金融科技周落幕，29 家企业累计融资 1337 亿韩元

5 月 26 日，2021 韩国金融科技周如期举行。本次科技周采取“线上+线下”的模式，面向金融科技企业、金融公司、相关机构和个人推出了各类展览、教育、投资和就业机会。据统计，共有 29 家企业在本次活动周期期间收获融资，总额 1337 亿韩元。共有 54 家企业和机构参与在线招聘会活动，175 人成功收获职位邀请。此外，共有来自 13 个国家和地区的 39 家国际金融科技企业、金融公司和投资者参与了在线展览，为该行业的全球合作与海外扩张提供了可能。

香港金融管理局（HKMA）将举办“释放监管科技之力”活动

6 月 1 日，香港金融管理局（HKMA）宣布将于 6 月 30 日举办其在监管科技领域的重要活动“释放监管科技之力”（Unlocking the Power of Regtech）。该活动是香港金融管理局于 2020 年 11 月公布的两年期监管科技推广路线图中的一个重要里程碑，活动汇集了全球监管科技生态系统的高层人士，包括金融机构、监管方、监管科技供应商及其他行业专家来分享他们关于监管科技的经验与见解，活动的主要发言人包括：中国人民银行副行长范一飞、国际清算行创新中心负责

人 Benoît Cœuré、英格兰银行国际银行监管部执行董事 David Bailey、花旗银行 CEO Jane Fraser 等等。本次线上活动对所有对于监管科技感兴趣的相关方开放，名额有限，香港金融管理局鼓励尚未登记的有关各方尽早注册登记。

印度国家银行卡片部门基于云生态拓展业务

4月14日，印度国家银行（State Bank of India, SBI）卡片部门将人力资源和营销功能迁移到云上，以此获得更高的灵活性、可扩展性及合作效率。与通用电气合作关系的结束是 SBI Card 决心开发更简单高效的新系统、流程和框架的契机，卡片部门想要将整个人力资源生态系统迁移至云上，将员工的整体体验提升到一个新的水平。

我国将加快建设“5G+工业互联网”融合应用先导区

6月15日，2021世界工业互联网大会召开。工业和信息化部、国务院国资委相关负责人在会上集中发声，将全面推进工业互联网网络、平台、安全三大体系建设，支持工业互联网产业示范基地建设，建设“5G+工业互联网”融合应用先导区，加快培育工业互联网骨干龙头企业。当前工业互联网创新发展取得积极成效。工业和信息化部信息通信管理局副局长鲁春丛在会上表示，新型基础设施建设加速推进。高质量外网覆盖全国300多个地市，企业内网改造也在加快推进。“5G+工业互联网”全国在建项目超过1500个。多层次工业互联网平台体系初步形成。

人民银行在14省启动金融数据综合应用试点

5月26日，人民银行在14个省（区、市）组织商业银行、清算机构、非银行支付等开展金融数据综合应用试点，旨在探索运用人工智能、大数据、物联网、隐私计算等新一代信息技术，在安全合规的前提下推进金融数据高效治理、安全共享，实现跨层级、跨机构、跨行业数据融合应用，提升金融核心竞争力和惠民利企能力。据悉，试点将着力加强数据能力建设，建立数据全生命周期管理体系，建设支持信息整合、建模处理、算法分析的综合型数据中台，增强数据质量、易用性和安全性。同时，将促进数据规范共享，充分发挥全国一体化政务服务平台的数据共享枢纽作用，运用多方安全计算、联邦学习、联盟链等技术实现多主体间数据规范共享。此外，将深化数据融合应用，加强金融与公共服务领域数据综合应用；强化监管科技应用，提升金融风险防控数字化、智能化水平。

人民银行等7部门启动金融科技赋能乡村振兴示范工程

4月16日，人民银行会同农业农村部、工信部、人力资源社会保障部、交通运输部、商务部、国家卫生健康委，在江苏省、安徽省、福建省、江西省、山东省、河南省、重庆市、四川省、陕西省启动金融科技赋能乡村振兴示范工程。示范工程旨在探索运用新一代信息技术因地制宜打造惠农利民金融产品与服务，全面提升农业产业现代化水平、农村金融承载能力和农民金融服务可得性，为乡村振兴战略实施提供坚实金融保障。

2021年三季度《交易技术前沿》征稿启事

《交易技术前沿》由上海证券交易所主管，上交所技术公司主办，以季度为单位发刊，主要面向全国证券、期货等相关金融行业的信息技术管理、开发、运维以及科研人员。

2021年三季度征稿主题如下：

一、云计算

（一）云计算架构

主要包含但不限于：云架构剖析探索，云平台建设经验分享，云计算性能优化研究。

（二）云计算应用

主要包含但不限于：云行业格局与市场发展趋势分析，国内外云应用热点探析，金融行业云应用场景与实践案例。

（三）云计算安全

主要包含但不限于：云系统下的用户隐私、数据安全探索，云安全防护规划、云安全实践，云标准的建设、思考与研究。

二、人工智能

（一）应用技术研究

主要包含但不限于：语音识别与自然语言处理，计算机视觉与生物特征识别，机器学习与神经网络，知识图谱，服务机器人技术。

（二）应用场景研究

主要包含但不限于：智能客服、语音数据挖掘、柜员业务辅助等。

主要包含但不限于：监控预警、员工违规监控、交易安全等。

主要包含但不限于：金融预测、反欺诈、授信、辅助决策、金融产品定价、智能投资顾问等。

主要包含但不限于：金融知识库、风险控制等。

主要包含但不限于：机房巡检机器人、金融网点服务机器人等。

三、数据中心

（一）数据中心的迁移

主要包含但不限于：展示数据中心的接入模式和网络规划方案；评估数据中心技术合规性认证的必要性；分析数据中心迁移过程中的影响和业务连续性；探讨数据中心迁移的实施策略和步骤。

（二）数据中心的运营

主要包含但不限于：注重服务，实行垂直拓展模式；注重客户流量，实行水平整合模式；探寻数据中心运营过程中降低成本和提高服务质量的途径。

四、分布式账本技术（DLT）

（一）主流分布式账本技术的对比

主要包含但不限于：技术架构、数据架构、应用架构和业务架构等。

(二) 技术实现方式

主要包含但不限于：云计算 + 分布式账本技术、大数据 + 分布式账本技术、人工智能 + 分布式账本技术、物联网 + 分布式账本技术等。

(三) 应用场景和案例

主要包含但不限于：结算区块链、信用证区块链、票据区块链等。

(四) 安全要求和性能提升

主要探索国密码算法在分布式账本中的应用，以及定制化的硬件对分布式账本技术性提升的作用等。

五、信息安全与 IT 治理

(一) 网络安全

主要包括但不限于：网络边界安全的防护、APT 攻击的检测防护、云安全生态的构建、云平台的架构及网络安全管理等。

(二) 移动安全

主要包括但不限于：移动安全管理、移动互联网接入的安全风险、防护措施等。

(三) 数据安全

主要包括但不限于：数据的分类分级建议、敏感数据的管控、数据共享的风险把控、数据访问授权的思考等。

(四) IT 治理与风险管理

主要包括但不限于：安全技术联动机制、自主的风险管理体系、贯穿开发全生命周期的安全管控、安全审计的流程优化等。

六、交易与结算相关

(一) 交易和结算机制

主要包含但不限于：交易公平机制、交易撮合机制、量化交易、高频交易、高效结算、国外典型交易机制等。

(二) 交易和结算系统

主要包含但不限于：撮合交易算法、内存撮合、双活系统、内存状态机、系统架构、基于新技术的结算系统等。

投稿说明

1、本刊采用电子投稿方式，投稿采用 word 文件格式（格式详见附件），请通过投稿邮箱 ftt.editor@sse.com.cn 进行投稿，收到稿件后我们将邮箱回复确认函。

2、稿件字数以 4000-6000 字左右为宜，务求论点明确、数据可靠、图表标注清晰。

3、本期投稿截止日期：2021 年 9 月 30 日。

4、投稿联系方式 021-68607128, 021-68607131 欢迎金融行业的监管人员、科研人员及技术工作者投稿。稿件一经录用发表，将酌致稿酬。

附件：投稿格式

标题（黑体 二号 加粗）

作者信息（姓名、工作单位、邮箱）（仿宋 GB2312 小四）

摘要：（仿宋 GB2312 小三 加粗）

关键字：（仿宋 GB2312 小三 加粗）

一、概述（仿宋 GB2312 小三 加粗）

二、一级标题（仿宋 GB2312 小三 加粗）

（一）二级标题（仿宋 GB2312 四号 加粗）

1、三级标题（仿宋 GB2312 小四 加粗）

（1）四级标题（仿宋 GB2312 小四）

正文内容（仿宋 GB2312 小四）

图：（标注图 X. 仿宋 GB2312 小四）

正文内容（仿宋 GB2312 小四）

表：（标注表 X. 仿宋 GB2312 小四）

正文内容（仿宋 GB2312 小四）

三、结论 / 总结（仿宋 GB2312 小三 加粗）

四、参考文献（仿宋 GB2312 小四）

杂志订阅与反馈

各位读者，如您想订阅《交易技术前沿》纸质版，欢迎扫描右侧二维码填写问卷进行订阅，同时可以向我们提出关于《交易技术前沿》的建议与意见反馈。如您希望赏阅电子版，欢迎访问我们的电子平台 <http://www.sse.com.cn/services/tradingservice/tradingtech/sh/transaction/>（或扫描封面尾页二维码）。我们的电子平台不仅同步更新当期的文章，同时还提供往期所有历史发表文章的浏览与查阅，欢迎关注！





扫描在线浏览

联系电话：021-68607128

021-68607131

投稿邮箱：ftt.editor@see.com.cn

ITRDC

证券信息技术研究发展中心（上海）



中国上海浦东南路528号

邮编：200120

公众咨询服务热线：4008888400

网址：<http://www.sse.com.cn>

内部资料 免费交流

本资料仅为内部交流使用，本季度印650册，编印单位为上交所技术有限责任公司，面向证券期货行业发送，印刷日期为2021年4月，印刷单位主人印刷厂。

部分图片或文字来源于互联网等公开渠道，其版权归属原作者所有。如有版权相关事宜，请发送邮件至ftt.editor@sse.com.cn