

摘要

本文研究了基于Black-Scholes模型的离散时间版本的Delta对冲策略与理论上的Delta对冲策略之间的误差，该误差随着离散时间点的抽样次数的增加而减小。由于实际交易中交易成本一类的问题，所以在一定的误差风险范围内，我们希望能够用最少的交易次数来完成既定的交易策略。给定抽样次数，我们使用蒙特卡洛方法计算出该误差在某一可接受范围内的概率。若该概率能够高于一个事先确定可接受值，则我们认为这个抽样次数是可接受的。本文利用高性能计算在Intel最新的MIC架构上计算出了最优的离散时间点的抽样次数。我们首先通过理论给出最优抽样次数的上界，然后通过二分法在由该上界给出的闭区间内搜索最优解。由于本文所讨论的数据依赖性本质上源自于独立高斯分布的随机数组成的数组，我们对此给出了两种并行思路。在我们的第一种并行思路中，我们只需要生成一条随机流，每次计算某段子集的时候空闲的线程利用随机数发生器生成下一个紧邻子集对应的随机数。所有随机数由全部线程共享，从而解除了数据的依赖性。在第二种并行思路中，每个线程保有一条私有的随机数流，根据自己计算的进度按需生成相应的随机数。这种情况下对于数据依赖性的处理则在于随机数的“种子”。

章节 1

背景综述

高性能计算(High Performance Computing)一直是计算机科研领域的重要主题。伴随近20年来超级计算集群 性能和数量的指数式增长, 我们的超算能力已经迈进Petaflop时代 (10^{15} 次浮点运算), 更强大的 exaflop时代(10^{18} 次浮点运算)也即将到来。但是目前能够有效充分利用超算集群的应用程序还为数 不多, 因此超算应用是制约HPC界发展的一大瓶颈。

在超算的应用领域, 科研活动占据了一大部分, 如天文学, 粒子物理, 计算化学等基础学科; 另外在工程和 商业领域也有重要的应用, 如汽车, 飞机设计中得计算流体力学模拟仿真, 油田勘探中的海量信息处理和开采 方案设计以及在商业投资领域的高频交易等。此外随着云计算和大数据相关研究和商业应用的快速崛起, 怎样将高性能计算和数据挖掘分析等技术结合起来也成为了当前超算领域的研究热点。本文的主题是探讨 高性能计算在金融领域的一个重要应用以及其在Intel最新的MIC架构上的实现。首先在中我们将简单介绍 高性能计算和金融领域超算应用的一些基本情况。

1.1 高性能计算的诞生和发展

高性能计算的概念和实现依赖于算法理论和硬件制造的发展。在算法理论方面, 并行运算(Parallel computing) 的研究在70年代就开始, 但是受制于当时硬件领域的发展瓶颈, 并行运算方面的很多理论没有得到充分地发展。如图 1.1 所示, Intel的处理器的在2000年之前晶体管的数量和时钟频率都能依照摩尔定律增长。但是当单个处理器的制造工艺到达纳米级别后, 受制于量子效应已经无法进一步提高单个处理器的时钟频率。随后业界开始转而使用多核系统(Multi-core system), 直至目前出现的众核系统(Manycore system)来提升超算的性能。目前最权威的HPC超算 集群排行榜Top500给出的结果显示, 无论是排名第一的机器还是所有500强计算能力的综合都较好地符合摩尔定律逐年增长。但是这些峰值运算速度仅仅代表了理论上一台超算集群能够达到的速度, 其测试结果也是通过相对简单地基准程序(benchmark)获得的, 而真正复杂的应用程序大多数 还未能超算集群上跑出好的使用效率。制约在超算上展开大规模应用的因素有以下几点。

1. 缺少能极大地提供并行性的算法
2. 数据间的通信受制于内存读取速度和处理器间的连接速度。
3. 从串行编程模型转到并行编程的学习成本

算法的并行性是最为重要的一点, 目前的大多数数学模型和对应的算法没有足够的并行性能够充分发挥超算集群的潜力。我们需要的是面向并行建立新的数学模型和 算法来解决实际问题。同时我们的编程模型和编程语言也需要相应地升级, 使之能更自然和原生地支持并行架构。这一点也被认为是同向Exascale计算的关键。而对于第二点, 目前出现的CPU/coprocessor架构已经迈出了重要的一步。Nvidia公司的GPU和GPGPU编程模型可以充分利用GPU的充足带宽(bandwidth)优化数据并行性德问题 (Data Parallel, SIMD), 而Intel公司则相应地推出了自己的协处理器Xeon Phi (Many Integrated Core Architecture, MIC Architecture)。Xeon Phi同样拥有强大的 带宽(382G/s)。此外Xeon Phi在编程难度上相比于GPGPU更有利于初学者掌握, 实际上用户可以相对简单地从CPU的串行程序过渡到Xeon Phi的并行代码上。本课题的应用即在MIC架构上展开, 并在数据通信等方面进行优化。

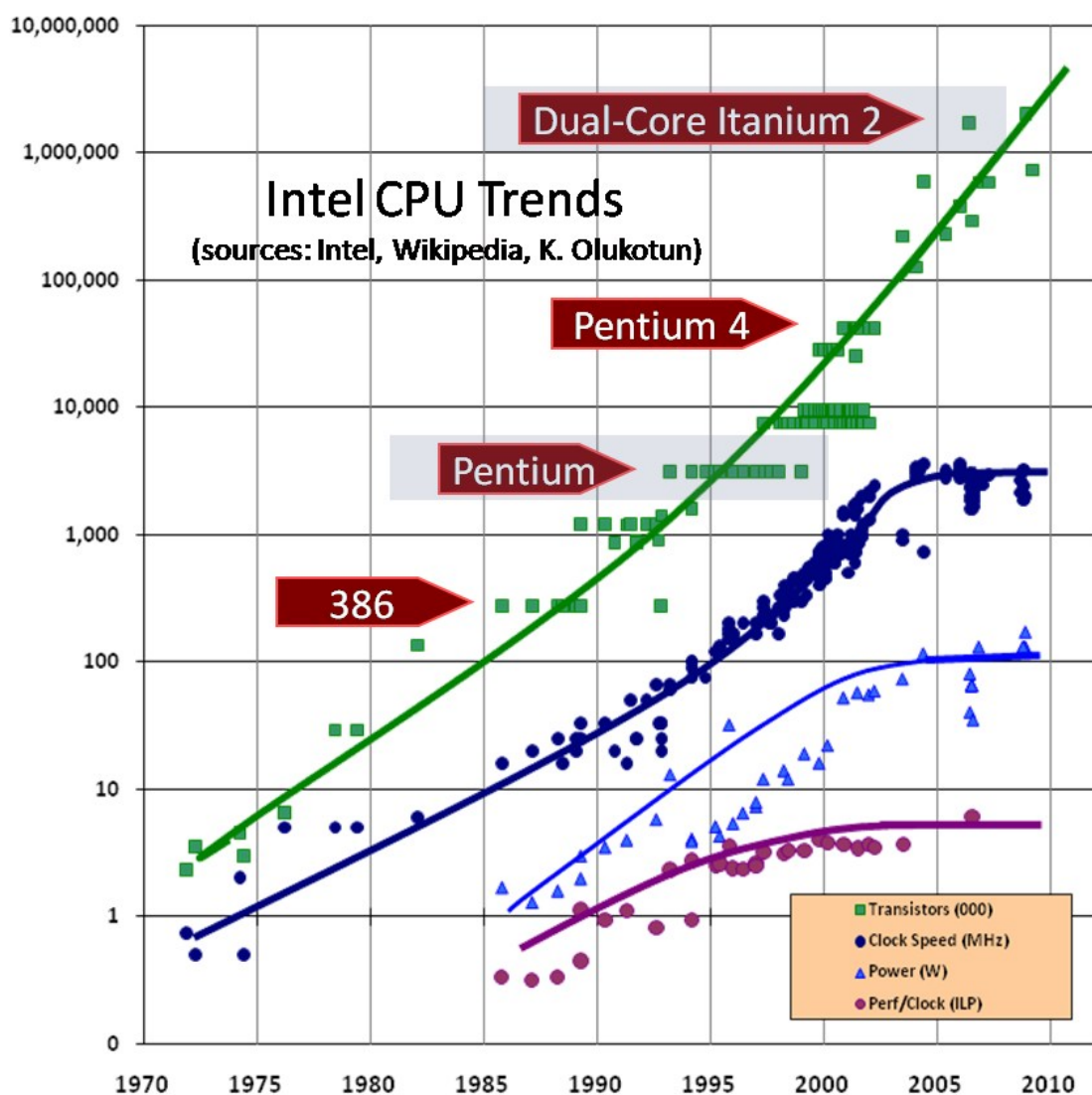


Figure 1.1: Intel处理器的发展趋势

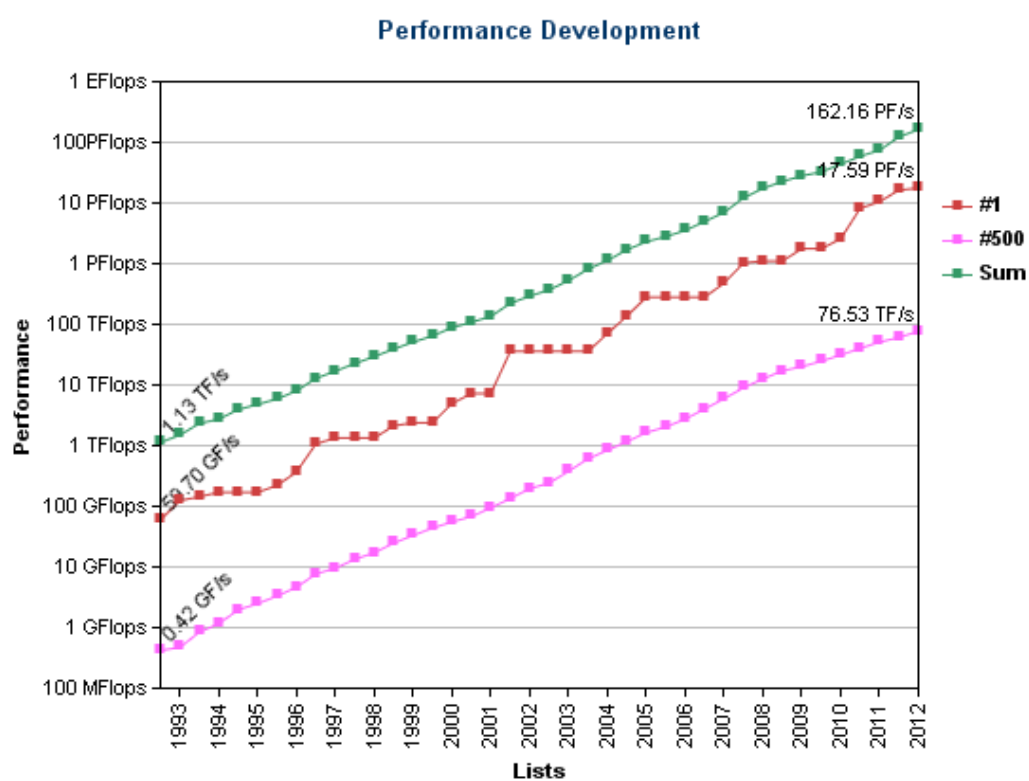


Figure 1.2: Top500超算集群的性能发展

1.2 金融领域的超算应用

当前的高性能计算在金融领域里的应用有如下几个方面：

1. 高频交易
2. 风险管理
3. 金融衍生品的定价
4. 定量经济分析

金融交易借助高性能计算能获得相对的竞争优势，同时也能降低交易风险。2007-2008 次贷危机的主要原因就是所用的金融工具，如担保债务凭证(CDO) 的风险 没有得到正确的认识。而造成这一误判的一个重要原因是模型中用来表示风险参数算法过于复杂，传统的计算机技术对此不能很快得到结果。从这个角度来说，高性能计算在金融中起着积极的作用因而高性能计算在金融中也越来越重要。

以美国高频交易为例，其所涉及的IT技术分成三部分：

- 交易数据在电脑外的加速，就是如何快速地把交易数据联接和推送到运行程序化交易策略的电脑。
- 交易数据在电脑内的加速，即如何在电脑内部把交易数据最快地传到交易策略程序。
- 其他数据的逻辑调用，即如何快速地获取程序化交易策略下单逻辑所需要的其他判断数据。

交易数据在电脑外的加速是一个花销不菲的任务，许多美国交易所已经实现了co-location 的业务，即客户可以支付一定费用将行情服务器、策略服务器等机器放在 交易所撮合引擎服务器所在的数据中心里。这一做法达到物理距离更近。交易所提供的数据支持主要以1G 和10G以太网接口为主。随着数据量的增加，以后也会应用40G的InfiniBand 以及更快的以太网。交易数据从交易所出来后， 交易公司通过自己使用高速路由器和交换机来获得进一步的加速。

如果程序化交易策略只交易一个品种，在一个交易所co-location就可以满足要求了，但如果策略需要做多个品种，例如期现套利，这就要涉及到两个数据中心的连接。美国期货的交易主要集中在中西部的芝加哥，股票现货交易主要在东北部的纽约，由于光速是固定的，最直的光纤带来最快的速度。美国网络提供商Spread Networks，曾经花了两年铺了纽约和芝加哥目前为止最直的一条网络， 该网络连接往返速度只需13.33毫秒（ms,千分之一秒），比原来的网络少了3毫秒。

交易数据从网络传到了电脑后，具体说是到了网卡的网端后，下一步的工作便是把数据传递到交易策略程序了。一般交易策略程序是作为操作系统的用户程序存在。交易数据主要通过UDP与TCP两种方式进行网络传输，下面以UDP包为例，（TCP 更复杂些，但原理差不多），简单介绍交易数据在电脑内的传输、解析。交易数据到达网卡后，通过PCI Express 或HyperPort接口，再传到操作系统的内核（OS Kernel）进行TCP/IP 协议栈的解析，再把解析后的数据内容通过socket 接口，推送到用户程序中。然后用户程序再对数据按照规定的格式进行解析，然后应用在策略逻辑中，进行分析，下单等动作。以上的这种方式主要是基于现有电脑的结构的基本做法，配置好的快些，但总的来说整个过程的处理时间，数量级在两位数微秒（us，百万分之一秒）左右。数据量多的时候，操作系统内核对TCP/IP 的解析所用CPU也消耗很多，处理速度会加速下降。一般来说，1G 的数据需要一个GHZ的CPU进行解析，一般的CPU的速度为2-3GHZ，故而消耗是相当客观的。在整个交易数据内部传递与解析的过程中，能够优化加速的主要是TCP/IP 协议的解析和业务数据的解析。这主要由专用的网卡来做。目前有两种方法来实现，一种是在网卡附带的CPU上进行处理；另一种是网卡+FPGA的形式，用FPGA（现场可编程门阵列）来做TCP/IP的解析和业务数据的解析。FPGA 因为全部是硬件，速度会更快些。理论上来说，用ASIC（专用集成电路）会更快些，但是由于经济成本方面的考虑，目前尚未有实际应用。通过以上的这些硬件加速的方法，基本上能达到个位数微秒（us）的延迟。用硬件做的好处除了延迟小外，在处理的容量上也有诸多好处。硬件能够做到无论在数据多少都达到固定的性能，这是基于电脑操作系统结构没法做到的。

剩下的便是对程序化交易策略的分析了。就技术难点而言，主要是对大量数据的分析，尤其是对大量时间序列数据的保存和快速访问。通用办法是用平面文件或者关系型的数据库如ORACLE, MY SQL, SQL Server。这种办法主要瓶颈在于磁盘硬碟的读写的IO速度上。现在有些基于内存的数据库（In-memory database），可以大大加快读写速度。

另外，高频交易所用的其它周边系统还包括主要是来传递各个子系统间的信息传送的信息传输系统和主要用来保存每天产生的海量数据以及这些数据的读写的数据储存系统。单单把美国股票市场的数据保存，每天就需要超过100G，一个月超过2T。如果要保存多市场多品种的数据，数据量更是庞大。这些技术难点和瓶颈都可能是HPC的用武之地。

最后，由于许多高频的交易的做法和思路都比较公开，所以一个成功的高频交易策略，除了在交易算法和细节处理之外，更多的是在于他所使用的交易系统的处理能力和速度上。确切的说，高频策略的成功与否，很大程度上取决于能否在众多竞争者中抢到自己要的单子。美国高频交易者往往需要耗巨资搭建交易网络和购买最好的硬件，使自己能够得到最快的数据和交易通道。同时，他还要不停地跟踪策略的交易情况，调整策略算法逻辑，不断地跟踪策略对行情的反应时间、下单的速度以及交易延迟的情况，试着把策略的反应时间再缩短1微秒。

章节 2

课题陈述

本文所要研究的金融问题是基于欧洲式期权的离散时间对冲策略的风险控制。欧式期权(European Option) 是期权的一种类型。另外两种期权类型是美式期权 (American Option)和亚洲期权(Asian Option)。期权本质上是一种金融合约，它授予期权的拥有者（买方）一种在某一特定时间点或之前，以某一个在合约中给出的价格 (strike price) 购买或者出售某一种金融产品(underlying asset) 的权利。买家在执行自己的权利时，卖家必须按照合约规定的价格进行交易，但买家同时需要付给买家一定的额外交易手续费(premium)。期权的研究主要集中在对其价值的评估。期权的价值一般可以分为两个部分。一是期权的内部价值 (intrinsic value)，主要由该期权基于的金融产品在交易时的市场价格和合约定价的差值决定。二是期权的时间价值 (Time Value)，这一部分取决于交易前的其他因素如 利息，增值等。期权的研究从19世纪就开始了，但目前的研究一般都是基于1973年提出的Black-Scholes模型。我们的课题也是基于Black-Scholes模型，并对欧式期权 对冲策略的风险控制进行研究。

2.1 欧式期权

欧式期权的特点在于买方必须在合约规定的时间到期时才能行使权力。通常将欧式期权的合理价格价定为其期望价值。相比与美式期权，二者在执行时间上有所不同。美式期权合同在到期日前的任何时候或在到期日都可以执行合同，结算日则是在履约日之后的一天或两天，大多数的美式期权合同允许持有人在交易日到履约日之间随时履约，但也有一些合同规定一段比较短的时间可以履约，如“到期日前两周”。欧式期权合同要求其持有者只能在到期日履行合同，结算日是履约后的一天或两天。目前国内的外汇期权交易都是采用的欧式期权合同方式。通过比较，结论是：欧式期权本少利大，但在获利的时间上不具灵活性；美式期权虽然灵活，但付费十分昂贵。因此，国际上大部分的期权交易都是欧式期权。中国国内银行设立的外汇期权业务都是采用的欧式期权交易。

2.2 Black-Scholes模型

Black-Scholes 是一个70年代由Fischer Black和Myron Scholes 提出的用于金融衍生品交易的数学模型。这个模型可以推出Black-Scholes公式，并用于预测欧式期权定价的理论值。70年代至今在众多的实际交易中，该公式的预测估值被认为是非常接近实际观测到的价值。这个模型的成功也直接导致了期权交易市场的繁荣，并能使芝加哥期权交易市场等大的期权交易场所得到科学的有效的监管。Black-Scholes 公式是一个偏微分方程，用于预测期权价格随时间的变化。而该模型背后的策略则是通过合理地买卖期权来对冲风险，这种对冲策略被 成为"Delta Hedging"，这也是许多更复杂的对冲机构策略的基础。此后Robert C. Merton 进一步发展了该模型，并和Schole分享了1997年的诺贝尔经济学奖。

2.3 课题表述

本课题研究的对象就是欧式期权利用对冲策略来降低交易的风险。我们考虑基于Black-Scholes模型的Delta 对冲策略，该策略为一个连续时间策略，也即策略所要求的交易行为发生在每时每刻，关于时间轴连续。我们知道现实生活中，交易只能发生在一些离散的时间点上，即交易员只能在一系列相继的时刻点上交易。实际中想要完美的复制实现一个给定的连续时间策略是不可能的。本课题研究的就是离散时间版本的Delta 对冲策略与理论上的Delta 对冲策略之间的误差及风险控制。文章[1, 3, 4] 研究了连续时间和离散时间金融交易模型的关系，但这些文章更多的关注了两者的渐进性质而没有考虑如何最优的复制连续时间策略。本文将弥补这一空白。

在一般化的Black-Scholes模型中，假设在时刻 t ，期权所基于的风险资产价格为 X_t ，另外有一无风险资产价格为 S_t^0 。 X_t 满足如下的一维随机偏微分方程：

$$dX_t = \mu(X_t)X_t dt + \sigma(X_t)X_t dB_t, \quad (2.1)$$

$X_0 = x_0 > 0, \sigma(x) > 0, \forall x > 0$ 而无风险资产价格满足一个常微分方程：

$$dS_t^0 = rS_t^0 dt \quad (2.2)$$

。我们引进一个新的随机过程 W_t ：

$$W_t = B_t + \int_0^t \frac{\mu(X_s) - r}{\sigma(X_s)} ds \quad (2.3)$$

，在中性风险的概率空间下，这个随机过程是一个标准的布朗运动。我们在此中性风险概率空间下考虑问题。现在风险资产所满足的随机微分方程变为如下形式：

$$dX_t = rX_t dt + \sigma(X_t)X_t dW_t \quad (2.4)$$

我们假设考虑的问题满足如下两个条件：

(\mathcal{H}_1) μ 和 σ 有限的，两次连续可微且二次偏微分满足Holder条件。更准确的说，令 $\hat{\mu}(x) = \mu(\exp(x))$ ， $\hat{\sigma}(x) = \sigma(\exp(x))$ ，且 $\delta \in (0, 1)$ ， $K > 0$ ，对任意 $(x, x') \in \mathbb{R}^2$ ，我们有：

$$|\hat{\mu}'(x)| + |\hat{\mu}''(x)| + \frac{|\hat{\mu}''(x) - \hat{\mu}''(x')|}{|x - x'|^\delta} + |\hat{\sigma}'(x)| + |\hat{\sigma}''(x)| + \frac{|\hat{\sigma}''(x) - \hat{\sigma}''(x')|}{|x - x'|^\delta} < K. \quad (2.5)$$

(\mathcal{H}_2) 存在 $\sigma_0 > 0$ 使得对任意 $x > 0$ 有 $|\sigma(x)| \geq \sigma_0$ 。

在这两个条件下若欧洲期权的回报函数为 $f \in L^2(X_t)$ ，则该期权在0时刻的价格由下面公式给出：

$$h(f)(x_0) = \mathbb{E}(\exp(-rT)f(X_T)|X_0 = x_0). \quad (2.6)$$

令

$$u(t, x) = \mathbb{E}(e^{-r(T-t)}f(X_{T-t}^x)), \quad (2.7)$$

则 u 是如下方程的解：

$$-\frac{\partial}{\partial t}u(t, x) = \frac{1}{2}\sigma^2(x)x^2 \frac{\partial^2}{\partial x^2}u(t, x) + rx \frac{\partial}{\partial x}u(t, x) - ru(t, x) \quad (2.8)$$

这里 $(t, x) \in [0, T) \times (0, \infty)$ 且 $u(0, x) = h(f)$ ， $u(T, x) = f(x)$ 。

期权定价公式如下：

$$e^{-rT}f(X_T) = h(f) + \int_0^T \xi_t d\tilde{X}_t \quad (2.9)$$

这里 $\tilde{X}_t = e^{-rT}X_t$ 是风险资产的折旧贴现价格。Itô 公式给出了Delta 对冲策略 ξ ：

$$\xi_t = \frac{\partial u}{\partial x}(t, X_t) \quad (2.10)$$

换句话说，为了达到完美对冲，交易员需要在 $t \in [0, T]$ 的时间内无时不刻的交易，并且使自己在时刻 t 持有恰好 ξ_t 单位的风险资产。这在现实中是不可能达到的。

一个替代的解决方案是用在离散的时刻点交易的方法来逼近原先的对冲策略。假设交易员在 $t \in [0, T)$ 时间内交易 N 次, 交易时间被定义为 $t_k = kT/N$, ($k \in \{0, \dots, N\}$), 在 $t \in [t_k, t_{k+1})$ 时间内, 交易员使自己持有 ξ_{t_k} 单位的风险资产 X_t 。在此离散化策略的逼近下, 在期权到期时刻 T , 完美对冲和离散化对冲的差值为:

$$\begin{aligned}
 M_T^N &= e^{-rT} f(X_T) - (u(0, x) + \int_0^T \frac{\partial u}{\partial x}(\varphi(t), X_{\varphi(t)}) d\tilde{X}_t \\
 &= \int_0^T \frac{\partial u}{\partial x}(t, X_t) d\tilde{X}_t - \int_0^T \frac{\partial u}{\partial x}(\varphi(t), X_{\varphi(t)}) d\tilde{X}_t \\
 &= (X_T - K)^+ - \mathbb{E}[(X_T - K)^+] - \int_0^T \frac{\partial u}{\partial x}(\varphi(t), X_{\varphi(t)}) dX_t \\
 &= (X_T - K)^+ - x_0 N(d_1(0)) + KN(d_2(0)) - \sum_{i=0}^{n-1} N(d_1(t_i))(X_{t_{i+1}} - X_{t_i}) \\
 &= (X_T - K)^+ - \frac{x_0}{\sqrt{2\pi}} \int_{-\infty}^{\frac{\log(\frac{x_0}{K}) + \frac{1}{2}\sigma^2 T}{\sigma\sqrt{T}}} e^{-\frac{v^2}{2}} dv + \frac{K}{\sqrt{2\pi}} \int_{-\infty}^{\frac{\log(\frac{x_0}{K}) - \frac{1}{2}\sigma^2 T}{\sigma\sqrt{T}}} e^{-\frac{v^2}{2}} dv \\
 &\quad - \sum_{i=0}^{n-1} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{\log(\frac{X_{t_i}}{K}) + \frac{1}{2}\sigma^2(T-t_i)}{\sigma\sqrt{T-t_i}}} e^{-\frac{v^2}{2}} dv (X_{t_{i+1}} - X_{t_i})
 \end{aligned} \tag{2.11}$$

这里 $\varphi(t) = \sup\{t_i | t_i \leq t\}$.

对于该误差 M_T^N , 我们已知它有如下的渐进性质:

定理 2.3.1 M_T^N 在概率上趋向于0:

$$M_T^N \xrightarrow[N \rightarrow +\infty]{\mathbb{P}} 0 \tag{2.12}$$

证明: 见[6]

定理 2.3.2 令 X_t 随机偏微分方程 $dX_t = \sigma(X_t)X_t dW_t$ 的解所对应的随机过程, u 如前文定义, 则我们有

$$\sqrt{n}M_T^N \xrightarrow[N \rightarrow +\infty]{d} \hat{W}_\tau \tag{2.13}$$

这里 $\tau = \frac{1}{2} \int_0^T (\frac{\partial^2 u}{\partial x^2}(t, X_t))^2 \sigma^4(X_t) X_t^4 dt$, 且 \hat{W} 是一个独立于 τ 另一个布朗运动。

证明: 见[7]

上面的两条渐进性质不仅告诉我们随着 N 的增大, 离散对冲策略的逼近将越来越成功, 两者相差的误差将在概率意义下趋于零, 而且还告诉我们此误差的渐进表现 类似于一个系数逐渐变小的布朗运动。特别的, 我们从中可知, 通过增大 N , 在概率意义下, 离散化的Delta 对冲策略可以任意程度的逼近完美Delta 对冲策略。

由于实际交易中交易成本一类的问题, 所以在一定的交易风险范围内, 我们希望能够用最少的交易次数来完成既定的交易策略。也即我们希望找到一个最小的交易次数 N 来最优化交易成本。假设在 $t = T$ 时, 交易的误差可表示为 $|P(X) - P_{T_N}(X)|$, 而可接受的最大误差为 ϵ , 我们将采用蒙特卡洛模拟算法, 通过 M 次模拟, 计算出 $|P(X) - P_{T_N}(X)| < \epsilon$ 的概率 $Prob(M, N)$, 当 $Prob(M, N)$ 的值大于一个预先设定的概率值时, 我们认为 N 是一个可接受的抽样次数。算法1给出了我们计算验证一个 N 值的串行算法伪代码。

Algorithm 1 欧式期权对冲策略误差控制的串行算法

输入参数 X_0	▷ 金融产品在 $t = 0$ 时的初始价格
输入参数 σ	▷ 市场波动性
输入参数 K	▷ 期权合约价格
输入参数 T	▷ 期权到期时间
输入参数 ϵ	▷ 可接受的误差上限
输入参数 $Prob$	▷ 离散策略误差可接受概率下限
输入参数 N_0	▷ 理论给出的初始离散抽样次数上界
输入参数 M	▷ 蒙特卡洛模拟次数

```

1: procedure BSERROR
2:    $error \leftarrow 0$ 
3:    $NRV[N]$ 
4:    $BM[N]$ 
5:    $PX[N + 1]$ 
6:    $\delta t \leftarrow T/N$ 
7:    $count \leftarrow 0$ 
8:    $N \leftarrow N_0$ 
9:   for  $m = 1 : M$  do
10:     $error \leftarrow 0$ 
11:    for all  $NRV[j]$  do  $NRV[j] \leftarrow GaussianNumGenerator()$ 
12:    end for
13:     $BM[0] \leftarrow NRV[0] \cdot \sqrt{\delta t}$ 
14:    for all  $BM[j]$  do  $BM[j] \leftarrow BM[j - 1] + NRV[j] \cdot \sqrt{\delta t}$ 
15:    end for
16:     $PX[0] \leftarrow X_0$ 
17:    for all  $j = 1 : N + 1$  do
18:       $PX[j] \leftarrow X_0 \cdot \exp(-0.5 \cdot \sigma^2 \cdot j \cdot \delta t + \sigma \cdot BM[j - 1])$ 
19:    end for
20:    for all  $j = 0 : N - 1$  do
21:       $Upper \leftarrow (\log(PX[j]/K) + 0.5 \cdot \sigma^2 \cdot (T - T_j)) / (\sigma \cdot \sqrt{T - T_j})$ 
22:       $error \leftarrow error - 1/(\sqrt{2\pi}) \cdot (PX[j + 1] - PX[j]) \cdot \int_{-\infty}^{Upper} e^{-t^2/2} dt$ 
23:    end for
24:    if  $PX[N] > K$  then
25:       $error \leftarrow error + (PX[N] - K)$ 
26:    end if
27:     $Upper1 \leftarrow (\log(X_0/K) + 0.5 \cdot \sigma^2 \cdot T) / (\sigma \sqrt{T})$ 
28:     $Upper2 \leftarrow (\log(X_0/K) - 0.5 \cdot \sigma^2 \cdot T) / (\sigma \sqrt{T})$ 
29:     $error = error + K / (\sqrt{2\pi}) \cdot \int_{-\infty}^{Upper2} e^{-t^2/2} dt - X_0 / (\sqrt{2\pi}) \cdot \int_{-\infty}^{Upper1} e^{-t^2/2} dt$ 
30:    if  $error < \epsilon$  then
31:       $count \leftarrow count + 1$ 
32:    end if
33:  end for
34:   $prob \leftarrow count/M$ 
35:  if  $prob < Prob$  then
36:     $N \leftarrow N + 1$ 
37:  else
38:     $N \leftarrow N - 1$ 
39:  end if
40:  return  $N$ 
41: end procedure

```

章节 3

参数选择及收敛条件

我们进行 M 次蒙特卡罗模拟实验，对于每一次模拟，选取某个 N ，计算相应的误差 M_T^N ，我们希望该误差小于可接受的最大误差 $\epsilon = X0 * 10^{-2}$ 。我们用蒙特卡洛的方法计算 $M_T^N \leq \epsilon$ 的概率，并希望该概率能超过某一个特定值： $Prob = 95\%$ 。在这个问题中 M 和 N 的初始值以及其更新方法至关重要。对 M 而言，一方面较大的 M 会增大蒙特卡罗方法以及通过此方法计算出来的概率的可信度，另一方面 M 过大，将不可避免的造成更高的资源和硬件的要求，如何在保持一定可行度的条件下选择一个较小的 M 需要一个精巧的方法。对 N 而言，较大的 N 会使得逼近更准确，从而误差 M_T^N 较小，但也会造成资源的浪费和迭代收敛的过程冗长。较大或者较小的 M 和 N 的初始值将导致收敛缓慢甚至不收敛。如何选择合适的 M 和 N 的初始值，如何让它们互相迭代更新，以及怎样判断收敛与否是本章所需要解决的问题。

3.1 参数优化

关于 N 的初始值的选取，一个首要的问题是，数值模拟中， M 会影响我们对 N 的判断，甚至一个选取不好的 M 会给出错误的 N 的更新方向，从而导致问题得不到收敛。为了避免这种问题，我们在这里给出一种不依赖于 M 的 N 的初始值选定方法。

我们需要一个尽可能小的 N ，使得存在某个关于 N 的函数 $P(N)$ ，如下不等式成立：

$$Probability(M_T^N \geq \epsilon) = \mathbb{P}(M_T^N \geq \epsilon) < P(N)。 \quad (3.1)$$

若我们能发现这样的函数 $P(N)$ ，则所有满足不等式：

$$P(N) \leq 1 - Prob = 5\% \quad (3.2)$$

的正整数 N ，都满足 $\mathbb{P}(M_T^N < \epsilon) > Prob = 95\%$ 。我们只需要取满足不等式的最小的正整数 N 即可。此时 N 的选取独立于 M ，也即无论我们选取怎样的 M ，理论上都会有 $\mathbb{P}(M_T^N < \epsilon) > Prob$ 成立。

为了寻找函数 $P(N)$ ，我们先引入两个引理：

引理 3.1.1 (Burkholder-Davis-Gundy inequality) 令 $T > 0$ ， $(M_t)_{0 \leq t \leq T}$ 为一个局部的鞅，且 $M_0 = 0$ 。则对于所有的 $0 < p < +\infty$ ，存在独立于 T 和 $(M_t)_{0 \leq t \leq T}$ 的常数 c_p ， C_p 使得

$$c_p \mathbb{E}[\langle M \rangle_T^{\frac{p}{2}}] \leq \mathbb{E}[(\sup_{0 \leq t \leq T} |M_t|)^p] \leq C_p \mathbb{E}[\langle M \rangle_T^{\frac{p}{2}}] \quad (3.3)$$

证明：见[5]。

在文章中[2]中，作者证明了 $C_p \leq (2\sqrt{p})^p$ 。

引理 3.1.2 (Generalized Minkowski inequality) 设 (S_1, μ_1) 和 (S_2, μ_2) 是两个可测空间，且 $F : S_1 \times S_2 \rightarrow R$ 为可测映射。则广义的Minkowski 积分不等式是：

$$(\int_{S_2} |\int_{S_1} F(x, y) d\mu_1(x)|^p d\mu_2(y))^{\frac{1}{p}} \leq \int_{S_1} (\int_{S_2} |F(x, y)|^p d\mu_2(y))^{\frac{1}{p}} d\mu_1(x), \forall p \geq 1 \quad (3.4)$$

证明: 见[8]。

不失一般性, 在以下的计算中我们设 $r = 0$, $\sigma(X_t) = \text{constant} = \sigma > 0$, 回报函数 $f(x) = (x - K)^+$, $K > 0$ 。

由Chebyshev不等式, 我们有:

$$\mathbb{P}(M_T^n \geq \gamma) \leq \frac{\mathbb{E}[(M_T^n)^p]}{\gamma^p}, \forall p > 0 \quad (3.5)$$

再由 Burkholder-Davis-Gundy 不等式有:

$$\mathbb{E}[(M_T^n)^p] \leq C(p) \mathbb{E}[\langle M^n \rangle_T^{\frac{p}{2}}] \quad (3.6)$$

这里我们用已知最优的结果 $C(p) = (2\sqrt{p})^p$ 。

对方程 (2.7), 关于 x 微分, 我们有:

$$\frac{\partial^2 u}{\partial t \partial x}(t, x) + \frac{1}{2} \sigma^2 x^2 \frac{\partial^3 u}{\partial x^3}(t, x) = -\sigma^2 x \frac{\partial^2 u}{\partial x^2}(t, x) \quad (3.7)$$

对 $\frac{\partial u}{\partial x}(t, X_t) - \frac{\partial u}{\partial x}(\varphi(t), X_{\varphi(t)})$, 从 $\varphi(t)$ 到 t 使用 Itô 公式。结合方程 (3.7), 我们有:

$$\begin{aligned} & \frac{\partial u}{\partial x}(t, X_t) - \frac{\partial u}{\partial x}(\varphi(t), X_{\varphi(t)}) \\ &= \int_{\varphi(t)}^t \left(\frac{\partial^2 u}{\partial x \partial t}(\theta, X_\theta) + \frac{1}{2} \frac{\partial^3 u}{\partial x^3}(\theta, X_\theta) \sigma^2 X_\theta^2 \right) d\theta + \int_{\varphi(t)}^t \frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta dW_\theta \\ &= \int_{\varphi(t)}^t \frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta (dW_\theta - \sigma d\theta) \end{aligned} \quad (3.8)$$

所以

$$M_T^n = \int_0^T \int_{\varphi(t)}^t \frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta (dW_\theta - \sigma d\theta) \sigma X_t dW_t \quad (3.9)$$

$$\langle M^n \rangle_T = \int_0^T \left(\int_{\varphi(t)}^t \frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta (dW_\theta - \sigma d\theta) \right)^2 \sigma^2 X_t^2 dt \quad (3.10)$$

我们假设 $p \geq 2$ 。由广义Minkowski不等式, 我们有:

$$\begin{aligned} \mathbb{E}[\langle M^n \rangle_T^{\frac{p}{2}}] &= \mathbb{E}\left[\left(\int_0^T \left(\int_{\varphi(t)}^t \frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta (dW_\theta - \sigma d\theta)\right)^2 \sigma^2 X_t^2 dt\right)^{\frac{p}{2}}\right] \\ &\leq \left(\int_0^T \left(\mathbb{E}\left[\left(\int_{\varphi(t)}^t \frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta (dW_\theta - \sigma d\theta)\right)^p \sigma^p X_t^p\right]\right)^{\frac{2}{p}} dt\right)^{\frac{p}{2}} \end{aligned} \quad (3.11)$$

考虑一个满足下面条件的新的概率 $\tilde{\mathbb{Q}}$:

$$\frac{d\tilde{\mathbb{Q}}}{d\mathbb{Q}} = e^{\sigma W_t - \frac{\sigma^2 t}{2}} = \frac{X_t}{X_0} \quad (3.12)$$

则在此新的概率 $\tilde{\mathbb{Q}}$ 下, $\tilde{W}_t = W_t - \sigma t$ 是一个布朗运动。

由Cauchy 不等式, 在概率 $\tilde{\mathbb{Q}}$ 下, 我们有:

$$\mathbb{E}[\langle M^n \rangle_T^{\frac{p}{2}}] \leq x_0 \left(\int_0^T \left(\mathbb{E}_{\tilde{\mathbb{Q}}} \left[\left(\int_{\varphi(t)}^t \frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta d\tilde{W}_\theta \right)^{2p} \right] \right)^{\frac{1}{p}} \left(\mathbb{E}[\sigma^{2p} X_t^{2p-2}] \right)^{\frac{1}{p}} dt \right)^{\frac{p}{2}} \quad (3.13)$$

再次使用Burkholder-Davis-Gundy不等式和广义Minkowski不等式, 我们有:

$$\begin{aligned} \mathbb{E}[\langle M^n \rangle_T^{\frac{p}{2}}] &\leq x_0 \left(\int_0^T \left(C(2p) \mathbb{E}_{\tilde{\mathbb{Q}}} \left[\left(\int_{\varphi(t)}^t \left(\frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta \right)^2 d\theta \right)^p \right] \right)^{\frac{1}{p}} \left(\mathbb{E}[\sigma^{2p} X_t^{2p-2}] \right)^{\frac{1}{p}} dt \right)^{\frac{p}{2}} \\ &\leq x_0 C(2p)^{\frac{1}{2}} \left(\int_0^T \left(\int_{\varphi(t)}^t \left(\mathbb{E}_{\tilde{\mathbb{Q}}} \left[\left(\frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta \right)^{2p} \right] \right)^{\frac{1}{p}} d\theta \right) \left(\mathbb{E}[\sigma^{2p} X_t^{2p-2}] \right)^{\frac{1}{p}} dt \right)^{\frac{p}{2}} \end{aligned} \quad (3.14)$$

$$\begin{aligned}
\mathbb{E}[X_t^{2p-2}] &= X_0^{2p-2} \mathbb{E}[e^{(2\sigma W_t - \sigma^2 t)(p-1)}] \\
&= X_0^{2p-2} e^{2t(p-1)^2 \sigma^2 - p(t-1)\sigma^2} \\
&\leq X_0^{2p-2} e^{\sigma^2 p}, \forall p \in [2, +\infty)
\end{aligned} \tag{3.15}$$

定义 $\tilde{C}(p)$ 如下:

$$\tilde{C}(p) = X_0^{2p-2} e^{\sigma^2 p}, \forall p \in [2, +\infty) \tag{3.16}$$

另一方面我们有:

$$\mathbb{E}_{\tilde{\mathbb{Q}}}[(\frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta)^{2p}] = \frac{1}{(T-\theta)^p} \mathbb{E}_{\tilde{\mathbb{Q}}}[N(d_1(\theta))^{2p}] \tag{3.17}$$

这里

$$N(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, d_1(\theta) = \frac{1}{\sigma\sqrt{T-\theta}} (\log(\frac{X_\theta}{K}) + \frac{1}{2}(T-\theta)) \tag{3.18}$$

记 $d_2(\theta) = \frac{1}{\sigma\sqrt{T-\theta}} (\log(\frac{X_\theta}{K}) - \frac{1}{2}(T-\theta))$, 此记号将在后面用到。

由 $X_t = x_0 \exp(\sigma \tilde{W}_t + \frac{1}{2}\sigma^2 t)$, 我们有:

$$\mathbb{E}_{\tilde{\mathbb{Q}}}[(\frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta)^{2p}] = (\frac{1}{2\pi(T-\theta)})^p \mathbb{E}_{\tilde{\mathbb{Q}}}[\exp(-p(\frac{\log(\frac{x_0}{K}) + \frac{1}{2}\sigma^2 T + \sigma \tilde{W}_\theta}{\sigma\sqrt{T-\theta}})^2)] \tag{3.19}$$

定义常数 λ 如下: $\lambda = \log(\frac{x_0}{K}) + \frac{1}{2}\sigma^2 T$, 则不等式为:

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathbb{Q}}}[(\frac{\partial^2 u}{\partial x^2}(\theta, X_\theta) \sigma X_\theta)^{2p}] &= (\frac{1}{2\pi(T-\theta)})^p \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\theta}} e^{-\frac{z^2}{2\theta}} e^{-\frac{p}{T-\theta}(z+\frac{\lambda}{\sigma})^2} dz \\
&= (\frac{1}{2\pi(T-\theta)})^p \sqrt{\frac{T-\theta}{T-\theta+2p\theta}} e^{-\frac{p}{T-\theta+2p\theta} \frac{\lambda^2}{\sigma^2}} \\
&\leq \frac{\sqrt{T-\theta}}{(2\pi(T-\theta))^p} \frac{\sigma}{\lambda\sqrt{2p}} e^{-\frac{1}{2}}
\end{aligned} \tag{3.20}$$

所以

$$\begin{aligned}
\mathbb{E}[\langle M^n \rangle_T^{\frac{p}{2}}] &\leq x_0 C(2p)^{\frac{1}{2}} (\int_0^T (\int_{\varphi(t)}^t (T-\theta)^{\frac{1}{2p}-1} d\theta) \frac{1}{2\pi} (\frac{\sigma}{\sqrt{2p\lambda}} e^{-\frac{1}{2}})^{\frac{1}{p}} x_0^{2-\frac{2}{p}} e^{\sigma^2 t} dt)^{\frac{p}{2}} \\
&= x_0^p C(2p)^{\frac{1}{2}} (\frac{1}{2\pi})^{\frac{p}{2}} (\frac{\sigma}{\lambda\sqrt{2\pi}})^{\frac{1}{2}} e^{\frac{2\sigma^2 p-1}{4}} (\int_0^T \int_{\varphi(t)}^t (T-\theta)^{\frac{1}{2p}-1} d\theta dt)^{\frac{p}{2}}
\end{aligned} \tag{3.21}$$

$$\begin{aligned}
(\int_0^T \int_{\varphi(t)}^t (T-\theta)^{\frac{1}{2p}-1} d\theta dt)^{\frac{p}{2}} &= (2p \int_0^T ((T-\varphi(t))^{\frac{1}{2p}} - (T-t)^{\frac{1}{2p}}) dt)^{\frac{p}{2}} \\
&= (2p \frac{T}{n} \sum_{i=0}^{n-1} (T-t_i)^{\frac{1}{2p}} - \int_0^T (T-t)^{\frac{1}{2p}} dt)^{\frac{p}{2}} \\
&= (2p)^{\frac{p}{2}} (\frac{T}{n})^{\frac{2p+1}{4}} (\sum_{i=1}^n i^{\frac{1}{2p}} - \int_0^n (n-v)^{\frac{1}{2p}} dv)^{\frac{p}{2}} \\
&= (2p)^{\frac{p}{2}} (\frac{T}{n})^{\frac{2p+1}{4}} (\sum_{i=1}^n i^{\frac{1}{2p}} - \frac{2p}{2p+1} n^{\frac{2p+1}{2p}})^{\frac{p}{2}}
\end{aligned} \tag{3.22}$$

由于 $0 < \frac{1}{2p} = q < 1$, 我们知道函数 $g(x) = x^q$ 是凹函数(concave function)。由 $g(x)$ 的凹性质, 我们有:

$$\sum_{i=1}^{n-1} i^q + \sum_{i=1}^{n-1} \frac{(i+1)^q - i^q}{2} < \int_1^n x^q dx \tag{3.23}$$

所以

$$\begin{aligned} \sum_{i=1}^{n-1} i^q &< \frac{1}{q+1}(n^{q+1} - 1) - \frac{1}{2}(n^q - 1) \\ &< \frac{n^{q+1}}{q+1} - \frac{n^q}{2} \end{aligned} \quad (3.24)$$

$$\sum_{i=1}^n i^q = \sum_{i=1}^{n-1} i^q + n^q < \frac{n^{q+1}}{q+1} + \frac{n^q}{2} \quad (3.25)$$

$$\sum_{i=1}^n i^{\frac{1}{2p}} - \frac{2p}{2p+1} n^{\frac{2p+1}{2p}} < \frac{n^{\frac{1}{2p}}}{2} \quad (3.26)$$

因此:

$$\mathbb{E}[\langle M^n \rangle_T^{\frac{p}{2}}] \leq x_0^p C(2p)^{\frac{1}{2}} \left(\frac{1}{2\pi}\right)^{\frac{p}{2}} \left(\frac{\sigma}{\lambda\sqrt{2\pi}}\right)^{\frac{1}{2}} e^{\frac{2\sigma^2 p-1}{4}} (2p)^{\frac{p}{2}} \left(\frac{T}{n}\right)^{\frac{2p+1}{4}} \left(\frac{1}{2}\right)^{\frac{p}{2}} n^{\frac{1}{4}} \quad (3.27)$$

故而对于 $p \geq 2$,

$$\mathbb{P}(M_T^n \geq \gamma) \leq L(p) = C_0 \frac{1}{\gamma^p} C(p) x_0^p C(2p)^{\frac{1}{2}} \left(\frac{1}{2\pi}\right)^{\frac{p}{2}} e^{\frac{\sigma^2 p}{2}} p^{\frac{p}{2}} \left(\frac{T}{n}\right)^{\frac{p}{2}} \quad (3.28)$$

这里 $C_0 = T^{\frac{1}{4}} \left(\frac{\sigma}{\lambda\sqrt{2\pi}}\right)^{\frac{1}{2}} e^{-\frac{1}{4}}$ 是一个与 p, n 无关的常数。通过令等式 $\frac{\partial \log(L(p))}{\partial p} = 0$ 成立。我们找到最优的 p_* 。

$$\log(L(p)) = \frac{3}{2}p \log(p) + \log(C_0) - \frac{p}{2} \log(C_1 n) \quad (3.29)$$

这里 $C_1 = \frac{\gamma^2 \pi e^{-\sigma^2}}{16 x_0^2 T}$ 是与 p, n 无关的常数。

$$0 = \frac{\partial \log(L(p))}{\partial p} = \frac{1}{2}(3 + 3 \log(p) - \log(C_1 n)) \quad (3.30)$$

$$p = \frac{(C_1 n)^{\frac{1}{3}}}{e} \quad (3.31)$$

最优的 p 是 p_* :

$$p_* = \begin{cases} 2, & \log(C_1 n) < 3 \log(2) + 3 \\ \frac{(C_1 n)^{\frac{1}{3}}}{e}, & \log(C_1 n) \geq 3 \log(2) + 3 \end{cases} \quad (3.32)$$

因此 $L(p)$ 的最小值是:

$$L(p_*) = \begin{cases} \frac{8C_0}{C_1 n}, & \log(C_1 n) < 3 \log(2) + 3 \\ C_0 e^{-\frac{3(C_1 n)^{\frac{1}{3}}}{2e}}, & \log(C_1 n) \geq 3 \log(2) + 3 \end{cases} \quad (3.33)$$

现在我们得到了一个 N 的估界:

定理 3.1.1 假设如上文, 我们有:

$$\mathbb{P}(M_T^n \geq \gamma) \leq \begin{cases} \frac{128\sigma^{\frac{1}{2}} x_0^2 T^{\frac{5}{4}} e^{\sigma^2}}{e^{\frac{1}{4}} \gamma^2 \pi (\log(\frac{x_0}{K}) + \frac{1}{2} \sigma^2 T)^{\frac{1}{2}} (2\pi)^{\frac{1}{4}}} \frac{1}{n}, & \log(C_1 n) < 3 \log(2) + 3 \\ T^{\frac{1}{4}} \left(\frac{\sigma}{(\log(\frac{x_0}{K}) + \frac{1}{2} \sigma^2 T) \sqrt{2\pi}}\right)^{\frac{1}{2}} e^{-\frac{1}{4}} e^{-\frac{3\gamma^{\frac{2}{3}} \pi^{\frac{1}{3}}}{2e x_0^{\frac{2}{3}} T^{\frac{1}{3}} \frac{1}{16} \frac{1}{e} \frac{\sigma^2}{e^{\frac{2}{3}}}}} n^{\frac{1}{3}}, & \log(C_1 n) \geq 3 \log(2) + 3 \end{cases} \quad (3.34)$$

3.2 迭代算法

在由上面定理确定了 N 的迭代初始值之后 N_0 ，我们可以使用该值来确定一个合适大小的 M 的迭代初始值。一种选取方式是在 N 为其初始迭代值时，选取尽量小的 M 的值应该使得蒙特卡罗模拟出的概率 $\mathbb{P}(M_T^N < \epsilon)$ 收敛稳定。也即，固定 N 为其初始值不变，当 M 在其迭代初始值附近扰动时，相应的 $\mathbb{P}(M_T^N < \epsilon)$ 也只是在可接受范围内微小扰动。

在我们的问题背景下，通过动态调优和经验发现，一个较为恰当的 M 初始值可以为 $M = 10^5$ ，我们通过比较 $M = 10^5 + i * 1000$ ， $i = 0, \dots, 10$ 。所对应的 $\mathbb{P}(M_T^N < \epsilon)$ 来调整 M ，若对应的概率在 10^{-2} 范围内，则将 M 减小为原先的一半，否则增大 M 为原先的两倍。重复此过程，知道发现满足条件的最小的 M 。我们然后固定这个最优的 M ，然后尝试找出最优的 N 。

根据前面的理论，最优的 N 存在于 $[0, N_0]$ 区间内。我们仍然使用二分法寻找最优的 N ，首先我们令 $N_1 = N_0/2$ ，在此参数下模拟相应的 $\mathbb{P}(M_T^N < \epsilon)$ ，若此模拟概率大于 $Prob$ ，则令 $N_2 = N_1/2$ ，反之，则令 $N_2 = \frac{N_0 + N_1}{2}$ 。重复此过程直至找到最优的 N 。

章节 4

并行化算法

算法并行化的第一步通常是设计一个符合待处理问题及处理器架构特点的并行方案，然后根据该方案确定相应的编程模型，最后分析并行程序的性能，有针对性地做具体的优化。并行程序的性能优化和普通串行程序有许多共通之处。具体来说主要分为两个大类：针对内存使用的优化和针对for循环的改进。

由于日益增长的处理器速度和发展相对较缓的内存速度之间逐渐拉开了差距，内存通常成为程序蓄能的瓶颈。现代处理器通常采用分层内存体系，高带宽，低延迟（lower latency），低能耗的高速缓存对提升数据局部性(data locality)至关重要。针对缓存优化的方法有很多，例如对齐数据（data alignment）使得缓存的加载更有效率，依据缓存的大小处理相应的数据块（cache blocking），预读取（prefetching）提前将需要处理的数据加载到缓存中，或者通过流式存储技术（streaming store）将非时间相关（nontemporal）的计算结果"绕过"缓存直接写入主内存从而避免"污染"缓存其他数据等。和数据代码缓存一样，页表缓存（Translation Lookaside Buffer）也是影响性能的一个因素。页表缓存存储了一部分标签页表条目，用于改进从虚拟内存地址到物理地址的转译速度，对于特定问题，通过使用不同尺寸的页面大小，重构数据可以避免页表缓存中的热点。

针对for循环优化的一般技术包括循环展开（loop unrolling），分块循环（loop tiling），循环互换（loop interchange），循环合并（loop fusion），循环偏移（loop skewing），循环剥离（loop peeling）等。这些改进很多时候也是为了更好地使用内存或者为进一步并行化做准备。目前绝对意义上的串行程序在实际中已不多见，因为在编译过程中编译器会或多或少引入不同程度的并行化处理。然而由于编译器优化的前提是基于代码的正确性，串行程序的逻辑遵循一个绝对的代码执行顺序，因而其编写和调试难度远低于并行程序。并行程序的运行由于同时驱动不同的逻辑运算单元，容易出现数据的竞态条件（race condition）等导致最终结果不确定的因素。本章将着重描述几种针对本课题不同的并行方案，及其具体实现。

4.1 并行性分析

并行的目的就是在充分利用单机资源的基础上，将计算扩展到多个机器，在可接受的时间范围内处理规模更大，复杂度更高的问题。在单个机器上，原则上我们需要发掘串行算法中的并行机会，将基本的并发性操作表述为任务（task）的形式提交给调度器（scheduler），再由调度器将其分发给不同的线程执行。在MIC这样的众核处理器上，足够的任务级并行度对于充分利用片上丰富的线程资源非常重要。由于时间的限制我们选择不重新实现一个新的调度算法，而是使用现有的工具，如OpenMP，Cilk+或者Threading Building Blocks(TBB)。在多个节点上，我们采用消息传递模式如MPI来进行通信，目标是尽量降低通信量的同时，将计算和通信重叠起来，提高整体效率。

算法1描述了欧式期权对冲策略误差控制的串行算法。本章的核心目的就是论述基于该算法的并行化方案。分析该算法可知蒙特卡洛模拟的结果依赖每次模拟求得的误差 $error$ ，而求 $error$ 的过程除了两项非随机量，其余均与 PX 相关。数组 PX 表示的是在离散时间点的期权价格，取值和一个随机布朗运动的各个状态相关联。该布朗运动可视作一个马尔科夫过程，前后两项相差一个符合高斯分布的随机数。由于其每两项之间存在数据依赖性，使得其并行度十分有限。因此问题的关键在于如何有效地打破这种数据之间的依赖性。本文所讨论的数据依赖性本质上源自于 N 个独立高斯分布的随机数组成的数组 $NRV[N]$ 。对于一个给定的 N 和从 $BM[N]$ 中选取的任何一段连

续的子集 $bm[n]$ ，如果可以并行地处理这些分段子集，我们就可以极大地提高计算效率，实现本节最初提出的目标。

计算某个子集的条件是已知其初始状态和该子集所对应 $NRV[N]$ 中的子集。而该子集的初始状态可表示为其之前所有状态对应的正态分布随机数之和乘以一个常量。因此有两种并行思路。第一种是将整个长度为 N 的数组分为连续的子集，按顺序每次所有的计算资源并发地参与计算其中一段子集，计算完成该段子集之后再计算紧邻的下一段。如此，计算下段子集的时候，它的初态就是上一段的终态加上一个随机数，而上段的终态是已知的。依据这种方案，基本的并行任务（task）是已知某段子集对应的所有的随机数，计算其中某个布朗运动状态，其对应的离散时间点的期权价格，以及最终的误差。

第二种思路是将我们之前所讨论的某段子集的处理直接作为基本的并行任务分配给线程计算。这样的话子集的初始状态是未知的，因为不能假设某个线程先后处理的是连续子集。在这种情况下，需要根据之前生成的随机数来推断某段子集的初始状态。这就无法回避一个重要的问题，随机数发生器。

在目前的计算机应用里，几乎所有用到的随机数发生器都只能产生伪随机数。其原理通常是依据某种算法计算出一连串在实际应用中可认为相互独立的数。当该算法的结果空间足够大的时候，生成的伪随机数几乎没有可能重复，从而保证了算法的有效性。在该课题中，我们使用的是基于梅森旋转算法（Mersenne twister）的高斯分布随机数发生器，它有 $2^{19937} - 1$ 的非常长的周期。该随机数发生器的并行版本，在于生成不同的随机流（stream），不同的流使用不同的“种子（seed）”在不同的结果子空间中计算随机数，相互之间保持独立。

在我们的第一种并行思路中，我们只需要生成一条随机流，每次计算某段子集的时候空闲的线程利用随机数发生器生成下一个紧邻子集对应的随机数。所有随机数由全部线程共享，从而解除了数据的依赖性。在第二种并行思路中，由于某段子集作为一个基本的任务分配给线程，不同线程间异步地对不同段子集进行操作，因此无法共享单条随机数组。在这种情况下，每个线程保有一条私有的随机数流，根据自己计算的进度按需生成相应的随机数。这种情况下对于数据依赖性的处理则在于随机数的“种子”。对于不同的流使用不同的种子固然是可以得到互相独立的随机数组。但如果使用相同的种子则会生成完全一样的随机数组。由于不同的线程虽然处理的子集不同，但都是针对同一个给定 N 的离散价格误差的计算， N 个随机数理应一致。通过不同的随机流，不同的线程独立生成随机数，避免了相互之间的通信，从而解除了数据的依赖性。

根据以上提供的两种思路，我们利用Intel OpenMP实现了单机上的并行版本。由于该算法的最终目的是寻找一个最优的 N 以满足给定的可容忍概率。每一次迭代都需要 M 次蒙特卡洛模拟计算一个新的 N 值。在实际中，只有相对较大的 M 值才能得到令人信服的结果。因此对于把单机版本扩展到多个节点或机器，我们选择在 M 这个维度上并行。基本思路是基于Master-Slave模型由一个主机（boss）收集由工作机（worker）发送过来的结果，分析整理之后决定下一次计算的 N ¹值并发送给工作机。工作机在等待指令的同时自行选取一个更小的 N 值继续计算。接受到主机指令之后假如自行计算的 N 值和主机指令相同，则继续完成。否则舍弃计算结果重新开始一个新的 N 值的计算。为了更好地重叠计算和通信，我们采用非阻塞通信模式，如此，工作机可以充分发挥计算效能，避免空隙或等待。具体的实现细节参见??。

4.2 单机并行算法

根据上节4.1的分析，我们在这里给出两种单机并行思路的具体实现细节。在实际中，我们采用Intel MKL（Math Kernel Library）提供的高斯分布随机数发生器。该发生器基于梅森旋转算法。每次可生成单个或多个随机数。

4.2.1 方案一

该方案将整段 N 分为相邻连续子集，分段对每一段子集进行并行处理。

N 值的选择

¹关于 N 的最优化搜索具体讨论见??

Algorithm 2 基于多线程 (multithreading) 和矢量化 (vectorization) 的单机并行算法(单次蒙特卡洛模拟)

输入参数 $X_0, \sigma, K, T, \epsilon, Prob, N_0, M$ (参见算法1)

输入参数 N_{cache}

▷ 依序并行处理的子集大小

```

1: procedure  $mono_1(N_{cache})$ 
2:    $error \leftarrow 0$ 
3:    $NRV[N_{cache}] \leftarrow vdRngGaussian(N_{cache})$       ▷ 一次生成  $N_{cache}$  个符合高斯分布的随机数
4:    $BM[N_{cache}]$ 
5:    $PX[N_{cache}]$ 
6:    $PX[0] \leftarrow 0$ 
7:    $\delta t \leftarrow T/N$ 
8:    $BM[0] \leftarrow \sqrt{\delta t} \times NRV[0]$ 
9:   for  $k = 1 : (int)N/N_{cache}$  do
10:    Start parallel region
11:    for  $i = 1 : N_{cache}$  do
12:       $BM[i] = BM[0] + reduce\_add(NRV[1 : i])$       ▷ SIMD reduction操作
13:       $PX[i + 1] = X_0 \times \exp(-0.5\sigma^2 \times (kN_{cache} + i + 1) \times \delta t + \sigma \times BM[i])$ 
14:    end for
15:    End parallel region  $PX[1] = X_0 \times \exp(-0.5\sigma^2 \times (kN_{cache} + 1) \times \delta t + \sigma \times BM[i])$ 
16:    Start parallel region reduction(+:error) nowait
17:    for  $i = 0 : N_{cache}$  do
18:       $j \leftarrow kN_{cache} + 1$ 
19:       $T_j \leftarrow j \times T/N$ 
20:       $Upper = (\log(PX[i]/K) + 0.5 \times \sigma^2 \times (T - T_j)) / (\sigma \times \sqrt{T - T_j})$ 
21:       $error \leftarrow error - \frac{1}{\sqrt{2\pi}} \times (PX[i + 1] - PX[i]) \times \int_{-\infty}^{Upper} e^{-\frac{t^2}{2}} dt$ 
22:    end for
23:    End parallel region
24:     $NRV[N_{cache}] \leftarrow vdRngGaussian(N_{cache})$ 
25:     $BM[0] = BM[N_{cache} - 1] + \sqrt{\delta t} \times NRV[0]$ 
26:     $PX[0] \leftarrow PX[N_{cache}]$ 
27:  end for
28: end procedure

```

4.3 N的最优化搜索

在获得一个可接受的 N 值后，我们可以继续寻找满足条件的更小的 N 值，一般初始的 N 值取的较大，所以这种最优化的寻找可以通过一个二分法的搜寻算法来实现。在算法 3 中我们设置的收敛

Algorithm 3 最优 N 值的搜索算法

```

1: 参数  $M0$                                 ▷ 蒙特卡洛模拟的次数， $M0$ 越大模拟结果可信度越高
2: procedure NBSECT( $N0, \lambda$ )                ▷  $N0$ 是初始化抽样次数， $\lambda$ 是最优解的置信区间
3:    $nL \leftarrow 1$                             ▷ 设置 $n$ 值的初始下界
4:    $nU \leftarrow N0$                           ▷ 设置 $n$ 值的初始上界
5:    $n \leftarrow 0.5 \cdot (nL + nU)$ 
6:   while  $|nU - nL| > \lambda$  do
7:     if  $BSEERROR(M0, n) < n$  then           ▷ 缩小搜索范围
8:        $nU \leftarrow n$ 
9:        $n \leftarrow 0.5 \cdot (nL + nU)$ 
10:    else                                   ▷ 扩大搜索范围
11:       $nL \leftarrow n$ 
12:       $n \leftarrow 0.5 \cdot (nL + nU)$ 
13:    end if
14:  end while
15:  return  $n, nU, nL$ 
16: end procedure

```

条件为 $|nU - nL| \leq \lambda$ 这是因为蒙特卡洛模拟本身具有一定的随机性。在 N 的收敛过程中会出现波动，导致无法收敛到一个单一的值上，在 $BSEERROR$ 中给定了置信概率 $Prob$ 的情况下，我们很自然地就能利用置信区间这一概念来表述我们取到的最优值 N 。在实际应用中， N 的初始值通常需要取得很大，然后通过二分法来逐步缩小范围取得最优值。此外蒙特卡洛模拟得次数 M 也需要取一个很大的值来保证其精确性。如果 N 和 M 的取值都较大，就会极大地增加计算时间和内存空间需求。这时串行程序在普通的计算机上就难以满足问题的需求，我们将需要转而开发并行程序并使其运行在超级计算机上。

参考文献

- [1] Dimitris Bertsimas, Leonid Kogan, and Andrew W. Lo. When is time continuous? *Journal of Financial Economics*, Elsevier, 55(2):173–204, February 2000.
- [2] Paul Kree. Eric Carlen. L_p Estimates on Iterated Stochastic Integrals. *The Annals of Probability*, 19(1):1354–368, January 1991.
- [3] Emmanuel Gobet and Emmanuel Temam. Discrete time hedging errors for options with irregular payoffs. *Finance and Stochastics*, 5(3):357–367, February 2001.
- [4] Takaki Hayashi and Per A. Mykland. Evaluating hedging errors: An asymptotic approach. *Math. Finance*, 15:309–343, 2005.
- [5] Jean-Francois Le Gall. Mouvement brownien, martingales et calcul stochastique. *Mathématiques et Applications*, Vol. 71. Springer, 2013.
- [6] D. Revuz and M Yor. Continuous Martingales and Brownian motion. *Grundlehren der mathematischen Wissenschaften* ; 293. Springer, 1999.
- [7] Holger Rootzen. Limit distributions for the error in approximations of stochastic integrals. *The Annals of Probability*, 8(2):241–251, 04 1980.
- [8] Elias M. Stein. *Singular Integrals and Differentiability Properties of Functions*. Princeton University Press. Princeton University Press, 1971.