

更快的 Quant 开发:

通过加速 C++ 构建提升 Quant 开发
的速度与质量

简介

一个高效敏捷的 Quant 团队对金融服务公司来说是如虎添翼。

他们可以追踪每一个细小的不确定性或风险，精准评估进行交易的确切时间，甚至计算事件发生的概率，精确到小数位。

最关键的是，他们要与竞争对手比拼，以风驰电掣之速抢占先机。正是因为对速度要求更高，Quant 和 Quant 工具开发者偏爱快速、高效、低级的语言（）如 C++）。

Quant 由两种对立力量主导：极速前进，却害怕出错——因为这将增加故障停机时间、错失机会，甚至招致风险。

为了平衡这两种力量，Quant 需要寻找加速工具。但这也是许多金融服务公司不愿做最后一搏的原因，他们担心在 Quant 相关的进程中引入新元素会有一定风险，可能会影响性能。

但这种情况不一定发生。

借助正确的理解和可靠的合作伙伴，公司可以使用 C++ 构建加速工具来为 Quant 提供更敏捷的服务，并保持他们的竞争优势，同时不会失去稳定性、引入安全风险，也不需要他们更改工具链或现有代码。

在这篇白皮书中，我们将介绍：

- Quant 及 Quant 工具开发者面临的最大挑战
- C++ 库（以及市场上热门库）的优势
- Quant 构建中使用加速器工具的好处（以及挑选加速器的一些主要考虑因素）
- 两个使用 C++ 加速金融服务构建的案例

Quant 在金融服务软件开发中的作用

如果你正在阅读这篇白皮书，你很可能已经对 Quant 及其工作有了相当清晰地了解，所以我们在这里不会深入探讨其定义。

相反，我们来探究一下为什么 Quant 是金融服务的一个重要组成部分。

为什么金融行业喜欢 Quant?

在现代金融服务业，量化分析是企业明智决策、优化业绩、管理风险等不可或缺的方法。

这就是 Quant 的用武之地。

量化分析法以数据为基础，通过科学的数据分析做出决策（而非盲目猜测），帮助金融服务公司化繁为简，在变化中找到不变性。

当涉及到优化金融软件系统的性能时，更是如此。通过使用数学模型和算法，开发者可以设计最优的系统来支持 Quant，更高效地进行交易、处理海量数据，并提供更快、更准确的结果。

在这样一个快节奏的市场中，这些改进可以带来无与伦比的竞争优势。

Quant 也与风险管理紧密相连。这是一个以光速发展的行业。错误的决定可能会给个人、公司 and 经济带来严重后果。其中涉及影响风险和回报的因素是极其复杂的。

在这些情况下，量化分析法可以帮助识别软件系统中的潜在风险和漏洞，远远快于手动识别。帮助开发者理解并解决这些问题，确保系统的可靠性和安全性。

让我们简要了解一下 Quant 在金融服务软件行业中一些更具体的应用。

交易软件

股票和资产价格瞬息万变，一纳秒就可能是一个股票交易所带来惊人的利润或灾难性的损失。公司在分秒间诞生或毁灭。

其结果是，交易软件公司迫切需要快速、可靠的软件，能够支持这种快速决策。

在这个领域，Quant 借助一定的软件为公司提供快速的建议，优化交易大厅的瞬时决策。这些软件由专门的 Quant 开发者设计。当然，这种实时支持只是 Quant 在交易软件中扮演的角色之一。Quant 对于模型创建（以及可用于模型创建的工具）、波动交易和风险分析也至关重要，我们将在下文深入探讨。

风险管理

“Risk Quant” 负责开发帮助银行管理风险的工具。他们将监测、控制与金融稳定、声誉和合规有关的风险。

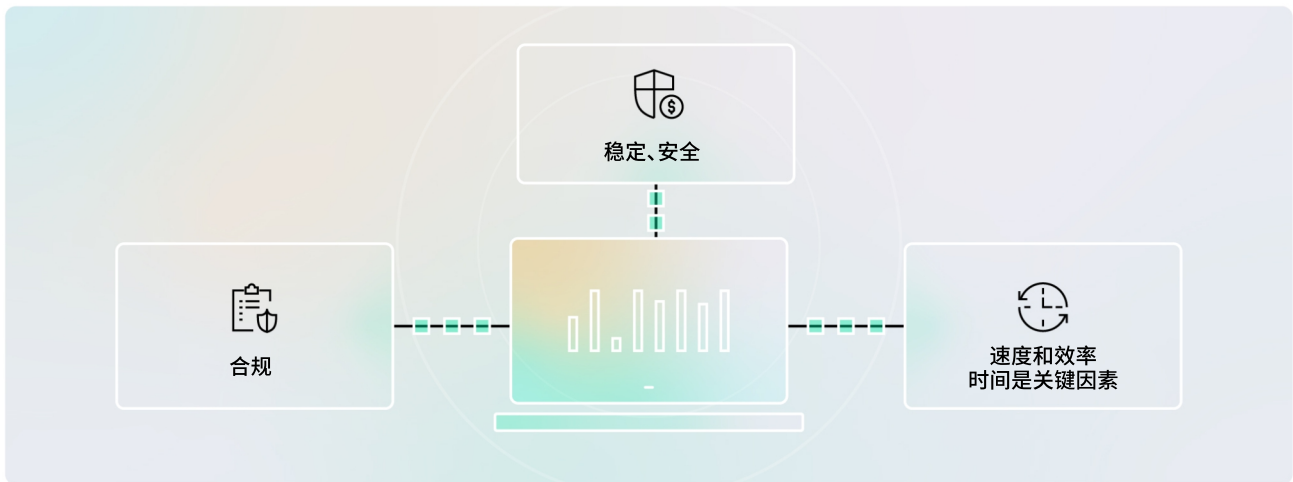
这一领域的 Quant 使用他们的模型来评估从信用风险、市场、流动性、操作或监管风险的所有信息。他们不断地帮助金融服务公司预测未来，这些预估的风险一旦成真，他们就可以及时采取行动，最大限度地减少，甚至完全避免损失。

金融建模

金融建模是一个广阔的领域。这个领域的 Quant 提供了实现金融进程自动化、改善数据分析、增强风险管理所需的模型。他们还将支持部分核心银行任务和一些进程，如投资组合管理或市场和贸易监督。



金融服务公司在 Quant 开发中的挑战



在上一章中我们提到，所有行业中，Quant 的努力成果是一个模型、算法或模拟。这个模型、算法或模拟可以用来精确地预测未来，分析风险，优化投资组合，或者承担上百个其他复杂的应用。

但是，无论应用如何，Quant 的价值总是通过相同的指标来衡量：时效性与准确性。

Quant 的能力是在公司或客户最需要的时候，即刻为他们提供服务，而不是拖到一段时间之后。

要在速度和稳定性之间保持平衡并不容易。但是，依照本白皮书的目的，我们将重点关注其中最大的三个挑战。

高效工作

要建立一个足够复杂的金融模型，在千变万化的市场中精确定位风险，或准确发掘适合交易的时刻，这难度已经很高了。但要高效地做到这一点更是难上加难。

Quant 模型是行业中最复杂的模拟之一。建立它们需要大量复杂的数据和密集的数学计算。而且，一旦建成，在投入使用之前还需要经过严格的验证和审核。

由于要遵循如此复杂的进程，进程、硬件或软件中任何细微低效的步骤都可能会导致数小时或数天的时间损失。

稳定性和安全性

Quant 要不断处理高度敏感的数据。这些数据一旦泄露，可能会导致客户身份盗窃，公司竞争对手获得内幕信息等后果。

保护这些敏感信息需要 Quant 完成一系列复杂的任务，包括加密、严格的数据访问控制等方面。其难点在于首先进程要尽可能保证安全，其次，持续不断地安全检查不能减慢软件或任何 Quant 工作人员的速度。

合规

很少有行业比金融行业的监管更严格。由于 Quant 要处理海量的敏感数据，在遵守这些监管规定方面，他们需要走在游戏的前面，如果他们不想被罚款，或保住饭碗的话。

熟悉所有这些监管要求且遵守规定并不容易，尤其是对于客户遍及多个国家和立法区域的 Quant 或构建工具专家来说。

但再一次强调，Quant 总是面临着合规、速度与准确性的压力。他们在寻找保护敏感数据的方法，并确保这些数据仍然足够可用和可访问，以便从中找到灵感。



为什么 C++ 是 Quant 最爱的语言？

C++ 几乎统治了定量分析领域。纵观任何一家贸易公司或金融服务程序开发者的开发文件，你会看到一行接一行的 .cpp 文件扩展名。

但 C++ 为何如此受欢迎？为什么开发者要优先加速 C++，而不是切换到另一种语言？

对于任何在这一行业工作过的人来说，C++ 流行的第一个原因很直白：

在一个高度抗拒变化的领域，C++ 是熟悉的语言。

但 C++ 确实有几个特征非常适合 Quant：

- **C++ 是高效的**—— C++ 直接编译成机器代码，所以编译时间比很多其他语言少得多，它的执行时间也很短。这使得它非常适合时间紧张的 Quant 应用。
- **C++ 是可移植的**—— C++ 不受限于平台，所以很容易将用 C++ 编写的程序移植到其他平台。

- **C++ 让开发者可以控制内存管理**—— C++ 不使用垃圾回收器，这意味着开发者可以完全控制内存管理。对于一些开发者来说，这意味着可以完全控制敏感的金融信息存储在哪里，以及如何存储。
- **C++ 背后有一个庞大的社区**—— C++ 用户花了数年时间开发了一个巨大的指南、课程、提示和技巧库。如果 C++ 出了什么问题，Quant 几乎总能找到所需的建议和支持。这个庞大的粉丝社群也意味着有大量的 C++ 库可以帮助加速开发过程，我们将在本白皮书的后面介绍这些库。
- **C++ 与遗留系统集成良好**——正如引言中提到的，C++ 多年来一直是金融服务公司的最爱语言，这意味着他们的大多数系统可能已经基于 C++ 构建，因此 C++ 应用程序集成易如反掌。

这些都是优势，但对于 Quant 团队来说，过度依赖 C++ 会带来一些难题。即开发和测试的构建、编译时间将明显拉长。缓慢的构建可能导致瓶颈，必要的软件开发、修复推迟，造成工作受阻。幸运的是，这并不是一个棘手的问题。

为什么要加速 C++ Quant 软件构建？

每个参与 Quant 的人都在不断地追求更快地开发。在进程的每个阶段都要争分夺秒，减少低效工作，并尽可能有效地利用所有计算能力。

Quant 构建是复杂的，随着数据量的增长以及金融机构在竞争中不断进步，其复杂性只会不断恶化。

这就是为什么 C++ 开发加速工具正在成为 Quant 的关键工具。

好的加速器会减少低效构建，避免拖延这些本就复杂耗时的构建进程。

稍后我们会介绍 C++ 构建加速的优点。但首先，让我们建立一些成功加速器的基本准则。

快速推进，且不出错

加速可以带来巨大的好处。但是，在这样一个多变且精确的行业，必须谨慎对待。

为了真正获得效益，你需要考虑三个关键因素，这是成败与否的关键。

平衡速度与稳定

在这个行业中，残缺的构建或粗制滥造的编译可能是灾难性的。它可能导致计算错误，甚至毁灭性的财务决策。其延迟，可能让一笔高利润的交易变成巨大损失。

糟糕的代码本身就是个问题。

糟糕的代码与缓慢的构建，将导致巨大的瓶颈，甚至危及延误交付期限。

然而，各种加速工具并不一样。如果你的工具不是为处理金融服务开发的复杂性而设计的，那么你可能会面临依赖、资源管理和集成受阻等各种问题。所有这些都是项目推进路上的绊脚石。

因此，寻找一款能够平衡速度与稳定的加速工具至关重要。这种经过实战检验的解决方案，可以给你力量和信心，让你知道你可以按时交付一个强大的构建。

为什么 C++ 是 Quant 最爱的语言？

要寻找一个解决方案，即：

- 构建能与现有技术栈平滑集成。这意味着你不必担心兼容性问题，团队可以快速掌握新工具。
- 设计考虑了合规与安全，以金融服务的专业知识为后盾。确保它从第一天起就适合所有的安全和合规框架，这样你就不必担心需要更改代码库或最终构建工件。
- 背后有企业级支持。这些都是至关重要的过程，所以无论何时何地需要，你都需要专业的支持。

首当其冲，赢得竞争

“没坏就不修”的心态仍然主导着金融服务行业。

企业、经济和生计对大型机构来说至关重要。如果出了问题，即便是小型金融服务公司也可能面临巨大损失。

因此，难怪一些公司不愿打乱已经持续多年（或几十年）的旧模式。

因此，新的开发加速工具很难得到支持。从一线开发者到高级管理人员，每个人都可能会牺牲效率收益，而选择“更安全”的选项。

问题是，这种“安全”的选择根本不安全。这个行业的发展之快史无前例，对 Quant 的需求蓬勃增长，而 Quant 自身也面临着越来越大的压力，他们需要在更短的时间内提供更精确的意见。

那些仍然受限于耗时构建的公司迟早会发现自己落后了。

竞争对手有足够的速度和敏捷性，新机会一出现就能迅速抢占先机，并更快地迭代，而你只会不断失去盈利的机会。

为了让开发加速取得成功，关键的受益人需要明白，如今，什么都不做与冒险尝试并拥抱新的加速工具风险程度不相上下。

你不需要说服紧张的受益人成为第一个吃螃蟹的人，把业务搞得天翻地转。最好的工具可以帮助你轻松加速 C++ 开发，他们的设计宗旨是快速、无痛且安全。

为了赢得受益人的支持，你需要寻找专注ROI的工具。这类工具通常易于集成，不需要你更改现有的代码或流程，就可以帮助你使用其技术，同时根据你的需求做出调整。他们还会花时间来确保其解决方案能真正有效并实现价值。

持续优先考虑合规

如果企业在这个过程中拿自己的合规，以及自己的声誉去冒险，那么加快步伐是没意义的。

为了取得成功，公司需要寻找能够为加速提供绝对安全环境的工具。寻找能够提供安全沙盒的工具，这种工具可以将数据紧密锁住，同时仍然能够模拟你的流程，有效运行所需的一切。

最好的工具可以不接触你的代码工件，保证合规和安全。它们不会收集或存储数据，也永远无法看到你的任何代码，这意味着它没有泄漏的机会。

最重要的是，你的加速器永远不会从代码本身中添加或删除任何信息，这样他们就无法更改任何东西，影响到你的合规。

Quant 最流行的 C++ 库

快速写出好的 Quant 代码并不是一件容易的事情。因此，Quant 开发者需要尽全力提升效率。他们不可能从头开始构建所有东西，浪费宝贵的时间去测试每一行代码，特别是当他们可以利用之前开发者的专业知识和经验时。开发者的工作并不容易。

C++ 的一个关键优势是它附带了大量的库。其中一些甚至是专门为解决常见的 Quant 进程和难点而开发的。

因此，量化开发者有一整套触手可及的快捷方式。

这些强大的、经过广泛测试的库可以轻松融入定量分析的过程，并保证定量分析迫切需要的那种速度和效率。

想深入研究量化库？以下是市场上最受欢迎的一些库：

gsl

gsl 代表 GNU 科学库。它是一个为 C 和 C++ 程序员设计的免费数学库，支持大量的数学例程，包括随机数生成器、特殊函数和最小二乘拟合。

Boost

对于 Quant 来说，Boost 通常最适用于线性代数、伪随机数生成、多线程和图像处理等任务。

Armadillo

Armadillo 是一个面向 C++ 的“高质量线性代数库”。对于 Quant 来说，它为向量、矩阵、立方和复数运算提供了一系列高效的类。

QuantLib

一个专门为量化金融设计的免费开源库。Quant 可以在 QuantLib 中找到从数学卷到蒙特卡洛框架、现金流、定价引擎、股权和市场模型以及波动性模型的所有内容。所有这些都在定量分析领域得到了广泛的测试。

Eigen

Eigen 主要用于线性代数——矩阵、向量和数值求解器。它一直在积极更新中，新版本会定期上市。开发过程包括大量的测试，以消除 bug，这对注重稳定性的 Quant 来说是一个重要考虑因素。

GLPK

一组用 ANSI C 编写的例程，主要用于大规模线性规划和混合整数规划等应用程序。GLPK 的性能是可变的，使用起来相对快速且简单。

C++ 库和编译中的加速

现在我们已经了解了 C++ 的优势，并建立了成功加速器的一些基本准则，让我们来看看具体细节。C++ 如何加速构建，帮助 Quant 和 Quant 开发者解决上面提到的效率、稳定和合规方面的难题？以下是 C++ 构建加速工具升级 Quant 进程的五种正确方式。

强化模型开发

Quant 被称为金融服务业的“火箭科学家”是有原因的：他们使用极其复杂的数学和统计学来创建一些业内最先进的模型。

而如果没有大量的微调和迭代，这类模型是无法形成的。问题是，当编译时间变慢时，模型的每一次微调和每一次新迭代都可能意味着数小时的时间损耗。

这就是为什么许多 Quant 团队和 Quant 开发者使用加速技术来加速编译。因为他们能够更快地迭代，并有助于平衡速度和准确性之间的老矛盾。

效率爆增

最好的加速技术将并行开发提升到了新的水平。例如，在输入不变的情况下，Incredibuild 使用构建缓存技术来重用之前构建的缓存输出。

它将开发进程分解为微小的进程。然后它或使用缓存技术重用现有的构建输出，或者将你的进程分布在本地或云端网络中的数百个空闲机器内核上，让它们都可以同时运行。

这意味着开发者可以更快地迭代，构建等待的时间减少，构建运行时从一个任务跳转到另一个任务所带来的上下文切换也更少。同时，加速工具可以利用增量构建，替代全量重建，使用更多分支帮助迭代开发。

对于管理者来说，更快的构建和迭代意味着更快的解决问题。反过来，这也能帮助他们实现最重要的目标：按时完成任务，交付更高质量的产品。

用更少的资源做更多工作

加速工具对于扩展 Quant 开发基础设施至关重要。它们将编译任务分布在数百个空闲 cpu 或未使用的云端资源中,处理更大的项目以及更耗时的构建,而不牺牲速度或质量。

随着时间的推移,效率提升可以大大节省成本,减少开支损耗,降低更新昂贵硬件的需求。

开发更快,干扰更少

最好的加速技术应该尽可能少地干扰现有的进程。找到一个可与现有的工作进程集成的工具,是改善进程的完美方案,同时也可以向持怀疑态度的受益者表明,进程和模型是安全的,不会被破坏。

你需要一种无需更改现有代码、工具链或开发进程就能使用的工具。它可以无缝插入到你的技术栈中,同时你的团队能立即轻松上手并开始使用。



C++ 开发加速在行动： 两个成功案例

让我们来看两个例子，c++ 开发加速对 Quant 开发者的生产力和效率产生了重大影响。

国际自营交易公司借助 Incredibuild 将构建 时间缩短了 50%

这家大型自营交易公司的 Quant 工程师使用 Linux 上的最新 C++ 功能去设计、实现和部署交易策略。

他们的开发周期短且频繁。他们的重点始终是保持市场领先，并使用 Quant 来优化交易的各个方面。团队中的工程师与交易和 Quant 研究人员组成的精英团队携手合作，开发行业最高效、及时的交易策略。这个团队处理的每个项目都是关键任务，因此对速度和质量的需求是极高的。

Incredibuild 的解决方案

Incredibuild 是市场上领先的开发加速平台，该公司的 Quant 和云工程团队向我们寻求支持。

我们与他们合作，以加速他们的 C++ 功能和在 Azure Kubernetes Service (AKS) 上的部署。在 Incredibuild 的帮助下，公司的构建时间缩短了 50%，保证了快速迭代、灵活性和稳定性，并夯实了他们在市场上的地位。

美国主流银行如何利用 Incredibuild 将回归 测试提速 95%

一家主流的美国金融机构联系了 Incredibuild，询问该公司的开发团队在完成几个模型的回归测试时遇到的问题。该公司需要加速回归和单元测试。此外，银行运行几种蒙特卡洛算法，这些算法也需要更快地运行。

Incredibuild 的解决方案

使用 Incredibuild，这家美国银行能够为 150 多名开发者加速他们的 C++ 库和 C# Quant 分析构建。此外，团队开始使用 Incredibuild 来加速他们的测试，并立即看到了成效。在团队完全熟练操作的情况下，回归测试和单元测试时间从 19 个小时下降到仅仅 1 个小时。我们能够通过使用 Python 和 Incredibuild API 来实现这一点。

不进则退

Quant 比任何人都清楚这个领域对速度的需求。他们知道，一纳秒的加载时间可以是惊人的利润或毁灭性的损失。

很多时候, 金融服务企业的结论是, 不采取行动是最安全的选择。但事实并非如此。

在 2022 年的一项调查中, C++ 开发者报告说, 长构建是他们使用这门语言时的第二大痛点。

这就是 Quant 分析行业迫切需要有效的 C++ 开发加速工具的原因。这些工具能够加速构建并缩短构建时间的软件, 消除减缓 Quant 进程的主要障碍, 避免被竞争对手超车。

强化模型开发

但是, 随着市场的发展, 看似“安全”的选择正变得越来越不安全。

随着越来越多的 Quant 接受加速工具, 整个行业即将迎来一次飞跃。而且, 在一个分秒必争的世界里, 持续交付、敏捷性和灵活性是所有公司梦寐以求的状态。无作为的公司将很快就被甩在后面, 苦苦挣扎着突破瓶颈, 努力按时交付准确的计算和高质量的代码。

如需了解更多信息:

www.incredibuild.cn

电话: +86 -28 -63207464 | 邮箱: marketing_cn@incredibuild.cn

© 2023 快编大师(成都)软件有限公司版权所有。