性能效率支柱

AWS 架构完善的框架

2020 年 7 月

声明

客户负责对本文档中的信息进行独立评估。本文档：(a) 仅供参考，(b) 代表 AWS 当前的产品服务和实践，如有变更，恕不另行通知，以及 (c) 不构成 AWS 及其附属公司、供应商或授权商的任何承诺或保证。AWS 产品或服务均“按原样”提供，没有任何明示或暗示的担保、声明或条件。AWS 对其客户的责任和义务由 AWS 协议决定，本文档与 AWS 和客户之间签订的任何协议无关，亦不影响任何此类协议。

目录

摘要

本白皮书重点介绍 Amazon Web Services (AWS) 架构完善的框架的性能效率支柱。文中提供了指导，可帮助客户在 AWS 环境的设计、交付和维护过程中应用最佳实践。

性能效率支柱为生产环境管理提供了最佳实践。本白皮书中不涉及非生产环境和过程（例如持续集成或持续交付）的设计和管理。

简介

AWS 架构完善的框架能够帮助您认识到您在 AWS 上构建工作负载时所做决策的优缺点。使用此框架有助于您了解在云中设计和运行可靠、安全、高效且经济实惠工作负载的架构最佳实践。该框架提供了一种方法，使您能够根据最佳实践持续衡量架构，并确定需要改进的方面。我们相信，拥有架构完善的工作负载能够大大提高实现业务成功的可能性。

该框架基于五大支柱：

- 卓越运营
- 安全性
- 可靠性
- 性能效率
- 成本优化

本白皮书重点介绍如何将性能效率支柱的原则应用于您的工作负载。在传统的本地环境中，实现持久的高性能比较困难。遵循本白皮书中的原则将帮助您在 AWS 上构建能够长期高效提供持续性能的架构。

本白皮书的目标读者是技术岗位的人员，如首席技术官 (CTO)、架构师、开发人员和运维团队成员。阅读本白皮书后，您将了解在设计高性能云架构时可以使用的 AWS 最佳实践和策略。本白皮书不提供实施细节或架构模式，但会提供对适当资源的引用。

性能效率

性能效率支柱专注于有效利用计算资源来满足需求的能力，以及如何在需求发生变化和技术不断演进的情况下保持高效率。

设计原则

以下这些设计原则可帮助您在云中实现并维护高效工作负载。

- **普及先进技术**：通过将复杂的任务委派给云供应商，降低您的团队实施高级技术的难度。与要求您的 IT 团队学习有关托管和运行新技术的知识相比，考虑将新技术作为服务使用是一种更好的选择。例如，NoSQL 数据库、媒体转码和机器学习都是需要专业知识才能使用的技术。在云中，这些技术会转变为团队可以使用的服务，让团队能够专注于产品开发，而不是资源预置和管理。
- **数分钟内实现全球化部署**：您可以在全球多个 AWS 区域中部署工作负载，从而以更低的成本为客户提供更低的延迟和更好的体验。
- **使用无服务器架构**：借助无服务器架构，您无需运行和维护物理服务器即可执行传统计算活动。例如，无服务器存储服务可以充当静态网站，从而无需再使用 Web 服务器，事件服务则可以实现代码托管。这不仅能够消除管理物理服务器产生的运行负担，还可以借由以云规模运行的托管服务来降低业务成本。
- **提升实验频率**：利用虚拟资源和可自动化的资源，您可以使用不同类型的实例、存储或配置来快速进行比较测试。
- **考虑软硬件协同编程**：使用最符合您目标的技术方法。例如，在选择数据库或存储方法时考虑数据访问模式。

定义

通过重点关注以下这些领域，在云中实现高性能效率：

- 选择
- 审核
- 监控
- 权衡

采用数据驱动型方法来构建高性能架构。收集架构各方面的数据，从总体设计到资源类型的选择与配置都包括在内。

定期检查您的选择，确保充分利用不断发展的 AWS 云的优势。监控可以确保您随时发现与预期性能的偏差。您可以对您的架构作出权衡以便提高性能，例如使用压缩或缓存，或放宽一致性要求。

选择

针对特定工作负载的最佳解决方案各不相同，而且解决方案通常会结合多种方法。架构完善的工作负载会使用多种解决方案，并且启用各种不同的功能来提高性能。

我们提供多种类型和配置的 AWS 资源，可让您更轻松地找到最能满足您的需求的方法。此外，我们还提供了无法使用本地基础设施轻松实现的选项。例如，Amazon DynamoDB 之类的托管服务可以提供完全托管的 NoSQL 数据库，确保在任何规模下都只会有几毫秒的延迟。

性能架构选择

一个工作负载通常需要采用多种方法来实现最佳性能。架构完善的系统会使用多种解决方案，并且利用各种不同的功能来提高性能。

使用数据驱动型方法来为您的架构选择模式和实施方式，获得经济高效的解决方案。AWS 解决方案架构师、AWS 参考架构和 AWS 合作伙伴网络 (APN) 合作伙伴可以根据自身的专业知识帮助您选择合适的架构，不过将需要通过基准测试或负载测试提取的数据来优化您的架构。

您的架构可能会结合多种不同的架构方法（例如事件驱动、ETL 或管道）。架构的实现将使用各种专门用于优化架构性能的 AWS 服务。在接下来的章节中，我们会介绍您应该考虑的四种主要资源类型：计算、存储、数据库和网络。

了解可用的服务和资源：学习并了解在云中可用的各种服务和资源。认识与您的工作负载相关的服务和配置选项，并了解如何实现更高的性能。

如果要评估现有工作负载，您必须生成评估所需使用的各种服务资源的清单。这份清单可帮助您评估可以用托管服务和较新技术替换的组件。

制定架构选择流程：借助关于云的内部经验和知识，或借助外部资源（例如已发布的使用案例、相关文档或白皮书），制定资源和服务选择流程。您应该制定一个流程，以鼓励对可能会用于工作负载的不同服务进行试验和基准测试。

在编写有关架构的重要用户案例时，应包括性能要求，例如指定每个重要案例应采用的执行速度。对于这些重要案例，您应该实施额外的脚本化用户体验，以确保您可以深入了解这些案例如何根据您的要求执行。

将成本要求纳入决策考量：工作负载通常具有运营成本要求。根据预测的资源需求，使用内部成本控制机制来选择资源类型和规模。

确定可以将哪些工作负载组件替换为完全托管的服务，例如托管数据库、内存中的缓存和其他服务。减少运营工作负载让您可以将资源集中到取得业务成果上。

有关成本要求最佳实践，请参阅成本优化支柱白皮书的"*经济高效的资源*"部分。

使用策略或参考架构：通过评估内部策略和现有参考架构，以及使用分析为工作负载选择服务和配置，来最大程度提高性能和效率。

使用来自云提供商或适当合作伙伴的指导：使用云公司资源（例如解决方案架构师、专业服务或适当合作伙伴）来指导您做出决策。这些资源有助于检查和改进您的架构，从而获得最佳性能。

如需其他指导或产品信息，请联系 AWS 获得帮助。AWS 解决方案架构师和 AWS 专业服务提供关于解决方案实施的指南。APN 合作伙伴提供 AWS 专业知识，可帮助您为业务开发敏捷性和创新能力

对现有工作负载的性能进行基准测试：对现有工作负载的性能进行基准测试，了解工作负载在云上的表现。使用从基准测试中收集的数据来推动架构决策。

结合使用基准测试与综合测试，生成有关工作负载组件性能的数据。相比负载测试，基准测试通常可以更快速地设置，适用于评估特定组件的技术。基准测试通常在新项目开始时进行，因为此时您还没有用于进行负载测试的完整解决方案。

您可以构建您自己的自定义基准测试，也可以使用行业标准测试（如 TPC-DS），来对您的数据仓库工作负载进行基准测试。行业基准适用于比较不同的环境。自定义基准适用于找出您希望在架构中执行的特定操作类型。

进行基准测试时，为了确保获得有效结果，预热您的测试环境尤为重要。多次运行同一基准测试，确保捕获在一段时间内的差异信息。

由于基准测试运行速度通常比负载测试快，它们可以在部署管道的早期使用，并能更快地提供有关性能偏差的反馈。当您评估一个组件或服务的重要更改时，您可以使用基准快速了解您是否有合理的理由来执行更改。结合使用基准测试与负载测试这一点很重要，因为负载测试会告诉您工作负载在生产环境中的表现如何。

对工作负载进行负载测试：使用不同的资源类型和大小在云上部署最新的工作负载架构。监控部署情况，捕获用于识别性能瓶颈或容量过剩的性能指标。使用此性能信息来设计或改进您的架构和资源选择。

负载测试要使用您的*实际*工作负载，因此您可以了解解决方案在生产环境中的表现。负载测试必须使用生产数据的合成或净化版本（删除敏感信息或身份识别信息）进行。大规模使用重演或预设的工作负载用户旅程，演练整个架构。作为交付管道的一部分，自动执行负载测试，并将结果与预定义的 KPI 和阈值进行比较。这可以确保您持续获得所需的性能。

Amazon CloudWatch 可以收集架构中各种资源的指标。您也可以收集和发布自定义指标，用于显示业务指标或派生指标。使用 CloudWatch 设置超出阈值警报，发生警报则表明测试没有实现预期性能。

利用 AWS 服务，您可以运行生产规模的环境来主动测试您的架构。您只需为需要的测试环境付费，因此执行全面测试的成本远远低于使用本地环境的成本。利用 AWS 云测试您的工作负载，了解工作负载无法扩展或者以非线性方式扩展的方面。您可以使用 Amazon EC2 Spot 实例以很低的成本生成负载，并在投入生产前发现瓶颈。

如果执行负载测试需要花费大量时间，请使用测试环境的多个副本并行执行负载测试。这样一来，您的成本不会有大的变动，但测试时间将会缩短。（将一个 EC2 实例运行 100 个小时的成本与将 100 个实例运行 1 小时的成本相同。） 您还可以通过使用 Spot 实例并选择成本低于生产所用区域的区域，来降低负载测试的成本。

负载测试客户端的位置应反映最终用户的地理分布。

资源

请参阅以下资源，详细了解有关负载测试的 AWS 最佳实践。

视频

- Introducing The Amazon Builders' Library (DOP328)

文档

- AWS 架构中心
- Amazon S3 性能优化

- Amazon EBS 卷性能
- AWS CodeDeploy
- AWS CloudFormation
- 负载测试 CloudFront
- AWS CloudWatch 控制面板

计算架构选择

适合特定工作负载的最佳计算方案会因应用程序设计、使用模式和配置设置而有所不同。架构可能会使用不同的计算方案来支持各种组件，并启用不同的功能来提高性能。为架构选择错误的计算方案可能会降低性能效率。

评估可用的计算方案：了解您可以使用的、与计算相关的方案的性能特性。了解实例、容器和函数的工作原理，以及它们对您的工作负载的有利影响和不利影响。

在 AWS 中，计算资源有三种形式：实例、容器和函数。

实例

实例是虚拟化服务器，因此您只需通过一个按钮或一次 API 调用即可对其功能进行调整。因为云中的资源决策不是固定不变的，所以您可以尝试使用不同的服务器类型。在 AWS 中，这些虚拟服务器实例具有不同的系列和尺寸，并且可以提供各种功能，包括固态硬盘 (SSD) 和图形处理单元 (GPU)。

Amazon Elastic Compute Cloud (Amazon EC2) 虚拟服务器实例具有不同的系列和大小。它们提供各种功能，包括固态硬盘 (SSD) 和图形处理单元 (GPU)。启动 EC2 实例时，您指定的实例类型决定了用于实例的主机的硬件。每种实例类型都提供不同的计算、内存和存储功能。我们按照这些功能把实例类型分组为实例系列。

基于数据选择最适合您工作负载的 EC2 实例类型，确保设置了正确的联网和存储选项，并且采用能够提高工作负载性能的操作系统设置。

容器

容器是实现操作系统虚拟化的一种方法，让您能够在资源隔离的进程中运行应用程序及其依赖项。

在 AWS 上运行容器时，您需要做出两个选择。首先，选择是否要管理服务器。AWS Fargate 是用于容器的无服务器计算引擎。如果您需要控制计算环境的安装、配置和管理，则可以使用 Amazon EC2。其次，选择是要使用 Amazon Elastic Container Service (ECS) 还是 Amazon Elastic Kubernetes Service (EKS) 作为容器编排工具

Amazon Elastic Container Service (Amazon ECS) 是一项完全托管的容器编排服务。通过此服务，您可以使用 AWS Fargate 在 EC2 实例或无服务器实例集群上自动执行和管理容器。您可以将 Amazon ECS 与其他服务本地集成，例如 Amazon Route 53、Secrets Manager、AWS Identity and Access Management (IAM) 和 Amazon CloudWatch。

Amazon Elastic Kubernetes Service (Amazon EKS) 是一项完全托管的 Kubernetes 服务。您可以选择使用 AWS Fargate 运行 EKS 集群，而无需预置和管理服务器。EKS 与 Amazon CloudWatch、Auto Scaling 组、AWS Identity and Access Management (IAM) 和 Amazon Virtual Private Cloud (VPC) 等服务深度集成。

使用容器时，您必须基于数据选择最适合工作负载的容器类型，就像基于数据选择 EC2 或 AWS Fargate 实例类型一样。还需要考虑容器配置选项，例如内存、CPU 和租户配置。如需在容器服务之间启用网络访问，不妨考虑使用服务网格（例如 AWS App Mesh，它可以对服务的通信方式进行标准化处理）。服务网格可为您提供端到端可见性，并确保您的应用程序具有高可用性。

函数

函数从您要执行的代码中抽象出执行环境。例如，您可以使用 AWS Lambda 在不运行实例的情况下执行代码。

借助 AWS Lambda，您可以为任何类型的应用程序或后端服务运行代码，而且无需任何管理。您只需上传代码，AWS Lambda 就会处理运行和扩展代码所需的一切工作。您可以将您的代码设置为自动从其他 AWS 服务触发，可以直接调用，也可以与 Amazon API Gateway 一起使用。

Amazon API Gateway 是一种完全托管的服务，可以帮助开发人员轻松创建、发布、维护、监控和保护任意规模的 API。您可以创建一个 API 充当 Lambda 函数的“前门”。API Gateway 负责处理多达数十万个并发 API 调用的接受和处理过程中涉及的所有任务，包括流量管理、授权和访问控制、监控以及 API 版本管理。

要使用 AWS Lambda 提供最佳性能，请为函数选择需要的内存量。我们会按比例为您分配 CPU 功率和其他资源。例如，如果选择 256MB 的内存，则为 Lambda 函数分配的 CPU 功率大约是选择 128MB 内存时的两倍。您可以控制每个函数运行的时间（最多 300 秒）。

了解可用的计算配置选项：了解各种选项如何辅助您的工作负载，以及哪些配置选项最适合您的系统。这些选项的示例包括实例系列、规模、功能（GPU、I/O）、函数大小、容器实例、单租户和多租户。

选择实例系列和类型时，您还必须考虑满足工作负载所需的可用配置选项：

- **图形处理单元 (GPU)** – 使用 GPU 通用计算 (GPGPU) 时，您构建的应用程序可以在开发过程中利用 CUDA 等平台，从而受益于 GPU 提供的高度并行性。如果您的工作负载需要 3D 渲染或视频压缩，GPU 还可以实现硬件加速计算和编码，从而提高工作负载的效率。
- **现场可编程门阵列 (FPGA)** – 使用 FPGA，您可以通过为要求最苛刻的工作负载定制硬件加速执行来优化工作负载。您可以利用受支持的通用编程语言（例如 C 语言或 Go 语言）或面向硬件的语言（例如 Verilog 语言或 VHDL 语言）来定义算法。
- **AWS Inferentia (Inf1)** – 构建 Inf1 实例以支持机器学习推理应用程序。借助 Inf1 实例，客户可以运行大规模机器学习推理应用程序，例如图像识别、语音识别、自然语言处理、个性化和欺诈检测。您可以在 TensorFlow、PyTorch 或 MXNet 等流行的机器学习框架中构建模型，并使用 GPU 实例（例如 P3 或 P3dn）来训练模型。在对机器学习模型进行训练以满足要求之后，您可以使用 AWS Neuron 在 Inf1 实例上部署模型，AWS Neuron 是一种专门的软件开发工具包 (SDK)，它由编译器、运行时和分析工具组成，可优化 Inferentia 芯片的机器学习推理性能。
- **突发性能实例系列** – 突发性能实例可以提供一定的基准性能，并且在您的工作负载需要时突增到更高的性能水平。这些实例适用于不经常或不持续使用全部 CPU 容量，但偶尔需要性能突增的工作负载。突发性能实例非常适合用于各种通用工作负载，例如 Web 服务器、开发人员环境和小型数据库。这些实例提供 CPU 积分，当实例必须提供更高性能时可以使用这些 CPU 积分。积分会在实例不需要时累积。
- **高级计算功能** – Amazon EC2 让您能够使用高级计算功能，例如管理 C 状态和 P 状态注册以及控制处理器的睿频加速。使用协同处理器，您可以通过 AES-NI 分流加密工作，也可以通过 AVX 扩展进行高级计算。

AWS Nitro 系统结合了专用硬件和轻量级管理程序，可实现更快的创新和更高的安全性。在 AWS Nitro 系统可用时，借助该系统可充分利用主机硬件的计算和内存资源。此外，专用 Nitro 卡可实现高速联网、高速 EBS 和 I/O 加速。

收集与计算相关的指标：了解计算系统性能的一个最佳方法是，记录和跟踪各种资源的真实利用率。此数据可用于更准确地确定资源需求。

工作负载（例如在微服务架构上运行的工作负载）可以生成指标、日志和事件形式的大量数据。确定您现有的监控和可观察性服务是否可以管理生成的数据。Amazon CloudWatch 可用于在单个平台上从 AWS 和本地服务器上运行的所有 AWS 资源、应用程序和服务中收集、访问和关联此数据，因此您可以轻松了解整个系统范围的运行情况并快速解决问题。

通过合理调整大小来确定需要的配置：分析您的工作负载的各种性能特性，以及这些特性与内存、网络和 CPU 使用率之间的关系。根据这些数据选择最适合您的工作负载配置文件的资源。例如，实例的 r 系列可以最好地处理内存密集型工作负载（例如数据库）。但是，弹性容器系统可为突增的工作负载提供更多优势。

使用可用的弹性资源：云让您能够通过各种机制灵活地动态扩展或缩减资源，以便满足不断变化的需求。结合与计算相关的指标，工作负载可以自动响应这些变化并利用一系列最优的资源来实现其目标。

实现最佳供需匹配能够尽可能降低工作负载成本，但您也必须准备充足的供应，以便应对预置时间问题和单个资源的故障。需求可以是固定的，也可以是变化的，所以需要通过指标和自动化来确保管理本身不会成为一种负担，而且不会产生不成比例的高成本。

在 AWS 中，您可以使用大量不同方法来实现供需匹配。《成本优化支柱》白皮书描述了如何使用以下方法进行成本优化：

- 基于需求的方法
- 基于缓冲区的方法
- 基于时间的方法

您必须确保工作负载部署可以处理扩展和缩减事件。创建缩减事件的测试方案，以确保工作负载按预期方式运行。

根据指标重新评估计算需求：使用系统级指标来确定工作负载在一段时间内的行为和要求。通过比较可用资源和这些要求来评估工作负载的需求，并对计算环境进行更改以实现与您的工作负载

配置文件的最佳匹配。例如，随着时间的推移，系统可能比最初认为的要更频繁地使用内存，所以转为使用其他实例系列或调整实例大小可能会提高性能和效率。

资源

请参阅以下资源，详细了解有关计算的 AWS 最佳实践。

视频

- Amazon EC2 foundations (CMP211-R2)
- Powering next-gen Amazon EC2: Deep dive into the Nitro system
- Deliver high performance ML inference with AWS Inferentia (CMP324-R1)
- 优化 AWS 计算的性能和成本 (CMP323-R1)
- Better, faster, cheaper compute: Cost-optimizing Amazon EC2 (CMP202-R1)

文档

- 实例：
  - 实例类型
  - EC2 实例的处理器状态控制
- EKS 容器：EKS 工作线程节点
- ECS 容器：Amazon ECS 容器实例
- 函数：Lambda 函数配置

存储架构选择

针对特定系统的最佳存储解决方案往往取决于访问方式的类型（数据块、文件或者对象）、访问模式（随机或者连续）、数据吞吐量要求、访问频率（在线、离线、归档）、更新频率（WORM、动态）以及可用性与持久性限制等因素。架构完善的系统会使用多种存储解决方案，并启用不同的功能来提高性能。

在 AWS 中，存储是虚拟化的，而且具有多种类型。这让您可以更轻松地找到贴合您需求的存储方式，并获得本地基础设施难以实现的存储方案。例如，Amazon S3 可以实现 99.999999999% 的强持久性。您还可以从使用磁性硬盘 (HDD) 改为使用固态硬盘 (SSD)，或在几秒内轻松地完成虚拟驱动器从一个实例到另一个实例的迁移。

性能可以通过吞吐量、每秒输入/输出操作 (IOPS) 和延迟来衡量。了解这些测量值之间的关系将有助于您选择最适合的存储解决方案。

存储	服务	延迟	吞吐量	可共享
数据块	Amazon EBS，EC2 实例存储	最低、一致	单个	安装在 EC2 实例上，通过快照进行复制
文件系统	Amazon EFS、Amazon FSx	低，一致	多个	许多客户端
对象	Amazon S3	低延迟	Web 规模	许多客户端
存档	Amazon S3 Glacier	几分钟到几小时	高	否

从延迟的角度来看，如果只有一个实例访问您的数据，那么您应该使用块存储，例如 Amazon EBS。对于 Amazon EFS 等分布式文件系统来说，每次文件操作的延迟产生的开销通常都很低，所以如果有多个实例需要访问数据，您应该使用分布式文件系统。

Amazon S3 具有可以降低延迟和提高吞吐量的功能。您可以使用跨区域复制 (CRR) 在不同的地理区域中提供低延迟的数据访问。

从吞吐量的角度来看，Amazon EFS 可以支持高度并发的工作负载（例如多个线程和多个 EC2 实例中的并发操作），从而实现较高的聚合吞吐量和每秒操作数。对于 Amazon EFS，请通过基准测试或负载测试来选择适当的性能模式。

了解存储特性和要求：了解在选择最适合您的工作负载的各种服务（例如对象存储、块存储、文件存储或实例存储）时所需的不同特性（例如可共享能力、文件大小、缓存大小、访问模式、延迟、吞吐量和数据持久性）。

确定工作负载的预期增长率，然后选择满足这些增长率的存储解决方案。 对象和文件存储解决方案（例如 Amazon S3 和 Amazon Elastic File System）实现了无限存储。Amazon EBS 具有预定义的存储大小。 弹性卷允许您动态增加容量、调整性能，以及更改任何新的或现有当前卷的类型，而不会导致停机或造成性能影响，但是需要更改 OS 文件系统。

评估可用的配置选项：评估各种特性和配置选项以及它们与存储的关系。了解在何处以及如何使用预置 IOPS、SSD、磁性存储、对象存储、存档存储或短暂存储来针对工作负载优化存储空间和性能。

Amazon EBS 提供一系列选项，让您能够优化工作负载的存储性能和成本。这些选项分为两大类：用于事务型工作负载、由 SSD 提供支持的存储，例如数据库和启动卷（性能主要取决于 IOPS）；用于吞吐量密集型工作负载、由 HDD 提供支持的存储，例如 MapReduce 和日志处理（性能主要取决于传输速度）。

SSD 支持的卷包括：具有最高性能的预置 IOPS SSD 卷，适用于对延迟要求较高的事务型工作负载；通用型 SSD 卷，可以针对各种事务数据实现价格和性能的平衡。

Amazon S3 Transfer Acceleration 可以在您的客户端与 S3 存储桶之间实现快速的远距离文件传输。Transfer Acceleration 利用 Amazon CloudFront 遍布全球的边缘站点，通过优化的网络路径来路由数据。对于 S3 存储桶中具有密集 GET 请求的工作负载，可结合使用 Amazon S3 与 CloudFront。上传大型文件时，使用分段上传同时上传多个部分，以便尽可能提高网络吞吐量。

Amazon Elastic File System (Amazon EFS) 提供了一个简单、可扩展、完全托管的弹性 NFS 文件系统，可配合 AWS 云服务和本地资源使用。为了支持各种云存储工作负载，Amazon EFS 提供了两种性能模式：通用性能模式和最大 I/O 性能模式。对于文件系统，还有两种吞吐量模式可供选择："突增吞吐量"和"预置吞吐量"。要确定对工作负载使用哪种设置，请参阅 Amazon EFS 用户指南。

Amazon FSx 提供两种文件系统可供选择：适用于企业工作负载的 Amazon FSx for Windows File Server 和适用于高性能工作负载的 Amazon FSx for Lustre。FSx 由 SSD 提供支持，旨在提供快速、可预测、可扩展且稳定的性能。Amazon FSx 文件系统提供持续的高读写速度和稳定的低延迟数据访问。您可以选择所需的吞吐量级别来满足工作负载需求。

根据访问模式和指标做出决策：根据工作负载的访问模式选择存储系统，并通过确定工作负载访问数据的方式对其进行配置。通过选择对象存储而不是块存储来提高存储效率。按照您的数据访问模式，配置您选择的存储选项。

访问数据的方式将影响存储解决方案的效果。选择最适合您的访问模式的存储解决方案，或者考虑根据存储解决方案更改访问模式，以便尽可能提高性能。

通过创建 RAID 0（零）阵列，与在单个卷上进行预置相比，您可以实现更高的文件系统性能。当 I/O 性能比容错能力更重要时，请考虑使用 RAID 0。例如，您可以将其用于已经单独设置了数据复制的常用数据库。

在工作负载使用的所有存储选项中，为您的工作负载选择合适的存储指标。当利用使用突增积分的文件系统时，创建警报，以便系统在您即将达到积分限额时通知您。您必须创建存储控制面板以显示工作负载存储的总体运行情况。

对于固定大小的存储系统（例如 Amazon EBS 或 Amazon FSx），请确保您正在监控使用的存储量与总体存储量大小之间的关系，可能的话创建自动化，以便在达到阈值时增加存储大小

资源

请参阅以下资源，详细了解有关存储的 AWS 最佳实践。

视频

- Deep dive on Amazon EBS (STG303-R1)
- Optimize your storage performance with Amazon S3 (STG343)

文档

- Amazon EBS:
  - Amazon EC2 存储
  - Amazon EBS 卷类型

    - I/O 特性

- Amazon S3：请求速率和性能注意事项
- Amazon Glacier：Amazon Glacier 文档
- Amazon EFS：Amazon EFS 性能
- Amazon FSx：
    - Amazon FSx for Lustre 性能
    - Amazon FSx for Windows File Server 性能

数据库架构选择

针对特定系统的最优数据库解决方案取决于您的具体需求，包括可用性、一致性、分区容错性、延迟、持久性、可扩展性以及查询能力等等。许多系统会使用多种不同的数据库解决方案来满足各种子系统的需求，并利用各种不同的功能来提高性能。为系统选择错误的数据库解决方案和功能可能会导致性能效率降低。

了解数据特性：了解工作负载中数据的不同特性。确定工作负载是否需要事务、工作负载如何与数据交互以及工作负载的性能需求有哪些。使用这些数据来选择适用于工作负载的最佳数据库方法（例如关系、NoSQL 键值、文档、宽列、图形、时间序列或内存中存储数据库）。

您可以从许多专用数据库引擎（包括关系、键值、文档、内存、图形、时间序列和分类账数据库）中进行选择。通过选择最佳数据库来解决特定问题或一组问题，您可以摆脱限制性的“一刀切”整体式数据库，并专注于构建应用程序以满足客户需求。

关系数据库通过预定义 schema 及其之间的关系存储数据。这些数据库旨在支持 ACID（原子性、一致性、隔离性、持久性）事务，并保持参照完整性和数据强一致性。许多传统应用程序、企业资源规划 (ERP)、客户关系管理 (CRM) 和电子商务都使用关系数据库来存储其数据。您可以在 Amazon EC2 上运行许多这些数据库引擎，或者从以下 AWS 托管数据库服务中进行选择：Amazon Aurora、Amazon RDS 和 Amazon Redshift。

键值数据库已针对常见的访问模式进行优化，通常用于存储和检索大量数据。这些数据库即使在出现大量并发请求的情况下也能实现快速响应。

高流量 Web 应用程序，电子商务系统和游戏应用程序是键值数据库的典型用例。在 AWS 中，您可以利用 Amazon DynamoDB，这是一个完全托管的多区域、多主表持久数据库，具有适用于互联网规模的应用程序的内置安全性、备份和恢复以及内存中的缓存

内存数据库用于需要实时访问数据的应用程序。对于毫秒延迟不足以满足需求的应用程序，这些数据库通过直接将数据存储在内存中来提供微秒延迟。您可以将内存数据库用于应用程序缓存、会话管理、游戏排行榜和地理空间应用程序。Amazon ElastiCache 是完全托管的内存数据存储，与 Redis 或 Memcached 兼容。

文档数据库旨在将半结构化数据存储为类似 JSON 的文档。这些数据库可帮助开发人员快速构建和更新应用程序，例如内容管理、目录和用户配置文件。Amazon DocumentDB 是一种快速、可扩展、高度可用且完全托管的文档数据库服务，支持 MongoDB 工作负载。

宽列存储是 NoSQL 数据库的一种类型。它使用表、行和列，但是与关系数据库不同的是，同一个表中各行的列名称和格式可能会有所不同。您通常会看到一个宽列存储在大规模工业应用程序中，用于设备维护、队列管理和路线优化。Amazon Managed Apache Cassandra Service 是宽列可扩展、高度可用且兼容托管的 Apache Cassandra 的数据库服务。

图形数据库适用于需要大规模以毫秒延迟在高度连接的图形数据集之间浏览和查询数百万关系的应用程序。许多公司将图形数据库用于欺诈检测、社交网络和推荐引擎。Amazon Neptune 是一种快速、可靠、完全托管的图形数据库服务，可用于轻松构建和运行适用于高度连接的数据集的应用程序。

时间序列数据库可以高效收集、合成数据，并从不断变化的数据中获得见解。IoT 应用程序、开发运营和工业遥测可以利用时间序列数据库。Amazon Timestream 是适用于 IoT 和运营应用程序的快速、可扩展、完全托管的时间序列数据库服务，可用于轻松存储和分析每天数以万亿计的事件。

分类账数据库提供可信中央机构，以维护每个应用程序的可扩展、不可变和允许以加密方式进行验证的交易记录。我们看到分类账数据库用于记录系统、供应链、注册甚至银行交易。Amazon Quantum Ledger Database (QLDB) 是一款完全托管的分类账数据库，提供可信中央机构拥有的透明、不可变和允许以加密方式进行验证的交易日志。Amazon QLDB 跟踪每个应用程序数据更改，并持续维护完整且可验证的更改历史记录。

评估可用的选项：在选择工作负载存储机制的过程中，评估可用的服务和存储选项。了解如何以及何时使用给定的服务或系统进行数据存储。了解可以优化数据库性能或效率的可用配置选项，例如预置 IOPS、内存和计算资源以及缓存。

数据库解决方案通常具有让您能够针对工作负载类型进行优化的配置选项。使用基准测试或负载测试，确定与您的工作负载相关的重要数据库指标。请针对您选择的数据库方法考虑存储优化、数据库级设置、内存和缓存等配置选项。

评估工作负载的数据库缓存选项。以下是三种最常见的数据库缓存类型：

- **数据库集成缓存**：某些数据库（例如 Amazon Aurora）提供了集成的缓存，该缓存在数据库引擎内进行管理并具有内置的直写功能。
- **本地缓存**：本地缓存将您经常使用的数据存储在应用程序中。这样可以加快数据检索的速度，并消除与检索数据相关的网络流量，从而使数据检索比其他缓存架构更快。
- **远程缓存**：远程缓存存储在专用服务器上，通常构建于 Redis 和 Memcached 等键值 NoSQL 存储上。它们每秒在每个缓存节点提供高达一百万个请求。

对于 Amazon DynamoDB 工作负载，DynamoDB Accelerator (DAX) 提供完全托管的内存中缓存。DAX 是一种内存中缓存，允许您使用完全托管的内存中缓存，来大规模地为您的表格提供快速读取性能。使用 DAX，您可以将 DynamoDB 表格的读取性能提高多达 10 倍 – 读取所需的时间从毫秒级缩短为微秒级，甚至每秒可读取数百万个请求。

收集并记录数据库性能指标：使用各种工具、库和系统来记录与数据库性能相关的性能衡量指标。例如，在访问数据库时，测量每秒事务数、慢速查询或产生的系统延迟。根据这些数据来了解您数据库系统的性能。

从您的工作负载中收集尽可能多的数据库活动指标进行检测。这些指标可能需要直接从工作负载中发布，或者从应用程序性能管理服务中收集。您可以使用 AWS X-Ray 分析和调试生产用分布式应用程序，例如使用微服务架构构建的应用程序。X-Ray 跟踪可以包含分段，这些分段封装单个组件的所有数据点。例如，当您的应用程序响应请求而对数据库进行调用时，它会为该请求创建一个分段，其中有一个子分段代表该数据库调用及其结果。该子分段可以包含诸如查询、使用的表、时间戳和错误状态之类的数据。检测后，您应为数据库指标启用警报，以指示何时超出阈值。

根据访问模式选择数据存储：根据工作负载的访问模式来确定要使用的服务和技术。例如，对于需要事务的工作负载，使用关系数据库，或者使用能够提供更高吞吐量但最终保持一致（如适用）的键值存储。

根据访问模式和指标优化数据存储：使用性能特性和访问模式来优化数据存储和查询方式，以便实现最佳性能。衡量索引、键分配、数据仓库设计或缓存策略等优化对系统性能或整体效率的影响。

资源

请参阅以下资源，详细了解有关数据库的 AWS 最佳实践。

视频

- AWS purpose-built databases (DAT209-L)
- Amazon Aurora storage demystified: How it all works (DAT309-R)
- Amazon DynamoDB deep dive: Advanced design patterns (DAT403-R1)

文档

- AWS 数据库缓存
- AWS 云数据库
- Amazon Aurora 最佳实践
- Amazon Redshift 性能
- Amazon Athena 十大性能技巧
- Amazon Redshift Spectrum 最佳实践
- Amazon DynamoDB 最佳实践
- Amazon DynamoDB Accelerator

网络架构选择

适合某个工作负载的最佳网络解决方案会根据延迟、吞吐量要求、抖动和带宽而有所不同。物理限制（例如用户资源或本地资源）决定位置选项。这些限制可以通过边缘站点或资源置放来抵消。

在 AWS 中，网络资源是以虚拟化的形式存在的，而且支持多种类型和配置。这让您可以更轻松地找到贴合您需求的网络方案。AWS 提供多种产品功能（例如增强联网、Amazon EC2 联网优化实例、Amazon S3 Transfer Acceleration、动态 Amazon CloudFront 等）来优化网络流量。AWS 还提供多种联网功能（例如 Amazon Route 53 的基于延迟的路由、Amazon VPC 终端节点、AWS Direct Connect 和 AWS Global Accelerator）来减少网络距离或抖动。

了解网络对性能的影响：分析并了解与网络相关的功能对工作负载性能的影响。例如，网络延迟通常会影响用户体验，而没有提供足够的网络容量可能会导致工作负载性能瓶颈。

由于网络位于所有应用程序组件之间，因此可能会对应用程序性能和行为产生巨大的正面和负面影响。还有一些严重依赖网络性能的应用程序，例如，对于高性能计算 (HPC)，深入了解网络对于提高群集性能很重要。您必须确定带宽、延迟、抖动和吞吐量方面的工作负载要求。

评估可用的联网功能：评估云中提供的有助于提高性能的各种联网功能。借助测试、指标和分析来衡量这些功能的影响。例如，利用可用的网络级功能来减少延迟、网络距离或抖动。

许多服务通常提供相关功能来优化网络性能。请考虑选择合适的产品功能来优化网络流量，如 EC2 实例网络功能、增强联网实例类型、Amazon EBS 优化实例、Amazon S3 Transfer Acceleration 以及动态 CloudFront。

AWS Global Accelerator 是一项使用 AWS 全球网络提高全球应用程序可用性和性能的服务。它利用庞大的无拥塞 AWS 全球网络优化了网络路径。它提供了静态 IP 地址，可轻松在可用区或 AWS 区域之间移动终端节点，而无需更新 DNS 配置或更改面向客户端的应用程序

借助 Amazon S3 内容加速功能，外部用户在向 Amazon S3 传输数据时可以通过 CloudFront 的网络优化获益。这样一来，您可以轻松地将大量数据从没有专用连接的远程位置传输到 AWS 云。

较新的 EC2 实例可以利用增强联网。N 系列的 EC2 实例（例如 M5n 和 M5dn）利用第四代定制 Nitro 卡和 Elastic Network Adapter (ENA) 设备为单个实例提供高达 100 Gbps 的网络吞吐量。与基础 M5 实例相比，这些实例提供了 4 倍的网络带宽和数据包处理能力，是网络密集型应用程序的理想选择。客户还可以在某些实例大小的 M5n 和 M5dn 实例上启用 Elastic Fabric Adapter (EFA)，以实现较低且一致的网络延迟。

Amazon Elastic Network Adapter (ENA) 为单一置放群组中的实例提供 20 Gbps 的网络容量，实现进一步优化。Elastic Fabric Adapter (EFA) 是 Amazon EC2 实例的网络接口，使您能够在 AWS 上大规模运行需要高级别节点间通信的工作负载。借助 EFA，使用消息传递接口 (MPI) 的高性能计算 (HPC) 应用程序和使用 NVIDIA Collective Communications Library (NCCL) 的机器学习 (ML) 应用程序可以扩展到数千个 CPU 或 GPU。

Amazon EBS 优化实例使用经过优化的配置堆栈，可以针对 Amazon EBS I/O 提供额外的专用容量。这种优化通过最小化您的 Amazon EBS I/O 与实例的其他流量之间的争用，来为 EBS 卷提供最佳性能。

Amazon Route 53 的基于延迟的路由 (LBR) 有助于提高工作负载面向全局受众的性能。LBR 的工作方式是，根据工作负载运行的不同 AWS 区域的实际性能衡量指标，将客户路由到可提供最快体验的 AWS 终端节点（EC2 实例、弹性 IP 地址或 ELB 负载均衡器）。

Amazon VPC 终端节点提供面向 AWS 服务（例如，Amazon S3）的可靠连接，无需互联网网关或网络地址转换 (NAT) 实例。

为混合工作负载选择适当大小的专用连接或 VPN：当需要进行本地通信时，请确保您有足够的带宽来保证工作负载性能。 根据带宽要求，单个专用连接或单个 VPN 可能不够，您必须启用多个连接之间的流量负载均衡。

您必须估算混合工作负载的带宽和延迟要求。这些数字将确定 AWS Direct Connect 或您的 VPN 终端节点的大小要求。

AWS Direct Connect 提供了速率为 50 Mbps 到 10 Gbps 的 AWS 环境专用连接。这样一来，延迟得到管理和控制，并且拥有预置带宽，让您的工作负载能够以轻松且高性能的方式连接到其他环境。使用 AWS Direct Connect 合作伙伴之一，您可以拥有多个环境的端到端连接，从而提供性能一致的扩展网络。

AWS 站点到站点 VPN 是 VPC 的托管 VPN 服务。建立 VPN 连接后，AWS 将提供到两个不同 VPN 终端节点的隧道。借助 AWS Transit Gateway，您可以简化多个 VPC 之间的连接，还可以通过单个 VPN 连接来连接到与 AWS Transit Gateway 连接的任何 VPC。AWS Transit Gateway 还可以通过在多个 VPN 隧道上启用等价多路径 (ECMP) 路由支持，使您扩展到 1.25 Gbps IPsec VPN 吞吐量限制之外。

利用负载均衡和加密卸载：将流量分布在多个资源或服务上，以使您的工作负载能够利用云提供的弹性。您也可以使用负载均衡机制来卸载加密终端，以便提高性能并有效管理和路由流量。

在实施想要在其中针对服务内容使用多个实例的横向扩展架构时，您可以利用 Amazon VPC 内部的负载均衡器。AWS 为 ELB 服务中的应用程序提供了多个模型。Application Load Balancer 最适合 HTTP 和 HTTPS 流量的负载均衡，面向交付包括微服务和容器在内的现代应用程序架构，提供高级请求路由功能。

若要对需要极高性能的 TCP 流量进行负载均衡，网络负载均衡器是最佳选择。网络负载均衡器每秒能够处理数百万请求，同时能保持超低延迟，还针对处理突发和不稳定的流量模式进行了优化。

Elastic Load Balancing 提供集成的证书管理和 SSL/TLS 解密，使您可以灵活地集中管理负载均衡器的 SSL 设置，并从工作负载中卸载占用大量 CPU 的工作。

选择网络协议以优化网络流量：根据对工作负载性能的影响，制定有关系统与网络之间的通信协议的决策。

延迟和带宽之间存在关系，以实现吞吐量。如果文件传输使用 TCP 协议，则延迟越高，整体吞吐量越低。有一些方法可以使用 TCP 调整和优化的传输协议来解决此问题，有些方法则使用 UDP 协议。

根据网络要求选择位置：使用可用的云位置选项来降低网络延迟或提高吞吐量。利用 AWS 区域、可用区、置放群组和边缘站点（例如 Outposts、本地扩展区和 Wavelength），来降低网络延迟或提高吞吐量。

AWS 云基础设施围绕区域和可用区构建。区域是指全球范围内的某个物理位置，每个区域有多个可用区。

可用区由一个或多个分散的数据中心组成，每个数据中心都拥有独立的配套设施，其中包括冗余电源、联网和连接。可用区能够提高生产应用程序和数据库的运行效率，使其具备比单个数据中心更强的可用性、容错能力以及可扩展性

请根据以下关键元素，为您的部署选择一个或多个合适的区域：

- **用户所在位置：**选择一个接近您的工作负载用户的区域，确保他们在使用工作负载时延迟较低。
- **数据所在位置：**对于数据密集型应用程序，延迟方面的主要瓶颈是数据传输。应用程序代码的执行位置应尽量接近数据位置。
- **其他制约：**考虑安全性和合规性等制约。

Amazon EC2 为联网提供置放群组。置放群组是单个可用区内实例的逻辑分组。使用具有支持的实例类型和 Elastic Network Adapter (ENA) 的置放群组，可使工作负载参与低延迟的 25 Gbps 网络。建议将置放群组用于可受益于低网络延迟和/或高网络吞吐量的工作负载。使用置放群组有降低网络通信抖动的优势。

延迟敏感型服务是使用全球边缘站点网络在边缘交付的。这些边缘站点通常提供内容分发网络 (CDN) 和域名系统 (DNS) 等服务。通过在边缘交付这些服务，工作负载可以低延迟响应内容或

DNS 解析请求。这些服务还提供地理定位服务，例如内容地理定位（基于最终用户位置提供不同内容），或基于延迟的路由，用于将最终用户引导至最近的区域（最小延迟）。

Amazon CloudFront 是一个全球性内容分发网络 (CDN)，可用于加速静态内容（如图像、脚本和视频）以及动态内容（如 API 或 Web 应用程序）。它依赖于全球边缘站点网络，可以缓存内容并为您的用户提供高性能的网络连接。CloudFront 也加快了许多其他功能，如内容上传和动态应用程序，从而使通过互联网提供流量的所有应用程序的性能有所提高。Lambda@Edge 是 Amazon CloudFront 的一项功能，使您可以更接近工作负载用户运行代码，从而提高性能并减少延迟。

Amazon Route 53 是一种高度可用且可扩展的云 DNS Web 服务。它的目的是为开发人员和企业提供一种非常可靠且经济高效的方式，将名称（如 www.example.com）转换为计算机用于互相连接的数字 IP 地址（如 192.168.2.1），从而将最终用户路由到互联网应用程序。Route 53 与 IPv6 完全兼容。

AWS Outposts 专为因延迟要求而需要保留在本地的工作负载而设计，在此您希望该工作负载与 AWS 中的其他工作负载一起无缝运行。AWS Outposts 是完全托管且可配置的计算和存储机架，这些机架使用 AWS 设计的硬件构建，可让您在本地运行计算和存储，同时无缝连接到云中 AWS 的广泛服务。

AWS 本地扩展区是一种新型的 AWS 基础设施，旨在运行需要十毫秒内延迟的工作负载，例如视频渲染和图形密集型虚拟桌面应用程序。本地扩展区使您可以获得使计算和存储资源更接近最终用户的所有优势。

AWS Wavelength 通过将 AWS 基础设施、服务、API 和工具扩展到 5G 网络，旨在向 5G 设备提供超低延迟应用程序。Wavelength 将存储和计算嵌入电信运营商 5G 网络内部，以在您的 5G 工作负载需要十毫秒内延迟时提供帮助，例如 IoT 设备、游戏流、自动驾驶汽车和实时媒体制作。

可使用边缘服务来减少延迟并启用内容缓存。请确保您为 DNS 和 HTTP/HTTPS 正确配置了缓存控制，以便通过这些方式获得最大优势。

根据各项指标优化网络配置：使用收集和分析的数据做出有关优化网络配置的明智决策。衡量更改带来的影响，并且根据衡量结果来做出进一步决策。

为您的工作负载使用的所有 VPC 网络启用 VPC 流日志。VPC 流日志功能使您能够进一步捕获有关传入和传出您的 VPC 中网络接口的 IP 流量的信息。VPC 流日志可帮助您完成许多任务，例如

解决为什么特定流量无法到达实例的问题，进而帮助您诊断过于严格的安全组规则。您可以使用流日志作为安全工具来监控到达实例的流量，以分析网络流量并查找异常的流量行为。

使用网络指标来随着工作负载的发展对网络配置进行更改。基于云的网络可以快速重建，因此有必要随着时间的推移改进网络架构，以保持性能效率。

资源

请参阅以下资源，详细了解有关联网的 AWS 最佳实践。

视频

- Connectivity to AWS and hybrid AWS network architectures (NET317-R1)
- Optimizing Network Performance for Amazon EC2 Instances (CMP308-R1)

文档

- 过渡到 Amazon Route 53 中基于延迟的路由
- AWS 联网产品
- EC2
  - Amazon EBS – 优化的实例
  - Linux 上的 EC2 增强联网
  - Windows 上的 EC2 增强联网
  - EC2 置放群组
  - 在 Linux 实例上启用 Elastic Network Adapter (ENA) 增强联网
- VPC
  - Transit Gateway
  - VPC 终端节点
  - VPC 流日志
- Elastic Load Balancer
  - Application Load Balancer
  - 网络负载均衡器

审核

在最初构建工作负载时，您可以选择的选项很有限。但是随着时间的推移，可提升工作负载性能的新技术和方法会不断涌现。在云中，由于基础设施是代码，因此试验新功能和服务会简单得多。

要采用数据驱动的架构方法，您应该实施性能审核流程，并考虑以下内容：

- **基础设施即代码**：使用 AWS CloudFormation 模板之类的方法定义您的基础设施即代码。使用模板，您可以将您的基础设施与应用程序代码和配置一道放入源代码控制中。这使您能够将用于开发软件的实践应用到基础设施，从而能够快速迭代。
- **部署管道**：使用持续集成/连续部署 (CI/CD) 管道（例如，源代码存储库、构建系统、部署和测试自动化）来部署您的基础设施。这使您能够以可重复、一致且低成本的方式进行迭代部署。
- **明确定义的指标**：设置您的指标和监控以捕获关键性能指标 (KPI)。我们建议您使用技术和业务指标。网站或移动应用程序的关键指标是首个字节捕获时间或渲染时间。其他常规的适用指标包括线程计数、垃圾收集速率以及等待状态。业务指标，如单次请求累计总成本，可以提醒您留意降低成本的方法。仔细考虑解读指标的方式。例如，您可以选择最大值或第 99 个百分位数，而不是平均值。
- **自动性能测试**：作为部署过程的一部分，在快速运行测试成功通过后自动触发性能测试。自动化应创建新环境、设置初始条件（如测试数据），然后执行一系列基准和负载测试。这些测试的结果应回绑到构建中，以便您可以随着时间推移跟踪性能变化。对于长时间运行测试，您可以使管道的这一部分与构建剩余部分实现异步。或者，您可以使用 Amazon EC2 Spot 实例来执行通宵性能测试。
- **负载生成**：您应该创建复制综合或预先记录的用户旅程的一系列测试脚本。这些脚本应该是幂等的，而不是耦合，您可能需要包含“预热”脚本以便产生有效结果。测试脚本应尽可能复制生产中的使用行为。您可以使用软件或软件即服务 (SaaS) 解决方案来生成负载。考虑使用 AWS Marketplace 解决方案和 Spot 实例 – 它们是用于生成负载的经济高效的方法。

- **性能可见性**：关键指标应该对您的团队可见，尤其是针对每个构建版本的指标。这让您能够随着时间推移看到所有重大的正面或负面趋势。您还应展示有关错误或异常数量的指标，以确保测试的是正常工作的系统。
- **可视化**：使用可视化技术，清楚了解出现性能问题、热点、等待状态或低利用率的位置。在架构图上叠加性能指标 – 调用图表或代码有助于快速发现问题。

此性能审核流程可以实施为现有部署管道的简单扩展，然后随着测试要求复杂程度逐渐加深而发展演变。对于未来的架构，您可以归纳方法并重复使用相同的流程和构件。

通常，不存在或损坏的性能审核过程会导致架构性能不佳。如果您的架构性能不佳，请实施性能审核流程，以便应用戴明的计划-执行-检查-处理 (PDCA) 循环来驱动迭代改进。

改进工作负载以便利用新的版本

利用由客户需求驱动的 AWS 持续创新。我们会定期发布新的区域、边缘站点、服务和功能。这些发布内容都可以明显提高架构的性能效率。

及时了解最新的资源和服务：当新服务、设计模式和产品问世时，评估可以提高性能的方法。通过临时评估、内部讨论或外部分析来确定哪些些方法可以提高性能或提高工作负载的效率。

制定相应流程，评估 AWS 推出的更新、新功能和新服务。例如，使用新技术构建概念验证或咨询内部团队。在尝试新想法或新服务时，运行性能测试，以衡量这些新想法或新服务对工作负载的效率或性能的影响。利用您在 AWS 上获得的灵活性，经常对新想法或新技术进行测试，以尽量减少成本或风险。

制定流程来提高工作负载性能：制定相应流程，以在新的服务、设计模式、资源类型和配置推出后，对它们进行评估。例如，对新实例产品运行现有性能测试，以确定它们改进工作负载的潜力。

工作负载的性能会面临一些关键约束。记录这些约束，以便您了解哪些创新可以提高工作负载的性能。当您知道有新的服务或技术推出时，借助这些信息来确定消除约束或瓶颈的方法。

随着时间的推移改进工作负载：组织需要使用在评估流程中收集的信息，积极推动对新推出的服务或资源的采用。

利用评估新服务或新技术时收集的信息来推动变革。随着您的业务或工作负载发生改变，性能需求也会改变。使用从工作负载指标中收集的数据来评估在哪些方面可以获得最大的效率或性能提升，并且积极采用新服务和新技术来紧跟需求。

资源

请参阅以下资源，详细了解有关基准测试的 AWS 最佳实践。

视频

- Amazon Web Services YouTube 频道
- AWS 在线技术讲座 YouTube 频道
- AWS 事件 YouTube 频道

监控

实施架构后，必须监控其性能，以便在问题对客户造成影响之前进行补救。您应该使用监控指标，确保系统在指标超出阈值时发出告警。

AWS 监控包含五个不同阶段，更多详细信息可参阅可靠性支柱白皮书：

1. **生成** – 监控、指标和阈值的范围
2. **聚合** – 基于多个源创建完整视图
3. **实时处理和报警** – 识别和响应
4. **存储** – 数据管理和保留策略
5. **分析** – 控制面板、报告和见解

Amazon CloudWatch 是一项针对 AWS 云资源和在 AWS 上运行的工作负载的监控服务。您可以使用 CloudWatch 收集和跟踪指标、收集和监控日志文件，以及设置告警。CloudWatch 可以监控各种 AWS 资源，例如 EC2 实例和 RDS 数据库实例，以及由您的工作负载和服务生成的自定义指标和您的应用程序生成的所有日志文件。您可以使用 CloudWatch 全面地了解资源使用率、应用程序性能和运行状况。使用这些分析结果，您可以快速做出反应，保证工作负载顺畅运行。

CloudWatch 控制面板允许您创建可重复使用的 AWS 资源图表和自定义指标，以便您监控运行状态并迅速找出问题。

有效监控解决方案的关键是确保不会看到误报。自动触发器可以避免人为错误，并且可以缩短解决问题的用时。请安排时间在生产环境中执行模拟以测试告警解决方案，确保它可以正确识别各种问题。

监控解决方案分为两种类型：主动监控 (AM) 和被动监控 (PM)。AM 和 PM 互为补充，使您能够概要了解工作负载的表现。

主动监控跨产品中的关键路径模拟脚本化用户旅程中的用户活动。AM 应连续执行，以便测试工作负载的性能和可用性。AM 的连续、轻便和可预测的特性有力补充了 PM。它可以跨所有环境（尤其是预生产环境）运行，能够及时发现问题或性能问题，防止最终用户受到影响。

被动监控通常用于基于 Web 的工作负载。PM 从浏览器收集性能指标（不是基于 Web 的工作负载可以使用类似的方法）。您可以跨所有用户（或用户子集）、地理位置、浏览器和设备类型收集指标。使用 PM，了解以下问题：

- **用户体验性能**：PM 为您提供有关用户体验的指标，让您可以持续了解生产运行情况，并让您了解变化产生的影响。
- **地理性能变化**：如果工作负载遍布全球，用户从世界各地访问工作负载，那么使用 PM 可以确保您能发现影响特定地理位置用户的性能问题。
- **使用 API 的影响**：现代工作负载使用内部 API 和第三方 API。PM 可以让您了解 API 的使用情况，以便您能识别源自内部 API 和第三方 API 提供商的性能瓶颈。

CloudWatch 可以提供监控并发送通知告警。您可以使用自动化技术通过 Amazon Kinesis（流数据处理服务）、Amazon Simple Queue Service (Amazon SQS) 和 AWS Lambda（无服务器计算服务）来触发操作，响应处理性能问题。

监控资源以便确保其性能符合预期

系统性能会随着时间的推移而降低。监控系统性能，以发现性能降低的情况，并针对内部或外部因素（例如操作系统或应用程序负载）采取修复措施。

记录与性能相关的指标：使用监控和可观察性服务记录与性能相关的指标。例如，记录数据库事务、慢速查询、I/O 延迟、HTTP 请求吞吐量、服务延迟或其他关键数据。

确定对工作负载至关重要的性能指标并记录下来。这些数据对于确定影响工作负载整体性能或效率的组件非常重要。

回顾客户体验，确定至关重要的指标。针对每个指标，确定其目标、衡量方式和优先程度。根据这些信息创建告警和通知，以主动解决与性能相关的问题。

在发生事件或意外事件时分析各项指标：在某个事件或意外事件发生后（或发生过程中），使用监控控制面板或报告来了解和诊断影响。这些视图可让您了解工作负载哪些部分的性能没有达到预期。

针对架构编写重要用户案例时，请纳入性能要求，例如指定每个重要案例应以多快速度执行。对于这些重要案例，实施额外的脚本化用户体验，以便确保您知道这些案例是如何根据您的要求执行的

建立关键性能指标 (KPI) 来衡量工作负载性能：确定用于指示工作负载是否按预期执行的 KPI。例如，基于 API 的工作负载可以使用整体响应延迟来指示整体性能，电子商务网站可以使用购买量作为其 KPI。

记录客户要求的性能体验，包括客户如何评价工作负载的性能。根据这些要求确定关键性能指标 (KPI)，用于指示系统的整体性能。

借助监控来生成基于告警的通知：根据您定义的与性能相关的关键性能指标 (KPI)，使用当测量值超出预期范围时能够自动生成告警的监控系统。

Amazon CloudWatch 可以收集架构中各种资源的指标。您也可以收集和发布自定义指标，用于显示业务指标或派生指标。使用 CloudWatch 或第三方监控服务设置超出阈值告警；告警表明指标超出预期范围。

定期检查指标：在例行维护时或者事件或意外事件发生后，检查收集到了哪些指标。通过这些检查，找出哪些指标对于解决问题至关重要，以及跟踪哪些其他指标会有助于发现、解决问题或预防问题发生。

在响应意外事件或事件的过程中，评估哪些指标有助于解决问题、哪些目前没有跟踪的指标会有助于解决问题。这样，您可以提高收集的指标的质量，从而预防或更快速地解决未来发生的意外事件。

主动监控和告警：使用关键性能指标 (KPI) 并结合监控和告警系统，主动解决与性能相关的问题。使用告警触发自动操作，在可能的情况下修复问题。如果无法实现自动响应，则将告警上报给能够响应的人员。例如，您的系统在关键性能指标 (KPI) 超出特定阈值时，能够预测预期 KPI 值并发出告警的系统；或者您的工具在 KPI 超出预期值时，能够自动停止或回滚部署的工具。

实施相应流程，让您在工作负载运行期间了解其性能。构建监控控制面板并确定预期性能基准，以确定工作负载的性能是否达到最佳。

资源

请参阅以下资源，详细了解有关监控的 AWS 最佳实践，以便提升性能效率。

视频

- Cut through the chaos: Gain operational visibility and insight (MGT301-R1)

文档

- X-Ray 文档
- CloudWatch 文档

权衡

在架构解决方案时，需要权衡各种因素才能确保获得最佳方案。根据具体情况，您可以在一致性、持久性和空间与时间或延迟之间进行权衡，以便实现更高的性能。

使用 AWS，您可以在几分钟内实现全球化部署，并可在世界范围内的多个位置部署资源，从而缩短与最终用户的距离。您还可以将只读副本动态添加到信息存储系统（例如数据库系统），以减少主数据库的负载。

AWS 可以提供 Amazon ElastiCache 和 Amazon CloudFront 等缓存解决方案。其中，Amazon ElastiCache 可以提供内存数据存储或缓存，Amazon CloudFront 则可以将静态内容的副本缓存在更接近最终用户的位置。Amazon DynamoDB Accelerator (DAX) 提供位于 Amazon DynamoDB 前面的直读/直写分布式缓存层，支持相同的 API，对缓存中的对象可以实现亚毫秒级的访问。

权衡各种因素以改善性能

在架构解决方案时，积极权衡各种因素可以帮助您选出最佳方法。通常，您可以用一致性、持久性和空间来换取时间和延迟，从而提高性能。权衡各种因素可能会增加架构的复杂性，而且需要进行负载测试以便确保获得可以量化的收益。

了解对性能最至关重要的因素：了解并确定在哪些方面提高工作负载性能，会对效率或客户体验产生积极的影响。例如，拥有大量客户交互的网站会因为使用边缘服务在距离客户更近的位置向客户分发内容而受益。

了解设计模式和服务：研究和理解有助于提高工作负载性能的各种设计模式和服务。在分析的过程中，确定您需要牺牲哪些方面来实现更高的性能。例如，使用缓存服务有助于减少数据库系统上的负载；不过，这需要您完成一些设计工作，以实现安全的缓存，或者可能需要在某些方面实现最终一致性。

了解您可以使用哪些性能配置选项以及这些配置选项对工作负载的影响。优化工作负载性能依赖于对以下内容的了解：这些选项如何与架构进行交互，以及这些选项对实际测量的性能和用户感知到的性能的影响。

Amazon Builders' Library 为读者提供了有关 Amazon 如何构建和运营技术的详细说明。这些免费文章均由 Amazon 的高级工程师撰写，其中涵盖架构、软件交付和运营等诸多主题。例如，您可以看到 Amazon 如何实现软件交付自动化，每年完成超过 1.5 亿次部署；或者 Amazon 的工程师如何实施随机分片等原理来构建具有高可用性和容错能力的弹性系统。

确定权衡机制对客户和效率的影响：在评估与性能相关的改进时，确定哪种选择会对客户和工作负载效率产生影响。例如，如果使用键值数据存储可以提高系统性能，那么评估它的最终一致性将如何影响客户就非常重要。

通过指标和监控确定系统中性能不佳的方面。确定如何提高性能、性能提高带来的利弊，并确定性能提高对系统和用户体验的影响。例如，缓存数据有助于大幅提高性能，但需要就如何以及何时更新缓存的数据或使其变得无效而制定明确的策略，以防止产生不正确的系统行为。

衡量性能改善的影响：在做出更改以提高性能时，请评估收集到的指标和数据。使用这些信息来确定性能提高对工作负载、工作负载组件和客户的影响。这种衡量可让您了解采用权衡机制后实现的性能提高，还可以帮助确定性能提高是否产生了任何不利的副作用。

架构完善的系统会使用各种与性能相关的策略。确定哪种策略会对给定的热点或瓶颈产生最大的积极影响。例如，对多个关系数据库系统中的数据进行分片可以提高整体吞吐量并保持对事务的支持，而且在每个分片内进行缓存有助于降低负载。

使用各种与性能相关的策略：如果合适，利用多种策略来提高性能。例如，可以使用缓存数据等策略来防止出现过多的网络或数据库调用；使用数据库引擎的只读副本来提高读取速度；尽可能对数据进行分片或压缩以减少数据卷；在数据可用时进行缓冲和传输，避免拥堵。

对工作负载进行更改时，需要收集并评估各项指标，以确定更改产生的影响。衡量对系统和最终用户的影响，以便了解权衡机制如何影响工作负载。使用负载测试等系统的方法来确定权衡机制是否可以提高性能。

资源

请参阅以下资源，详细了解有关缓存的 AWS 最佳实践。

视频

- Introducing The Amazon Builders' Library (DOP328)

文档

- Amazon Builders' Library
- 实施 Amazon ElastiCache 的最佳实践

总结

实现和维护性能效率需要一个数据驱动的方法。您应积极考虑访问模式和权衡机制，以便通过优化来实现更高的性能。使用基于基准和负载测试的审核流程，您可以选择合适的资源类型和配置。将基础设施视为代码，您能够快速、安全地改进您的架构，同时使用数据来制定有关架构的基于事实的决策。结合使用主动监控和被动监控，能够确保架构性能不随着时间推移而降低。

AWS 致力于帮助您构建性能高效、同时提供业务价值的架构。使用本文中讨论的工具和技术，以确保成功。

贡献者

以下是对本文档做出贡献的个人和组织：

- Eric Pullen，Amazon Web Services Well-Architected 性能效率主管
- Philip Fitzsimons，Amazon Web Services Well-Architected 高级经理
- Julien Lépine，Amazon Web Services 专家级 SA 经理
- Ronnen Slasky，Amazon Web Services 解决方案架构师

延伸阅读

如需更多帮助，请查阅以下资源：

- AWS 架构完善的框架

文档修订

日期	描述
2020 年 4 月	v2 的主要更新
2018 年 7 月	语法问题的次要更新
2017 年 11 月	刷新白皮书以反映 AWS 中的变化
2016 年 11 月	首次发布