Whitepaper zu AWS

Bewährte Methoden für Designmuster: Optimieren der Leistung von Amazon S3

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Bewährte Methoden für Designmuster: Optimieren der Leistung von Amazon S3: Whitepaper zu AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, die Kunden zu verwirren oder Amazon in einer Weise herabzusetzen oder zu diskreditieren. Alle anderen Marken, die nicht Eigentum von Amazon sind, sind Eigentum ihrer jeweiligen Inhaber, die mit Amazon verbunden oder nicht verbunden oder von Amazon gesponsert oder nicht gesponsert sein können.

Table of Contents

Überblick 1
Überblick 1
Einführung 2
Anleitungen zur Leistung von Simple Storage Service (Amazon S3) 4
Messen der Leistung 4
Horizontale Skalierung von Speicherverbindungen 4
Verwenden von Byte Range Fetches 5
Wiederholungsanforderungen für latenzsensitive Anwendungen 5
Kombinieren von Amazon S3 (Storage) und Amazon EC2 (Computing) in der gleichen AWS-Region 5
Verwenden von Amazon S3 Transfer Acceleration zur Minimierung der durch die Entfernung verursachten Latenz 6
Verwendung der neuesten Version der AWS SDKs 6
Leistungsdesignmuster für Simple Storage Service (Amazon S3) 7
Verwendung von Caching für Inhalte mit häufigen Zugriffen 7
Timeouts und Wiederholungsversuche für latenzsensitive Anwendungen 8
Horizontale Skalierung und Anforderungsparallelisierung für hohen Durchsatz 9
Verwendung von Amazon S3 Transfer Acceleration zur Beschleunigung geographisch disparater Datenübertragungen 11
Mitwirkende 12
Dokumentversionen 13
Hinweise 14

Bewährte Methoden für Designmuster: Optimieren der Leistung von Amazon S3

Datum der Erstveröffentlichung: Juni 2019 (Dokumentversionen)

Überblick

Bei der Entwicklung von Anwendungen, die Speicher zu Simple Storage Service (Amazon S3) hochladen und davon abrufen, sollten Sie die bewährten Methoden von AWS befolgen, um die Leistung zu optimieren. AWS bietet auch detailliertere Leistungsdesignmuster.

Einführung

Ihre Anwendungen erreichen schnell Tausende von Transaktionen pro Sekunde bei der Anfrageleistung, wenn Speicherinhalte zu Amazon S3 hochgeladen oder von dort abgerufen werden. Amazon S3 wird automatisch auf hohe Anfrageraten skaliert. Ihre Anwendung kann beispielsweise mindestens 3.500 `PUT/COPY/POST/DELETE` und 5.500 `GET/HEAD` Anforderungen pro Sekunde pro Präfix in einem Bucket erfüllen. Es gibt keine Einschränkungen für die Anzahl der Präfixe in einem Bucket. Sie können Ihre Lese- und Schreibleistung steigern, indem Sie Lesevorgänge parallelisieren. Wenn Sie beispielsweise 10 Präfixe in einem Amazon S3-Bucket für parallele Lesevorgänge einrichten, können Sie damit die Leseleistung auf 55 000 Leseanfragen pro Sekunde skalieren.

So prüfen etwa manche Data-Lake-Anwendungen in Simple Storage Service (Amazon S3) viele Millionen oder Milliarden von Objekten auf Abfragen, die über Petabytes von Daten ausgeführt werden. Solche Data-Lake-Anwendungen erreichen Single-Instance-Übertragungsraten, die die Netzwerkschnittstellennutzung für ihre Amazon-EC2-Instance maximieren und bis zu 100 GB/s auf einer einzelnen Instance erreichen können. Anschließend aggregieren diese Anwendungen den Durchsatz über mehrere Instances hinweg, um mehrere Terabit pro Sekunde zu erreichen.

Andere Anwendungen sind latenzempfindlich, wie etwa Social-Media-Messaginganwendungen. Diese Anwendungen können konsistente Small Object-Latenzen (und First-byte-out-Latenzen für große Objekte) von etwa 100-200 Millisekunden erreichen.

Andere AWS-Services können auch dabei helfen, die Leistung für unterschiedliche Anwendungsarchitekturen zu beschleunigen. Wenn Sie beispielsweise höhere Übertragungsraten über eine einzelne HTTP-Verbindung oder Latenzzeiten im einstelligen Millisekundenbereich wünschen, verwenden Sie Amazon CloudFront oder Amazon ElastiCache für die Zwischenspeicherung mit Amazon S3.

Und wenn Sie eine schnelle Datenübertragung über große Distanzen zwischen einem Client und einem S3-Bucket wünschen, verwenden Sie Amazon S3 Transfer Acceleration. Transfer Acceleration verwendet die weltweit verteilten Edge-Standorte in CloudFront zur Beschleunigung des Datentransports über geographische Entfernungen hinweg.

Wenn Ihr Workload für Simple Storage Service (Amazon S3) die serverseitige Verschlüsselung über AWS Key Management Service (SSE-KMS) verwendet, finden Sie unter AWS KMS-Limits im AWS Key Management Service-Entwicklerhandbuch Informationen zu den für Ihren Anwendungsfall unterstützten Anfrageraten.

Die folgenden Themen beschreiben bewährte Verfahren und Designmuster zur Optimierung der Leistung von Anwendungen, die Amazon S3 verwenden.

Diese Anleitungen ersetzen alle vorherigen Anleitungen zur Optimierung der Leistung für Amazon S3. Zum Beispiel empfahlen Leistungsanleitungen für Amazon S3 früher, die Präfixbenennung mit gehashten Zeichen zu randomisieren, um die Leistung für häufige Datenabrufe zu optimieren. Sie müssen Präfixbenennungen nicht mehr für Leistung randomisieren und können sequenzielle datumsbasierte Benennungen für Ihre Präfixe verwenden. Die aktuellsten Informationen zur Leistungsoptimierung bei Simple Storage Service (Amazon S3) finden Sie in den Leistungsrichtlinien und Leistungsdesignmustern.

Anleitungen zur Leistung von Simple Storage Service (Amazon S3)

Zur Erzielung der besten Leistung für Ihre Anwendung auf Simple Storage Service (Amazon S3) empfiehlt AWS die folgenden Vorgehensweisen.

Themen

- Messen der Leistung
- Horizontale Skalierung von Speicherverbindungen
- Verwenden von Byte Range Fetches
- Wiederholungsanforderungen für latenzsensitive Anwendungen
- Kombinieren von Amazon S3 (Storage) und Amazon EC2 (Computing) in der gleichen AWS-Region
- Verwenden von Amazon S3 Transfer Acceleration zur Minimierung der durch die Entfernung verursachten Latenz
- Verwendung der neuesten Version der AWS SDKs

Messen der Leistung

Beachten Sie bei der Leistungsoptimierung die Anforderungen an Netzwerkdurchsatz, CPU und Dynamic Random Access Memory (DRAM). Je nach der Mischung der Anforderungen für diese verschiedenen Ressourcen kann es sinnvoll sein, verschiedene Amazon EC2-Instance-Typen zu erwägen. Weitere Informationen zu verfügbaren Instance-Typen finden Sie unter Verfügbare Instance-Typen im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Weiterhin ist es für die Messung der Leistung sinnvoll, die DNS-Lookup-Zeit, die Latenz und die Datenübertragungsgeschwindigkeit mithilfe von HTTP-Analysetools zu untersuchen.

Horizontale Skalierung von Speicherverbindungen

Die Verteilung von Anfragen über zahlreiche Verbindungen ist ein übliches Designmuster für die horizontale Skalierung der Leistung. Wenn Sie hochleistungsfähige Anwendungen erstellen, stellen Sie sich Amazon S3 als sehr großes verteiltes System vor, nicht als einen einzelnen Netzwerkendpunkt wie ein herkömmlicher Speicherserver. Sie erreichen die beste Leistung durch

die Ausgabe mehrerer gleichzeitiger Anfragen an Amazon S3. Verteilen Sie diese Anfragen über separate Verbindungen, um die von Amazon S3 aus zugängliche Bandbreite zu maximieren. Amazon S3 hat keine Beschränkungen für die Anzahl der Verbindungen, die mit Ihrem Bucket hergestellt werden.

Verwenden von Byte Range Fetches

Mit dem Range-HTTP-Header in einer GET Object-Anfrage können Sie einen Bytebereich von einem Objekt abrufen, wobei nur der angegebene Teil übertragen wird. Sie können gleichzeitige Verbindungen zu Amazon S3 verwenden, um verschiedene Bytebereiche aus demselben Objekt abzurufen. Dies hilft beim Erreichen eines höheren aggregierten Durchsatzes als bei einer einzelnen Anforderung eines ganzen Objekts. Der Abruf kleinerer Bereiche eines größeren Objekts ermöglicht Ihrer Anwendung auch die Verbesserung der Zeiten für Wiederholungsversuche, wenn Anforderungen unterbrochen werden. Weitere Informationen finden Sie unter Objekte abrufen.

Typische Größen für Bytebereichanforderungen sind 8 oder 16 MB. Wenn für Objekte eine PUT-Aktion unter Verwendung eines mehrteiligen Uploads durchgeführt werden, ist es sinnvoll, die GET-Aktion mit denselben Teilgrößen (oder zumindest orientiert an den Teilgrenzen) durchzuführen, um eine optimale Leistung zu erzielen. GET-Anforderungen können sich direkt auf einzelne Teile richten, zum Beispiel `GET ?partNumber=N`.

Wiederholungsanforderungen für latenzsensitive Anwendungen

Aggressive Timeouts und Wiederholungsversuche sorgen für konsistente Latenz. Aufgrund des Umfangs von Amazon S3 gilt: Wenn die erste Anforderung langsam ist, nimmt eine alte Anfrage wahrscheinlich einen anderen Pfad und ist schnell erfolgreich. Die AWS SDKs verfügen über konfigurierbare Timeout- und Wiederholungsversuchwerte, die Sie an die Toleranzen Ihrer spezifischen Anwendung anpassen können.

Kombinieren von Amazon S3 (Storage) und Amazon EC2 (Computing) in der gleichen AWS-Region

Obwohl Namen von S3-Buckets global eindeutig sind, wird jeder Bucket in einer Region gespeichert, die Sie bei dessen Erstellung auswählen. Zur Optimierung der Leistung empfehlen wir, nach Möglichkeit auf den Bucket von Amazon EC2-Instances in derselben AWS-Region aus zuzugreifen. Dies hilft bei der Reduzierung der Netzwerklatenz und der Datenübertragungskosten.

Weitere Informationen zu den Kosten von Datenübertragungen finden Sie unter Amazon S3 – Preise.

Verwenden von Amazon S3 Transfer Acceleration zur Minimierung der durch die Entfernung verursachten Latenz

Amazon S3 Transfer Acceleration verwaltet die schnelle, einfache und sichere Übertragung von Dateien über größere Entfernungen zwischen Ihrem Client und einem S3-Bucket. Transfer Acceleration nutzt die weltweit verteilten Edge-Standorte von Amazon CloudFront. Sobald die Daten an einem Edge-Standort eingehen, werden sie über einen optimierten Netzwerkpfad an Amazon S3 weitergeleitet. Transfer Acceleration ist ideal für die regelmäßige Übertragung von Daten im Gigabyte- bis Terabyte-Bereich von Kontinent zu Kontinent geeignet. Die Funktion ist auch für Kunden nützlich, die Uploads in einen zentralen Bucket aus der ganzen Welt vornehmen.

Sie können das Amazon S3 Transfer Acceleration Speed Comparison-Tool verwenden, um beschleunigte und nicht beschleunigte Upload-Geschwindigkeiten in allen Regionen von Simple Storage Service (Amazon S3) zu vergleichen. Das Speed Comparison-Tool verwendet mehrteilige Uploads, um eine Datei von Ihrem Browser in verschiedene AWS S3-Regionen mit und ohne Amazon S3 Transfer Acceleration zu übertragen.

Verwendung der neuesten Version der AWS SDKs

Die AWS SDKs bieten integrierte Unterstützung für viele Empfehlungen zur Optimierung der Amazon S3-Leistung. Die SDKs bieten eine einfachere API zur Nutzung von Amazon S3 aus einer Anwendung heraus und werden regelmäßig nach bewährten Verfahren aktualisiert. Beispielsweise enthalten die SDKs Logik für die automatische Wiederholung von Anforderungen bei HTTP 503-Fehlern und investieren in Code zur Reaktion auf langsame Verbindungen.

Die SDKs bieten dazu den Transfer Manager, der die horizontale Skalierung von Verbindungen automatisiert, um, wo möglich mithilfe von Bytebereichanforderungen, Tausende von Anfragen pro Sekunde zu erreichen. Es ist wichtig, immer die neueste Version der AWS SDKs zu verwenden, um stets die aktuellsten Leistungsoptimierungsfunktionen nutzen zu können.

Sie können auch die Leistung optimieren, wenn Sie HTTP REST API-Anforderungen verwenden. Bei der Verwendung der REST API sollten Sie ebenfalls die bewährten Verfahren verwenden, die Teil der SDKs sind. Lassen Sie bei langsamen Anforderungen Timeouts und Wiederholungsversuche zu, sowie mehrere Verbindung, um den parallelen Abruf von Objektdaten zu ermöglichen. Informationen zur Verwendung der REST-API finden Sie in der Amazon Simple Storage Service API-Referenz.

Leistungsdesignmuster für Simple Storage Service (Amazon S3)

Beim Entwurf von Anwendungen zum Upload und Abruf von Speicher von Simple Storage Service (Amazon S3) sollten Sie die Designmuster unserer bewährten Methoden verwenden, um eine optimale Leistung für Ihre Anwendung zu erzielen. Wir bieten auch Leistungsrichtlinien an, die Sie bei der Planung Ihrer Anwendungsarchitektur berücksichtigen sollten.

Zur Optimierung der Leistung können Sie die folgenden Designmuster verwenden.

Themen

- Verwendung von Caching für Inhalte mit häufigen Zugriffen
- Timeouts und Wiederholungsversuche für latenzsensitive Anwendungen
- Horizontale Skalierung und Anforderungsparallelisierung für hohen Durchsatz
- Verwendung von Amazon S3 Transfer Acceleration zur Beschleunigung geographisch disparater Datenübertragungen

Verwendung von Caching für Inhalte mit häufigen Zugriffen

Viele Anwendungen, die Daten in Amazon S3 speichern, stellen einen „Arbeitssatz“ der Daten bereit, der von Benutzern häufig angefragt wird. Wenn eine Arbeitslast wiederholte GET-Anfragen für einen gemeinsamen Satz von Objekten sendet, können Sie einen Zwischenspeicher wie Amazon CloudFront, Amazon ElastiCache oder AWS Elemental MediaStore verwenden, um die Leistung zu optimieren. Die erfolgreiche Cache-Nutzung kann zu niedriger Latenz und zu hohen Datenübertragungsraten führen. Anwendungen, die die Zwischenspeicherung verwenden, senden auch weniger direkte Anforderungen an Amazon S3, was zur Senkung der Anfragekosten beitragen kann.

Amazon CloudFront ist ein schnelles CDN (Content Delivery Network), das Daten von Amazon S3 in transparenter Weise an zahlreichen geographisch verteilten PoPs (Points of Presence) zwischenspeichert. Wenn Objekte aus mehreren Regionen oder über das Internet zugänglich sein sollen, ermöglicht CloudFront die Zwischenspeicherung der Daten in der Nähe der Benutzer, die auf die Objekte zugreifen. Dies kann zu hoher Leistung bei der Bereitstellung beliebter Amazon S3-Inhalte führen. Weitere Informationen zu CloudFront finden Sie im Amazon CloudFront-Entwicklerhandbuch.

Amazon ElastiCache ist ein verwalteter In-Memory-Zwischenspeicher Mit ElastiCache können Sie Amazon EC2-Instances bereitstellen, die Objekte im Speicher zwischenspeichern. Dieses Caching führt zur Reduzierung der GET-Latenz im Bereich mehrerer Größenordnungen und zu einer erheblichen Zunahme des Downloaddurchsatzes. Zur Verwendung von ElastiCache modifizieren Sie die Anwendungslogik so, dass der Zwischenspeicher mit aktiven Objekten gefüllt und auf solche Objekte überprüft wird, bevor diese von Amazon S3 angefordert werden. Beispiele für die Verwendung von ElastiCache zur Verbesserung der Leistung von Amazon S3 GET finden Sie im Blogbeitrag Turbocharge Amazon S3 with Amazon ElastiCache for Redis.

AWS Elemental MediaStore ist ein Zwischenspeicherungs- und Content-Verteilungssystem, das speziell für Video-Workflows und die Medienbereitstellung von Amazon S3 entwickelt wurde. MediaStore bietet End-to-End-Speicher-APIs speziell für Video und wird für leistungssensitive Video-Workloads empfohlen. Informationen zu MediaStore finden Sie im AWS Elemental MediaStore-Benutzerhandbuch.

Timeouts und Wiederholungsversuche für latenzsensitive Anwendungen

Es gibt bestimmte Situationen, in denen eine Anwendung eine Antwort von Amazon S3 erhält, die darauf hinweist, dass ein Wiederholungsversuch erforderlich ist. Amazon S3 ordnet Bucket- und Objektnamen den damit verbundenen Objektdaten zu. Wenn eine Anwendung hohe Anforderungsraten generiert (typischerweise dauerhaft über 5.000 Anforderungen pro Sekunde für eine kleine Zahl von Objekten), erhält sie möglicherweise HTTP 503 Slowdown-Antworten. Wenn solche Fehler auftreten, implementiert jedes AWS SDK eine automatische Wiederholungsversuchlogik mit exponentiellem Backoff. Wenn Sie kein AWS SDK verwenden, sollten Sie eine Wiederholungsversuchlogik implementieren, wenn Sie den HTTP-Fehler 503 erhalten. Weitere Informationen finden Sie unter Wiederholversuche bei Fehlern und exponentielles Backoff in AWS in der Allgemeinen Amazon Web Services-Referenz.

Amazon S3 wird automatisch in Reaktion auf andauernde neue Anforderungsraten skaliert und optimiert so die Leistung in dynamischer Weise. Während Amazon S3 Optimierungen für eine neue Anforderungsrate durchführt, erhalten Sie temporär HTTP 503-Anforderungsantworten, bis die Optimierung abgeschlossen ist. Nachdem Amazon S3 die Leistung intern für die neue Anfragerate optimiert hat, werden alle Anfragen generell ohne Wiederholungsversuche bereitgestellt.

Für latenzsensitive Anwendungen empfiehlt Amazon S3 Nachverfolgung und aggressive Wiederholungsversuche bei langsameren Operationen. Wenn Sie eine Anfrage wiederholen,

empfehlen wir die Verwendung einer neuen Verbindung mit Amazon S3 und die Durchführung eines neuen DNS-Lookup-Vorgangs.

Wenn Sie sehr große Anforderungen unterschiedlicher Größe (beispielsweise mit mehr als 128 MB) durchführen, sollten Sie den erreichten Durchsatz nachverfolgen und für die langsamsten 5 Prozent der Anforderungen Wiederholungsversuche durchführen. Wenn Sie kleinere Anforderungen (etwa unter 512 KB) durchführen, bei denen die mittleren Latenzen oft im zweistelligen Millisekundenbereich liegen, ist es sinnvoll, nach zwei Sekunden eine GET- oder PUT-Operation zu wiederholen. Wenn weitere Wiederholungsversuche erforderlich sind, ist ein Backoff die beste Lösung. So empfehlen wir beispielsweise die Ausgabe eines Wiederholungsversuchs nach zwei Sekunden und eines zweiten Versuchs nach weiteren vier Sekunden.

Wenn Ihre Anwendung Anfragen mit fester Größe an Amazon S3 sendet, sollten Sie konsistentere Reaktionszeiten für diese einzelnen Anfragen erwarten. In diesem Fall ist es eine einfache Strategie, das langsamste Prozent der Anforderungen zu identifizieren und diese zu wiederholen. Selbst ein einziger Wiederholungsversuch ist oft erfolgreich für die Reduzierung der Latenz.

Wenn Sie AWS Key Management Service (AWS KMS) für die serverseitige Verschlüsselung verwenden, vgl. Kontingente im AWS Key Management Service-Entwicklerhandbuch für Informationen zu den Anforderungsraten, die für Ihren Anwendungsfall unterstützt werden.

Horizontale Skalierung und Anforderungsparallelisierung für hohen Durchsatz

Amazon S3 ist ein sehr großes verteiltes System. Um diese Größe zu nutzen, sollten Sie parallele Anforderungen horizontal zu den Amazon S3-Serviceendpunkten skalieren. Zusätzlich zur Verteilung der Anforderungen in Amazon S3 hilft dieses Skalierungskonzept dabei, die Last über mehrere Pfade im Netzwerk zu verteilen.

Für Übertragungen mit hohem Durchsatz empfiehlt Amazon S3 die Verwendung von Anwendungen mit mehreren Verbindungen für die parallele Ausführung von GET- und PUT-Aktionen. Dies wird beispielsweise von Amazon S3 Transfer Manager im AWS Java SDK unterstützt, und die meisten anderen AWS SDKs bieten ähnliche Konstrukte. Für manche Anwendungen können Sie parallele Verbindungen dadurch erreichen, dass Sie Anforderungen gleichzeitig in verschiedenen Anwendungsthreads oder in verschiedenen Anwendungsinstances starten. Das beste Konzept hängt von Ihrer Anwendung und der Struktur der Objekte ab, auf die Sie zugreifen.

Sie können die AWS SDKs verwenden, um GET- und PUT-Anforderungen direkt auszugeben, anstatt die Verwaltung von Übertragungen in dem AWS SDK zu nutzen. Dieses Konzept ermöglicht die direktere Abstimmung Ihrer Workloads, bei gleichzeitiger Nutzung der SDK-Unterstützung für Wiederholungsversuche und den Umgang mit eventuell auftretenden HTTP 503-Antworten. Generell gilt: Wenn Sie in einer Region große Objekte von Amazon S3 zu Amazon EC2 herunterladen, sollten Sie gleichzeitige Anfragen für Bytebereiche eines Objekts in der Größe von 8-16 MB starten. Starten Sie eine gleichzeitige Anforderung für jeweils 85-90 MB/s des gewünschten Netzwerkdurchsatzes. Zur Ausnutzung einer Netzwerkschnittstellenkarte (NIC) mit 10 Gb/s können Sie etwa 15 gleichzeitige Anforderungen über separate Verbindungen durchführen. Sie können die gleichzeitigen Anforderungen über mehr Verbindungen hochfahren, um schnellere NICs zu nutzen, etwa 25 oder 100 Gb/s.

Die Messung der Leistung ist wichtig für die Einstellung der Anzahl der gleichzeitig auszugebenden Anforderungen. Wir empfehlen, mit jeweils einer einzigen Anforderung zu beginnen. Messen Sie die erreichte Netzwerkbandbreite und die Nutzung weiterer Ressourcen durch Ihre Anwendung bei der Verarbeitung der Daten. Sie können dann die Engpassressource (die Ressource mit der höchsten Nutzung) identifizieren und so die Anzahl der Anforderungen ermitteln, die wahrscheinlich nützlich waren. Zum Beispiel: Wenn die Verarbeitung einzelner Anforderungen zu einer CPU-Nutzung von 25 Prozent führt, zeigt dies an, dass bis zu vier gleichzeitige Anforderungen möglich sind.

Messungen sind sehr wichtig, und es lohnt sich, die Ressourcennutzung zu ermitteln, wenn die Anforderungsrate erhöht wird.

Wenn ihre Anwendung Anforderungen direkt an Amazon S3 mit der REST API ausgibt, sollten Sie einen Pool von HTTP-Verbindungen verwenden und jede Verbindung für eine Serie von Anforderungen wiederverwenden. Die Vermeidung der Einrichtung von Verbindungen für jede Anforderung macht TCP-Slow-Starts und SSL-Handshakes bei jeder Anforderung überflüssig. Informationen zur Verwendung der REST-API finden Sie in der Einführung zur Amazon-S3-REST-API.

Schließlich ist es auch sinnvoll, auf den DNS zu achten und genau zu prüfen, ob Anfragen über einen großen Pool von Amazon S3-IP-Adressen verteilt werden. DNS-Anforderungen für Amazon S3 durchlaufen eine lange Liste von IP-Endpunkten. Das Caching von Resolvern oder Anwendungscode, der eine einzelne IP-Adresse wiederverwendet, profitiert nicht von der Adressendiversität und dem daraus resultierenden Lastenausgleich. Network Utility-Tools wie das `netstat`-Befehlszeilentool können die IP-Adressen anzeigen, die für die Kommunikation mit Amazon S3 verwendet werden, und wir bieten Anleitungen zu den zu verwendenden DNS-Konfigurationen. Weitere Informationen zu diesen Richtlinien finden Sie unter Anfragen-Routing.

Verwendung von Amazon S3 Transfer Acceleration zur Beschleunigung geographisch disparater Datenübertragungen

Amazon S3 Transfer Acceleration ist effektiv bei der Minimierung oder Beseitigung der durch geographische Entfernung zwischen global verteilten Clients und einer regionalen Anwendung mit Amazon S3 verursachten Latenz. Transfer Acceleration nutzt die weltweit verteilten Edge-Standorte in CloudFront für den Datentransport. Das AWS Edge-Netzwerk verfügt über Präsenzpunkte an mehr als 50 Standorten. Heute wird es für die Verteilung von Inhalten über CloudFront und für die Bereitstellung schneller Antworten auf DNS-Anfragen an Amazon Route 53 verwendet.

Dazu hilft das Edge-Netzwerk bei der Beschleunigung von Datenübertragungen zu und aus Amazon S3. Es ist ideal für Anwendungen, die Daten über oder zwischen Kontinenten übertragen, eine schnelle Internetverbindung nutzen, große Objekte verwenden oder zahlreiche Inhalte hochladen müssen. Sobald die Daten an einem Edge-Standort eingehen, werden sie über einen optimierten Netzwerkpfad an Ihren Amazon S3-Bucket weitergeleitet. Allgemein gilt: Je weiter Sie von einer Amazon S3-Region entfernt sind, um so größer ist die Geschwindigkeitsverbesserung, die Sie von Transfer Acceleration erwarten können.

Sie können Transfer Acceleration auf neuen oder bestehenden Buckets einrichten. Sie können einen separaten Amazon S3 Transfer Acceleration-Endpunkt verwenden, um AWS-Edge-Standorte zu nutzen. Die beste Möglichkeit zur Prüfung, ob Transfer Acceleration die Client-Anfrageleistung erhöht, ist die Verwendung des Amazon S3 Transfer Acceleration Speed Comparison-Tools. Netzwerkkonfigurationen und Bedingungen variieren von zeit zu Zeit und von Standort zu Standort. Sie werden daher nur für Übertragungen belastet, bei denen Amazon S3 Transfer Acceleration Ihre Upload-Leistung potentiell verbessern kann. Informationen zur Verwendung von Transfer Acceleration mit verschiedenen AWS SDKs finden Sie unter Amazon-S3-Transfer-Acceleration-Beispiele.

Mitwirkende

An diesem Dokument haben folgende Personen mitgewirkt:

- Mai-Lan Tomsen Bukovec, VP, Simple Storage Service (Amazon S3)
- Andy Warfield, Senior Principal Engineer, Simple Storage Service (Amazon S3)
- Tim Harris, Principal Engineer, Simple Storage Service (Amazon S3)

Dokumentversionen

Abonnieren Sie den RSS-Feed, um über Aktualisierungen des Whitepapers benachrichtigt zu werden.

Update-Historie-Änderung	Update-Historie-Beschreibung	Update-Historie-Datum
Aktualisiert	Auf technische Genauigkeit geprüft	10. März 2021
Erste Veröffentlichung	Erste Veröffentlichung	1. Juni 2019

Hinweise

Kunden sind eigenverantwortlich für die unabhängige Bewertung der Informationen in diesem Dokument zuständig. Dieses Dokument: (a) dient rein zu Informationszwecken, (b) spiegelt die aktuellen Produktangebote und Verfahren von AWS wider, die sich ohne vorherige Mitteilung ändern können, und (c) impliziert keinerlei Verpflichtungen oder Zusicherungen seitens AWS und dessen Tochtergesellschaften, Lieferanten oder Lizenzgebern. AWS-Produkte oder -Services werden im vorliegenden Zustand und ohne ausdrückliche oder stillschweigende Gewährleistungen, Zusicherungen oder Bedingungen bereitgestellt. Die Verantwortung und Haftung von AWS gegenüber seinen Kunden wird durch AWS-Vereinbarungen geregelt. Dieses Dokument ist weder ganz noch teilweise Teil der Vereinbarungen zwischen AWS und seinen Kunden und ändert diese Vereinbarungen auch nicht.

© 2020, Amazon Web Services, Inc. bzw. Tochtergesellschaften des Unternehmens. Alle Rechte vorbehalten.