Dynamo:Amazon's HIghly available Key-value Store

Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels

2010/11/18

1 概要

大規模での信頼性は、世界最大級の E-コマース事業である Amazon.com で向き合う課題の大きな一つである。ほんのわずかな機能停止でも、多大な経済的な余波を生むし、顧客の信頼に影響を与える。Amazon.com のプラットフォームでは、世界中の多くのウェブサイトでサービスを提供しているが、これらは世界各地に置かれる多くのデータセンターのなかに何万ものサーバーとネットワーク機器の上で実装されている。このような規模では、小さい要素、大きい要素が絶え間なく故障するのは常であり、そのように故障が起きる状況の上で永続的な状態をいかに管理していくかで、ソフトウェアシステムの信頼性とスケーラビリティ性が決定する。

この論文では、Dynamo のデザインと実装について論じる。Dynamo とは、Amazon のコアサービスが常に利用可能な経験を利用するために利用するアベイラビリティの高いキーバリューストアシステムのことである。このレベルのアベイラビリティを達成するために、Dynamo はある故障シナリオでは、一貫性を犠牲にしている。Dynamo は、使用する開発者に新しいインターフェースを提供するやり方で、オブジェクトバージョニングの拡張と、アプリケーションに補助されたコンフリクト解消機構を拡張利用 (多用？) している。

2 序論

Amazon は世界中の E コマースプラットフォームを運営してとおり、そこでは世界中のデータセンターに置かれた何万ものサーバーを使い、ピークタイムには何千万人もの顧客にサービスを提供している。Amazon のプラットフォームには、パフォーマンス、信頼性、効率性、そして透けラービリティの高いプラットフォームのニーズの絶え間ない成長をサポートするために、厳格な業務上の必要条件がある。信頼性は、わずかな機能停止でも重大な経済的な損失、顧客の信頼に影響を与えるので、最も重要な必要条件の一つである。加えて、持続的な成長を支えるためにプラットフォームはスケーラビリティが高い必要がある。
私たちの組織が Amazon のプラットフォームを運営するうえで学んだ教訓の一つにシステムの信頼性、スケーラビリティ性はいかにアプリケーションの状態が管理できるかによってかかってくることである。Amazon は分散された、ゆるく結びつけられた、サービス指向の、数百ものサービスから構成されるアーキテクチャを利用している。このようの環境では、常に利用可能であるストレージ技術へのニーズがある。例えば、ディスク

が故障していたり、ネットワークルーターがフラップ (ON/OFF を繰り返す) した時、データセンターが竜巻で破壊された時でも、顧客は商品を閲覧したり、ショッッピングカートに追加したりできなければならない。それゆえに、ショッピングカートを管理する責任があるサービスには、常にそのデータストアに書き込みや読み込みが可能であること、データは複数のデータセンタにまたがりアクセス可能であることが要求される。

数百もの要素から成り立つインフラの上で故障を扱うことは、我々の業務上の表十モードである。それは、任意の時点で、小さいが、多数のサーバーやネットワーク機器が常に故障しているからである。このような状況なので、Amazon のソフトウェアシステムは、アビイラビリティやパフォーマンスに影響を与えることなく通常のケースとして故障を取り扱う方法で構成される必要があるのである。

信頼性やスケーラビリティのニーズに応えるために、Amazon は多くのストレージ技術を開発した。Amazon Simple storage Service(Amazon の外部から利用可能であり、Amazon S3 として知られている。) はおそらく最も知られているだろう。この論文では、Dynamo のデザインと実装について論じていく。Dynamo は Amazon プラットフォームに構築されたもう一つのアベイラビリティ性の高く、スケーラブルな分散データストアのことである。非常に信頼性が高い必要条件をもち、アベイラビリティ性、一貫性、費用対効果、パフォーマンスの間でのトレードオフを厳格にコントロールする必要があるサービスの状態のを管理するのに Dynamo は利用される。Amazon のプラットフォームは多様で、異なるストレージ要件をもつアプリケーションがある。一部のアプリケーションでは、費用対効果の高くなるように、アベイラビリティを高く、保証されるパフォーマンスのトレードオフをもとにデータストアをアプリケーションデザイナーが設定することを許す程柔軟な、ストレージ技術を必要としている。

Amazon のプラットフォームでは、データーストアに対して、プライマリーキーアクセスしか必要のない多くのサービスがあう。多くのサービスにとって、例えば、ベストセラーリストや、ショッピングカート、顧客の好み、セッション管理、販売ランク、製品カタログなどをを提供するものだが、リレーショナルデータベースをつかう一般的なパターンは、非効率性、費用対効果の低さをもたらす。Dynamo は、これらのアプリケーションの必要条件にたいして、単純なプライマリーキーだけのインターフェースを提供する。

Dynamo はスケーラビリティとアベイラビリティを達成するために、多くの良く知られた技術を総合して利用している。データは、コンシステントハッシュイングを利用して、分割し複製化されており、一貫性はオブジェクトバージョニングによって補助されている。更新時のレプリカ同士の一貫性は、多数決のようなテクニックと、分散レプリカ同期プロトコルによって維持されている。Dynamo はゴシップベースの分散故障感知、メンバーシッププロトコルを採用している。Dynamo は、最低限の管理しか必要のない完全な非集中システムである。ストレージノードの追加や削除は、手動によるデータの分割や再配分を必要とせず行うことができる。

昨年、Dynamo は Amazon の E コマースプラットフォーム上の多くのコアサービスの基盤ストレージ技術であった (となった)。Dynamo は、祝日のショッピングシーズンのあいだ、不稼働期間がなく、きわめて高い負荷に対しても効率的にスケールし応対した。例えば、ショッピングカートを維持するサービス (ショッピングカートサービス) では、数千万ものリクエストに対応し、(すなわち、1 日で、300 万を超える清算を行ったことになる。) セッション状態を管理するサービスは、数十万ものアクティブなセッションを並行して処理を

したのである。

このコミュウニティに対して、主な貢献は、如何に異なるテクニックを組み合わせて一つのアベイラビリティの高いシステムを提供するかということを評価することである。イベンチュアリーコンシステントなストレージシステムが要求の厳しいアプリケーションがあるプロダクションで使うことができることをデモした。また、とても厳格なパフォーマンス要求のあるプロダクションシステムの必要条件を満たすための、これらの技術のチューニングについて洞察を与えた。

この論文は、以下のように構成されている。セクション 2 では背景、セクション 3 ではっ関連作業を論じている。セクション 4 ではシステムデザインを、セクション 5 では実装を記述している。セクション 6 ではプロダクション環境で Dynamo を運用してから得た経験と洞察を、セクション 7 ではこの論文の結論を記述している。この論文の各所で、情報をもっと載せた方が適切だということがあったが、Amazon のビジネスの利益を守るために、詳細部位をある程度切り取った。このため、セクション 6 にあるデータセンター内で、データセンター間での遅延、セクション 6.2 にある絶対的なリクエストレート、セクション 6.3 の故障時間、ワークロードは、絶対的な詳細情報の代わりに、集計された計測から解説 (提供？) している。

3 背景

Amazon の E コマースプラットフォームはレコメンド機能から注文の発行、詐欺検出までの機能が強調し合い数百のセービスから成り立っている。それぞれのサービスは、よく定義された (well-define) なインターフェースを通して、公開されており、ネットワーク越しからのアクセスが可能である。これらのサービスは、世界中の多くのデータセンターに股がり、配置された数万のサーバから構成される基盤の上にホスティングされている。いくつかのサービスは状態を持たないし、(つまり、他のサービスからのレスポンスを集めるサービス。) べつのサービスは状態を持つ、(つまり、永続的なストアで保存された状態の上でビジネスロジックが実行されることによりレスポンスを生み出すサービス)

伝統的に、プロダクションシステムは、リレーショナルデータベースでそれらの状態を保持していた。しかしながら、一般的に使用される状態持続のパターンの多くにとって、リレーショナルデータベースは、決して理想的ではないソリューションである。それらのサービスの多くに取っては、データをストアできて、プライマリーキーをもとに取得出来れば十分であり、RDBMS で提供されるような複雑なクエリーやマネージメント機構は必要ないのである。これらの余分な機能は、高価なハードウェアやオペレーションやスキルの高い個人が必要になり、それらをとても非効率的なソリューションにした。加えて、利用できる複製化技術は限られているし、典型的にアビイラビリティよりも一貫性を選択するものである。最近になり、多くの進化を遂げて入るものの、依然としてデータベースをスケールアウトするのも、負荷分散のためにスキーマを賢く分割するのは容易ではない。

この論文では、これらの重要なクラスのサービスの欲求に焦点を当てた、アベイラビリティ性が非常に高いデータストレージ技術、Dynamo について論じる。Dynamo はシンプルなキー・バリューストアインターフェースを持ち、明確に定義された一貫性のウィンドウを持ち、非常にアベイラビリティ性が高く、リソース使用において効率的であり、データセットサイズやリクエストレートの上昇に対応できるようなスケールアウ

ト可能なシンプルなスキームを兼ね備えている。Dynamo を利用する各サービスは、自身の Dynamo インスタンスを実行する。

3.1 システムの前提条件と要件

この種のサービスのためのストレージシステムは、以下の条件が必要である。

クエリーモデル:
キーによってユニークに特定できるデータアイテムに対して、シンプルな読み書き操作が可能であること。ユニークなキーによって特的できるバイナリーなオブジェクト (つまり、バイナリラージオブジェクト) で状態は保持される。複数のデータアイテムに対してのオペレーションはない。そして、リレーショナルスキーマにたいする要求もない。この条件は、Amazon のサービスの重要な部位が単純なクエリーモデルで動作し、リレーショナルなスキーマを持つ必要がないと言う考察がもとになっている。Dynamo は比較的に小さい (たいてい 1 MB 以下の) オブジェクトを保管するのに必要なアプリケーションに焦点を当てている。

ACID 特性:
ACID(Atomicity, Consistency, Isolation, Durability) 特性は、データベーストランザクションが信頼して行われたことを保証する特徴である。データーベースにおいては、データに対して単一の論理的なオペーレーションをトランザクションという。Amazon での経験では、ACID 保証を提供するデータストアは、アベイラビリティ性が低い傾向があることがわかった。これは、産業界やアカデミアの両方において、広く認められているものである。Dynamo は、もし、アベイラビリティ性の高いのであれば、一貫性の制約を緩めてもいいように動作を行うアプリケーションに焦点を当てている。Dynamo は孤立した保証を提供しないし、単一のキーの更新だけを許可する。

効率性
このシステムは、商用ハードウェアのインフラストラクチャ上で動作する必要がある。Amazon プラットフォームでは、一般的に分布の 99 %を測定の対象とした厳しいレイテンシー制約がある。状態アクセスがサービスオペレーションにおいて、決定的な役割を果たすことを考慮すれば、ストレージシステムは、厳しい SLA を満たすことが出来なければならない。サービスは、それらのレイテンシーやスループットの必要条件を一貫して達成できるように、Dynamo を設定できなければならない。パフォーマンス、コスト効率性、アベイラビリティ、持続性保障において、トレードオフがある。

3.2 サービルレベル契約 (SLA)