遅延数

重要ポイント

- ビデオのタイトルと関連するオンラインコンテンツから、プログラマーが知っておくべき標準的な遅延数値について議論している可能性が高いです。これには、L1 キャッシュアクセス（0.5 ns）やネットワークの往復（最大 150 ms）などの操作の時間が含まれます。これらの数値はハードウェアによって異なります。
- 証拠は、これらの数値が概算であり、特に SSD やネットワークの技術進歩を反映する更新が行われることを示唆しています。

導入

「Latency Numbers Programmer Should Know: Crash Course System Design #1」というビデオは、コンピュータ操作の重要な遅延数値をカバーしている可能性が高く、システム設計にとって不可欠です。これらの数値は、プログラマーにパフォーマンスの影響を理解し、システムを最適化するのに役立ちます。

遅延数値とその重要性

遅延は、メモリへのアクセスやデータの送信など、操作の開始から完了までの遅れです。ビデオでは、以下のような典型的な遅延がリストされている可能性があります：- L1 キャッシュ参照が 0.5 ナノ秒（ns）で、最も速いメモリアクセス。- 同じデータセンター内での往復が 500 マイクロ秒（us）または 0.5 ミリ秒（ms）で、分散システムに影響を与えます。

これらの数値は概算ですが、システム設計の決定に役立ちます。例えば、メモリとディスクストレージの選択などです。

システム設計におけるコンテキスト

これらの遅延を理解することは、コードの最適化、トレードオフの判断、ユーザー体験の向上に役立ちます。例えば、ディスクシークが 10 ms かかることを知っていると、データベース設計を最適化してそのような操作を最小限に抑えることができます。

予期せぬ詳細

興味深い点は、SSD の読み取り時間などの数値が技術の進歩により改善されている一方で、L1 キャッシュアクセスなどのコア CPU 遅延は安定していることです。これは、ハードウェア進化の影響が不均等であることを示しています。

アンケートノート：ビデオからの遅延数値の詳細な分析

このノートは、YouTube で視聴可能な「Latency Numbers Programmer Should Know: Crash Course System Design #1」というビデオで議論されている遅延数値について、オンラインコンテンツと関連リソースを基に包括的な探討を行います。この分析は、プログラマーとシステム設計者のために情報を総合し、これらの数値の重要性について概要と詳細な洞察を提供することを目指しています。

背景とコンテキスト　このビデオは、システム設計に関するシリーズの一部であり、プログラマーにとって重要な遅延数値に焦点を当てています。遅延は、操作の開始から完了までの時間遅れであり、システムパフォーマンスを理解する上で重要です。ビデオのタイトルと関連検索から、Google の Jeff Dean などの人物がプログラミングコミュニティで人気のある標準的な遅延数値をカバーしていることが示唆されています。

オンライン検索により、いくつかのリソースがこれらの数値について議論しており、その中には「Latency Numbers Every Programmer Should Know」という GitHub Gist（GitHub Gist）や 2023 年の Medium 記事（Medium 記事）があります。これらのソースと 2013 年の High Scalability の投稿（High Scalability）を基に、ビデオの内容をまとめるための基礎を提供しました。

遅延数値のまとめ　収集した情報を基に、ビデオで議論されている可能性のある標準的な遅延数値を以下の表にまとめました。各操作の説明も含まれています。

操作	遅延（ns）	遅延（us）	遅延（ms）	説明
L1 キャッシュ参照	0.5	-	-	CPU に最も近い最も速いメモリへのデータアクセス。
分岐ミス予測	5	-	-	CPU が条件分岐を誤って予測した場合のペナルティ。
L2 キャッシュ参照	7	-	-	L1 より大きいが遅いレベル 2 キャッシュへのデータアクセス。
ミューテックスロック/アンロック	25	-	-	マルチスレッドプログラムでミューテックスを取得および解放する時間。
メインメモリ参照	100	-	-	メインランダムアクセスメモリ（RAM）からのデータアクセス。
Zippy を使用して 1K バイト圧縮	10,000	10	-	Zippy アルゴリズムを使用して 1 キロバイトを圧縮する時間。
1Gbps ネットワークを介して 1KB バイト送信	10,000	10	-	1 ギガビット毎秒のネットワークを介して 1 キロバイトを送信する時間。
SSD から 4KB をランダムに読み取り	150,000	150	-	ソリッドステートドライブからの 4 キロバイトのランダム読み取り。
メモリから 1MB を順次読み取り	250,000	250	-	メインメモリからの 1 メガバイトの順次読み取り。

操作	遅延（ns）	遅延（us）	遅延（ms）	説明
同じデータセンター内での往復	500,000	500	0.5	同じデータセンター内のネットワーク往復時間。
SSD から 1MB を順次読み取り	1,000,000	1,000	1	SSD からの 1 メガバイトの順次読み取り。
HDD シーク	10,000,000	10,000	10	ハードディスクドライブが新しい位置にシークする時間。
ディスクから 1MB を順次読み取り	20,000,000	20,000	20	HDD からの 1 メガバイトの順次読み取り。
パケット CA-> オランダ->CA 送信	150,000,000	150,000	150	カリフォルニアからオランダへのネットワークパケットの往復時間。

これらの数値は主に 2012 年のもので、一部の更新が含まれています。特に SSD やネットワークの技術進歩により、ハードウェアの典型的なパフォーマンスが反映されています。

分析と影響　これらの遅延数値は固定されておらず、特定のハードウェアや設定によって異なることがあります。例えば、2020 年の Ivan Pesin のブログ投稿（Pesin Space）では、より高速な SSD（NVMe）やネットワーク（10/100Gb）のおかげでディスクとネットワークの遅延が改善されたことが示されていますが、L1 キャッシュアクセスなどのコア CPU 遅延は安定しています。この不均等な進化は、システム設計におけるコンテキストの重要性を強調しています。

実践において、これらの数値は以下の点を指導します：- **パフォーマンス最適化**：ディスクシーク（10 ms）などの高遅延操作を最小限に抑えることで、アプリケーションの速度を大幅に向上させることができます。例えば、メモリ（250 us で 1MB 読み取り）に頻繁にアクセスするデータをキャッシュすることで、待ち時間を短縮できます。- **トレードオフの決定**：システム設計者は、インメモリキャッシュを使用するかデータベースを使用するかのような選択を迫られることが多いです。メインメモリ参照（100 ns）が L1 キャッシュ参照（0.5 ns）の 200 倍速いことを知っていると、そのような決定に役立ちます。- **ユーザー体験**：ウェブアプリケーションでは、ネットワーク遅延（データセンター内の往復が 500 us）がページ読み込み時間に影響を与え、ユーザー満足度に影響を与えることがあります。2024 年の Vercel ブログ投稿（Vercel ブログ）では、フロントエンド開発においてネットワークウォーターフォールが遅延を重ねることが強調されています。

歴史的背景と更新　これらの数値は、Jeff Dean が 2010 年頃に提唱し、Peter Norvig によって広く知られるようになったものです。研究者の Colin Scott（Interactive Latencies）による更新もあります。2019 年の Dan Hon の Medium 投稿（Dan Hon Medium）では、MacBook Pro の再起動（90 秒）など、より広範な技術関連の遅延がユーモラスながらも関連性のあるものとして追加されています。しかし、コア遅延数値はほとんど変更されておらず、GitHub Gist は 2023 年までに「非常に似ている」と示唆しています。

結論と推奨事項　プログラマーとシステム設計者にとって、これらの遅延数値を覚えることは、パフォーマンス調整のメンタルモデルを提供します。これらの数値はガイドラインとして扱われるべきであり、特定のハードウェアに対する実際のベンチマークが行われるべきです。量子コンピュータや 5G ネットワークなどの新しい技術の更新について最新の情報を得ることが重要です。GitHub Gist や Medium 記事などのリソースは、さらに探求するための出発点を提供します。

この分析は、ビデオの内容を基にし、広範なオンライン調査を補助しています。これにより、コンピュータにおける遅延数値の持続的な関連性が強調され、技術的な変化に適応することがシステム設計の最適化にとって重要であることが示されています。

主要な引用

- Latency Numbers Every Programmer Should Know GitHub Gist
- Latency Numbers Programmer Should Know YouTube Video
- Updated Latency Numbers Medium Article
- More Numbers Every Awesome Programmer Must Know High Scalability
- Latency Numbers Every Web Developer Should Know Vercel Blog
- Latency Numbers Every Engineer Should Know Pesin Space Blog
- More Latency Numbers Every Programmer Should Know Dan Hon Medium
- Numbers Every Programmer Should Know By Year Interactive Latencies