

メタバースサービスにおける高負荷時のレスポンスに関する検討

松原 麻佑[†] 小口 正人[†]

[†] お茶の水女子大学

〒 112-8610 東京都文京区大塚 2-1-1

E-mail: [†]mayu@ogl.is.ocha.ac.jp, ^{††}oguchi@computer.org

あらまし メタバースは、電子データとして構築された仮想 3 次元空間に、インターネットを通じて接続し利用する新しいサービスとして将来的な普及が期待されている。このメタバースサービスの高性能化のために、サーバの負荷とレスポンスという観点から評価を行う。本研究では、オープンソースソフトウェアとして提供されており、ユーザが所持するマシンにインストールしてサービスを提供可能な分散型メタバースサーバである OpenSim システムを用いて評価する。すなわち、メタバース実行時のサーバ側の実測評価に焦点を当てて解析を行う。

キーワード メタバース, OpenSim, Second Life, 分散型サーバ, 集中型サーバ

Evaluation of highly-loaded server response in metaverse service

Mayu MATSUBARA[†] and Masato OGUCHI[†]

[†] Ochanomizu University

2-1-1 Ohtsuka, Bunkyo-ku, Tokyo 112-8610, Japan

E-mail: [†]mayu@ogl.is.ocha.ac.jp, ^{††}oguchi@computer.org

Abstract "Metaverse" which is expected to be spread in the future is a new service connecting to three dimensional space constructed as electronic data through the Internet. We evaluate it from the viewpoint of load and response of the server for making it to be high performance and efficient. OpenSim server that provides the metaverse service is constructed and evaluated. That is to say, we focus on server side of Metaverse system.

Key words Metaverse, OpenSim, Second Life, Distributed server, Centralized server

1. はじめに

近年、一般家庭におけるブロードバンドネットワークの普及や PC の性能向上によって、ユーザ主導のインターネット文化が形成されたことにより、ユーザ同士のネットワークを通じたコミュニケーションや情報提供の実現が強く求められてきている。本研究で注目したメタバースというサービスは、インターネット上に構築された現実世界に似た仮想 3 次元空間を指す。各ユーザがその空間に接続することで、よりリアルに、より容易にコミュニケーションを実現出来る場所を提供するものである。

ブログや SNS が浸透し、ユーザは情報サービスに対して受け身ではなく、発信者であり創作者になっている潮流から、メタバースサービスは大いに注目されている。世界最大規模のメタバースサービスとしては、米 Linden Reserch 社が提供している Second Life が有名であるが [1]、日本でも、(株) ココアが Meet-me というサービスを提供し始めており、多くの利用者を獲得している [2]。一方で、クライアント端末、特にそのグラ

フィクスに要求されるスペックがある程度高いことから^(注1)、期待されている程は普及していないという現実も挙げられるが、それでもなお、このメタバース業界に大手企業が参入しており、少しずつ広がりをみせている。2009 年 3 月の日本における Second Life の月間ユニークリピートログイン数は約 73 万 1 千人を記録しており、2007 年の中盤から Second Life を苦しめていた成長率停滞期が終焉を迎えつつある。また PC の性能向上は日進月歩の状態が続いていることから、端末のスペック不足の問題は今後解消の方向に進むと考えられる。

2. メタバース

2.1 メタバースの概要

メタバースとは、電子データとして構築された 3 次元空間に、インターネットを通じて接続し、その空間内の改変も自由な電子世界である。このサービスの利用者は、この空間内でア

(注1): Second Life クライアントビューアの推奨動作環境は CPU が 1.5GHz 以上、メモリが 1GB 以上、グラフィックカードが Nvidia は 6700 以上、ATI は X800 以上などとなっている

バタと呼ばれる自分の分身を作成する。そのアバタを通して、この空間内で他の利用者とコミュニケーションを持ったり、空間内で有効な通貨を用いた取引や広告表示などのビジネスをすることも可能である。メタバースの類義語としてバーチャルリアリティやサイバースペースが挙げられるが、バーチャルリアリティには五感にフィードバックする機能があり、サイバースペースには Web やゲームなど現実にはない空間全てを指すといった点から区別されている。

2.2 Second Life

Second Life は世界最大規模のメタバースサービスで、その名の通り「第2の人生」を楽しむ場所の提供として位置づけられている。その特徴としては、視覚的に美しいグラフィックスを提供し、スムーズなインタフェースでモノ作りが可能であり、作成したオブジェクトの売買や、Second Life 内で有効な通貨であるリンデンドルを US ドルに換金可能という点が挙げられる。ただし現実世界と大きく異なる点として、この空間内では物理的な制約を一切受けない。アバタは自由に空を飛ぶことができ、一瞬にして世界中の土地にアクセス出来る。Second Life を運営している Linden 社は、この Second Life のサーバソフトを現段階では非公開としているが、一方でクライアントビューアのコードを公開し始めた。

2.3 OpenSim

OpenSim はメタバースサーバのオープンソフトであり、いわば仮想世界を構築するためのプラットフォームである [3]。これは、前述した (株) ココアの Meet-me でも使用されている。Second Life のサーバとは別物であるが、公開されている Second Life のクライアントビューアを参考にして作られたサーバであるため、Second Life のクライアントビューアからこの OpenSim サーバに接続可能であり、Second Life サーバとの互換性が保証されている。またこの OpenSim はオープンソフトであるため、自分が所有しているサーバで構築が可能であり、カスタマイズも容易に出来る。

更に OpenSim は、スタンドアロンモードとグリッドモードの2つの動作モードで構築可能である。スタンドアロンモードでは OpenSim サーバを1つのデスクトップアプリケーションのように起動でき、容易に仮想世界サービスを立ち上げることが出来るものである。OpenSim は Second Life と共通なデータフォーマットを使用しており、現在多くのデザイナーがメタバース内の土地である島を制作する際、OpenSim のスタンドアロンモードで試作品のプレビューを行っている。スタンドアロンモードはその特性を生かし、3D プレゼンテーションツールとしての利用や、病院、遊園地の案内などに使用出来る。しかしスタンドアロンモードは設定が簡単ではあるが拡張性がないため、複数の OpenSim サーバの相互接続や大規模な仮想世界サービスの構築には向いていない。一方グリッドモードは、Second Life のような仮想世界サービスを構築するための動作モードである。グリッドモードでは、離れた場所にある OpenSim サーバ同士を相互接続させ、複数の島からなる巨大仮想空間をシームレスに構築出来る。これにより島がそれぞれ別の OpenSim サーバにあっても、隣の島に移動したり離れた島にレポートをし

たりすることが出来る。図1は、グリッドモードにおけるメタバースサービスのサーバの構成方法を表している。

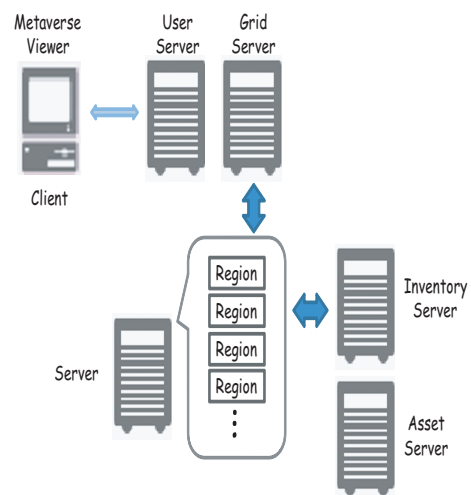


図1 グリッドモードにおけるメタバースサービスのサーバ構成方法

2.4 OpenSim サーバへのアクセス時の処理手順

グリッドモードにおける、サーバアクセス時の処理手順を以下に示す。

1. クライアントがユーザサーバにアクセスし、認証情報を送信する。認証情報には「名前」「パスワード」「アクセスするリージョン」が含まれる。ユーザサーバは認証情報を受信すると、送られてきたユーザ名とパスワードで認証処理を実行する。
2. 認証に成功するとユーザサーバは、グリッドサーバにクライアントがアクセスするリージョンの情報を問い合わせる。
3. グリッドサーバはリージョンの名前からリージョンサーバを検索し、ユーザサーバにリージョンのサーバ IP アドレス、座標などの情報を返す。
4. アクセスするリージョンの IP 情報を取得したユーザサーバは、該当するリージョンサーバに認証した ID とランダムなキー値を送信する。
5. ユーザサーバはインベントリサーバにユーザ ID を送信し、ユーザの持ち物のフォルダ情報の問い合わせる。
6. インベントリサーバはユーザ ID の該当するインベントリ情報をユーザサーバへ返す。
7. ユーザサーバは手順3で取得したリージョン情報、手順4でリージョンサーバに送信したキー値、手順6で取得したインベントリ情報をクライアントに返す。
8. クライアントは手順7で受信したキー値をリージョンサーバのアドレスに送信する。
9. リージョンサーバでは、手順4と手順8で受信した2つのキー値が一致する場合、クライアントのアクセスを受理する。受理された後ユーザのアバタがリージョンサーバ上で初期化され、クライアントとリージョンサーバのインタラクションが開始される。

以上のプロセスで、ユーザアカウントが認証されて、仮想世界にログインできるようになる。

次にアバタの初期化が行われる。

1. リージョンサーバはユーザサーバにユーザ ID を送信し、アバタのデータを要求する。
 2. ユーザサーバはアバタのデータをリージョンサーバに返す。このデータには、アバタの外形を記述するパラメータのほか、アセットサーバで保存されているアバタの服装テクスチャのアセット ID も含まれている。
 3. リージョンサーバは手順 2 で受け取ったアバタのデータをそのままクライアントに返す。
 4. アバタデータを受信したクライアントは外形パラメータに従い、アバタをスクリーンに描画する。そして、服装のテクスチャをアバタに張るために、リージョンサーバに服装テクスチャ ID を送信し、服装などの画像データをリクエストする。
 5. リージョンサーバが手順 4 で受信した服装テクスチャ ID をアセットサーバに送信する。アセットサーバは画像データをリージョンサーバに返す。
 6. リージョンサーバは取得した画像データをクライアントに送信。
 7. クライアントでテクスチャ画像をアバタに張り付ける。
- 以上でアバタがスクリーンに表示され、ユーザのログインが完了する。スタンドアロンの場合は、全てのサーバの処理が、リージョンサーバひとつで動作している。

2.5 既存研究

文献[4]では Second Life のクライアントのビューアの性能評価が行われており、これによると Second Life の実行中、クライアントの負荷は相当高くなっている。一方文献[4]では、サーバ側の解析としてはシュミレーションや数値解析のみが行われている。本研究では、メタバース実行時のサーバ側の実測評価に焦点を当てる。

3. 集中型サーバと分散型サーバ

メタバースはクライアント・サーバ型のシステムである。本節ではその枠組における集中型サーバと分散型サーバについて議論する。

3.1 集中型サーバ

集中型サーバは、サービスを提供する側と受ける側に役割分担されており、多くのクライアントから様々なリクエストを受けて処理するシステムである。そのためサーバサイト数が少なく、有名なサーバには様々な種類のデータが大量に存在する。データは一箇所に集まり、利用者のアクセスも一箇所に集中する。例を図 2 で示す。

3.2 分散型サーバ

分散型サーバは、データがリンクを構成しており、分散したサーバ間で情報が有機的に結びつけられたシステムである。サーバサイト数は多く、大量のデータは分散して存在している。Web サーバはこの分散型サーバの典型例と言える。例を図 3 で示す。

3.3 集中型と分散型の比較

サーバを集中型で構築すると、アプリケーションシステムの

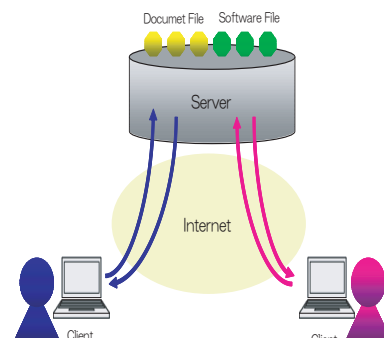


図 2 集中型サーバ

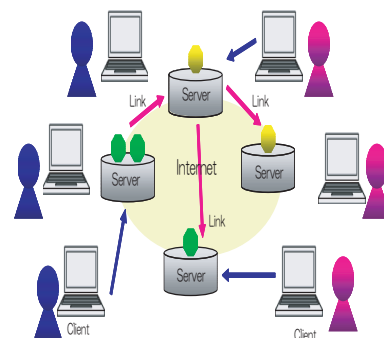


図 3 分散型サーバ

構築時に具体的なハードウェアにひも付ける必要がないことや、運用・管理コストが軽減されるという利点がある。ただし人気のあるデータを持つサーバにアクセスが集中する可能性がある。クラウドコンピューティングの一部はこの考え方を採用している。

一方分散型で構築すると、安全性を高め集中による負荷を避けることや、ハードウェアのコスト削減が可能となる。しかし、人気のあるデータを持つサーバにアクセスが集中してしまうこともある。グリッドコンピューティングの一部はこの考え方を採用している。

4. メタバースサーバの評価

4.1 メタバースサーバの実験環境

図 4 に示すのは OpenSim などメタバースにおける分散型サーバ例であり、サーバ同士はリンクが張られている。OpenSim サーバのどこかにクライアントはアクセスをしており、クライアントが欲しいデータは、あちこちに散らばった OpenSim サーバのどこかに置いてある。本研究では OpenSim を用いこのような分散型サーバを構築して評価を行う予定であるが、本稿ではまず、クライアントが分散型サーバのうち 1 台へアクセスしている部分を構築して評価した。

OpenSim サーバは、CPU が Intel Xeon 3.60GHz、Main Memory は 4GB、OS は Fedora Core9 (Linux2.6.15) で OpenSim0.6.1 を使用している。

一方クライアントは、メモリの容量が違う A、B の、2 つの PC を使用した。computerA は CPU が Intel Pentium 4CPU 3.00GHz、Main Memory は 2.5GB、OS は Windows XP、グラフィックは

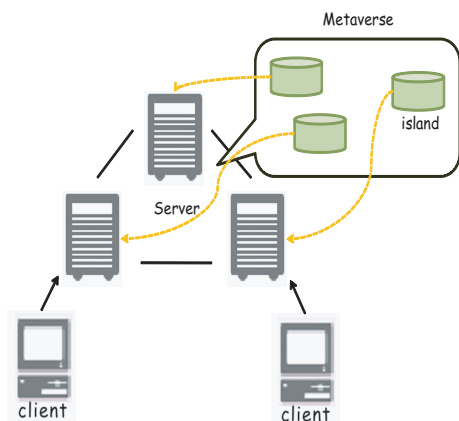


図4 メタバースにおける分散型サーバ例

Intel 82945G Express Chipset Family であり、computerB は Main Memory が 512MB であり、他のスペックは computerA と同じである。

5. 負荷による応答時間への影響

5.1 実験概要

本研究ではスタンドアロンモードとグリッドモードの両方で構築した OpenSim サーバに負荷を与える方法を変えて実験を行い、ユーザのログインにおける時間を測定する。ここでのログイン時間とは、ユーザがログインボタンを押してから、OpenSim が完全に立ち上がるまでの時間を指す。具体的な実験内容は、OpenSim サーバには負荷を与えずクライアントにキャッシュが存在する場合と、同じくサーバに負荷を与えずクライアント側でキャッシュを空にした場合、そしてベンチマークツールを実行し、サーバに負荷を与えている状態の3つの場合についての測定を行う。ベンチマークツールは、PostMark を使用した [5] これは、サーバの性能、特に Web サーバなどのリクエスト処理性能を調べるベンチマークツールである。具体的にはファイルを作成したり、書き込みを繰り返す行うといった負荷をかける処理であり、本研究ではこの PostMark を OpenSim サーバ上で起動させ、バックグラウンド負荷として使用した。

5.2 測定結果

以上の測定を表にしたものを図5、図6に示す。PostMark による負荷は、File Size を 5M～6M に設定し、ログインが終了するまで、PostMark は起動している状態にした。

5.2.1 スタンドアロンモードにおけるログイン時間

computer A において、負荷がなくキャッシュはある場合でのログイン時間は約 16 秒、負荷がなくキャッシュもない場合は約 19 秒、PostMark による負荷をかけた場合は約 29 秒、ログインに時間がかかっている。computer B においては、負荷がなくキャッシュはある場合は約 36 秒、負荷がなくキャッシュもない場合は約 42 秒、PostMark による負荷をかけた場合は約 45 秒であった。

5.2.2 グリッドモードにおけるログイン時間

computer A において、負荷がなくキャッシュはある場合での

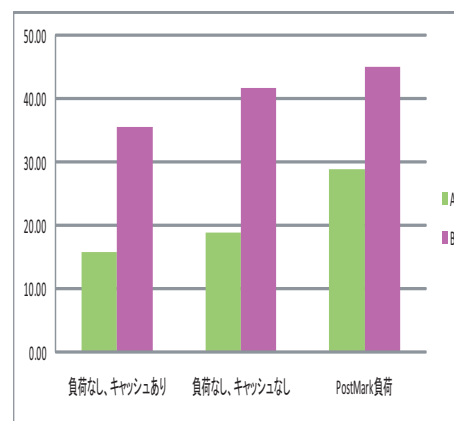


図5 スタンドアロンモードにおけるログイン時間

ログイン時間は約 18 秒、負荷がなくキャッシュもない場合は約 18 秒、PostMark による負荷をかけた場合は約 23 秒、ログインに時間がかかっている。computer B においては、負荷がなくキャッシュはある場合は約 39 秒、負荷がなくキャッシュもない場合は約 45 秒、PostMark による負荷をかけた場合は約 47 秒であった。

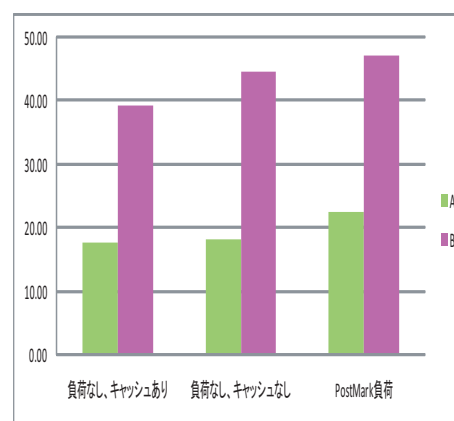


図6 グリッドモードにおけるログイン時間

5.3 測定結果のまとめと考察

サーバをスタンドアロンモードで構築した場合と、グリッドモードで構築した場合において、全体的にグリッドモードの方が時間はかかるものの、それほど大差は見られなかった。しかし、グリッドモードにおいて、ある程度スペックの高いクライアント PC (computer A) を使用した時には、高負荷時におけるログインタイムは抑えられているという特徴があげられる。

負荷に関する考察としては、何も負荷をかけずに、ある程度のスペックを持つクライアント PC を使用しても 16 秒かかり、サーバに負荷をかけた場合は更に影響が出るという結果が得られた。

クライアントのスペックに関しては、メモリの容量により大いに左右されるということがわかった。

以上の実験結果により、負荷がある時はもちろん、負荷がない時でも、トータルのログインまでの時間としては、Web を基準に考えると明らかに長すぎである。メタバースが今後より広

く用いられていくためには、レスポンスの向上は必須であると考えられる。

6. まとめと今後の課題

本稿では、メタバースサービスへユーザのログイン時間を OpenSim を使用し測定し、比較した。その結果から、OpenSim 上でベンチマークツールである PostMark を実行させると大幅な時間ロスになることがわかった。これからサーバ側を改善し、負荷に応じた対処を行うなど、向上を図る余地はある。今後は、このログインレスポンスタイムに関して、どのような処理が、何故遅いのかを細かく解明し、短縮する方法を検討していく。また、クライアントのノード数の増築や、分散型サーバの構築も行い、以上の結果と比較・評価していく。

文 献

- [1] SecondLife, <http://jp.secondlife.com/>
- [2] Meet-me, <http://www.meet-me.jp/>
- [3] OpenSimulator, <http://opensimulator.org/>
- [4] Sanjeer Kumar et.al.: "Second Life and the New Generation of Virtual Worlds", IEEE Computer, vol.41, No.9, pp.46-53 Sep. 2008.
- [5] PostMark, <http://www.asanegra.org/drupal/?q=node/7>