

多数参加型オンラインマルチメディア制作向け PaaS の開発 —プログラム初心者からの手軽な MMO 開発—

1. 背景

現在、多人数参加型オンラインゲーム(以下 MMO)は、開発や、設備にかかる初期コストが大きいため、基本的な仕様が画一化しており、ゲームとして新しい遊ばせ方を提供するようなゲームタイトルが出て来にくい状況になっている。

2. 目的

本プロジェクトでは、MMO の開発および運用を対象に、そこで要求されるシステム上の要件を、スケーラビリティを失うことなく満たすことのできる PaaS(platform as a service)を実装し、サービスの提供を行う。また同時に、このプラットフォームの上での MMO の開発を簡単に行うためのツールを実装する。

3. 開発の内容

・動作環境: linux (glibc 2.3.2, gcc 4.2 以上)

・構成: 以下のソフトウェアからなる。

- ・libnbr.a(C 言語: 約 12000 行): オンラインゲーム向けネットワーク IO ライブラリ
 - ・yues(C++ 言語: 約 12000 行): PaaS の構成要素として、実際にクライアントからのリクエストを処理するワーカーサーバー
 - ・yuem (C++ 言語: 約 11000 行): PaaS の構成要素として、クラスターを構成するノードの管理や全体で共有すべき情報の管理を行うマスターサーバー
 - ・libyuec.a(C++ 言語: 約 10500 行): C++クライアントアプリケーション向けに接続の管理やリクエストの送信機能を提供するライブラリ
 - ・yuec (C++ 言語: 約 10500 行): libyuec.a を内部的に利用して作成された PaaS のパフォーマンステスト用のロボット
- libnbr.a 以外は libnbr.a を通信ライブラリとして利用している。
libnbr.a 以外は約 10000 行のコア部分のソースを共通で利用する。

また、以上の実装のために以下の外部モジュールを援用した。

lua: 分散 KVS に組み込まれるスクリプト言語として使われるオブジェクト指向プログラミング言語である。

tokyocabinet: DBM の実装であり、秒間 100 万件以上の高速な読み書きを実行出来ることで知られている。分散 KVS のデータの永続化に利用される。

mmsgpack: バイナリ形式によってデータを構造化するバイナリシリアライズ形式を規定し、そのシリアライズ、デシリアライズの機能を与えるライブラリである。

SFMT: SIMD を使った高速なメルセンヌ・ツイスターの実装である。

libconhash: consistent hash アルゴリズムの実装を提供するライブラリで、libnbr.a に組み込まれて利用されている。内部的に赤黒木を利用しており、ノード数が増えた場合でも高速にキーに対応するノードを検索することができる。

・機能: 上記のソフトウェアのうち、yues, yuem が、分散 KVS と類似のクラスタを形成し、トータルで以下の機能を提供する。

(1) 大規模オンラインゲームサーバーとして十分な処理性能: リクエストのスループット及び処理遅延において、通常のアクション性のあまり高くないオンラインゲームを提供するのに十分な処理能力をもつ。

(2) 分散透過なプログラミング環境によるプログラムコストの削減能力: この仕組みの上で動作するプログラミング環境において、オンラインゲームの開発で最も難度の高い複数ノードに横断的にアクセスを行うような処理を、分散処理であることを一切意識せずに記述することができる。これによりオンラインゲームのプログラムを作成できる人の母数を増やし、低コストでの開発を可能にする。

(3) PaaS として動作することによるインフラコストの削減機能: これらのソフトウェアのクラスタのみで PaaS として動作することが可能である。これによりインフラストラクチャーを効率よく利用することが可能であり、インフラのコストの削減につながる。

以下、libnbr.a, yues, yuemについて別々に詳細を記載する。

libnbr.a

オンラインゲームで一般的に利用されているリクエスト処理技術を実装したライブラリである。それに加えて、リクエストのマルチスレッドによる処理にも対応しており、さらにオンラインゲームで頻繁に発生する、リクエストを受け取ったコネクション以外のコネクションに対する書き込みを安全かつ効率よく行うことができるように工夫されている。

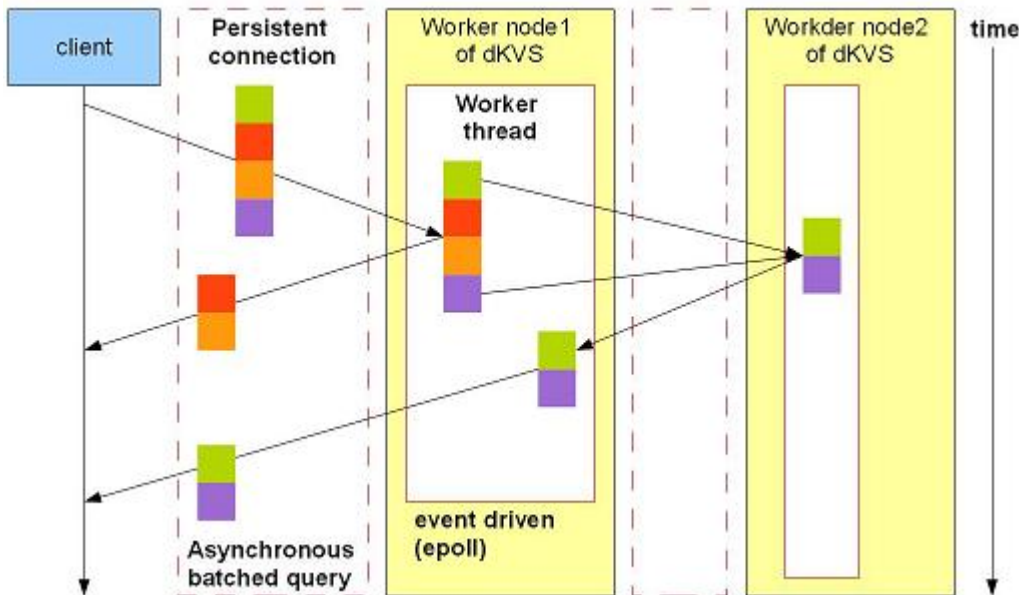


図 1 libnbr.a が提供するリクエスト処理の概要

yues

分散 KVS のワーカースタンプを拡張したような機能を提供する。データを保存するためのバックエンドとロジックを実行するフロントエンドとしての両方の機能を持ち、クライアントからの接続の受付、認証処理のスクリプトによる実行、クライアントからの RPC の処理、実行結果の永続化(レプリケーション含む)、障害発生時のリカバリなどの機能を持つ。

yuem

分散 KVS のマスターノードのような機能を持つ。全てのワーカースタンプと接続しており、各ゲームへのワーカースタンプの割り当てと障害の検出時のノード離脱処理を行う。また、セキュリティのためクライアントからのリクエストのパラメータのチェックをするために、各クライアントと一意に対応するオブジェクトをクライアントの接続ごとに割り当てる必要があるが、その処理も行う。

4. 従来の技術との相違

従来の技術と比較して、以下のような利点があると考えられる。

- ・プログラムコストの削減能力の高さ: オンラインゲームは分散コンピューティングシステム上でプログラムを行うが、このフレームワークを利用することで、通信処理やタイムアウトの検出といった、分散処理システムでのプログラムの難しさを一切考慮せずに、オブジェクト指向言語の逐次処理を記述できる程度の知識でオンラインゲーム向けの分散処理を記述することが可能である。これにより、オンラインゲームの開発に従事できる人材の範囲が広がり、プログラムコストの削減につながりやすい。
- ・インフラコストの削減能力の高さ: クラウド基盤の助けを借りず、単体で PaaS として動作することが可能である。内部のリソース管理はオンラインゲームに特化しているため、クラウド基盤を利用するよりさらに効率的にインフラを利用することが可能であると考えられる。

5. 期待される効果

本システムを利用することで、MMOの開発・立ち上げコストが大幅に軽減され、MMOの開発者がアイデアを具現化しやすい環境が実現し、MMOサービスの多様化を達成されることが考えられる。さらにMMOがPaaS上で提供できるということは、Second Lifeのような仮想世界もPaaS上で提供できるようになるため、サービスの提供者が自由にコントロールできるような仮想世界を、PaaS上に構築して短期間だけ利用することがコスト的に可能になる。現実世界における音楽ライブイベントのような、仮想世界を使った新しいビジネスモデルの創出や、ウェブ広告の一部としての仮想世界の利用を促進することができると考えている。

6. 普及の見通し

今回の開発期間中においては、目的としている機能を提供するためのサーバーアーキテクチャの設計及び実装に注力したが、成果がいままでオンラインゲームの開発にチャレンジした事がないような人たちに対しても広く使われるためには、クライアント側の作成まで含めた統合的な開発環境が必要であると考えられる。したがって、Unity3Dなどの3Dゲームクライアントの開発環境との統合を果たした上で、普及活動を進めていきたいと考えている。

7. クリエータ名(所属)

弥富 豪宏(株式会社コーエー(休職中))