

OSS に基づいた Moodle サイトの スケーラビリティに関する報告

王 躍[†] 小柏 香穂理[†]
刈谷 丈治^{††} 小河原 加久治^{†††}

本研究では、Moodle サイトのスケーラビリティ対策として、オープンソースソフトウェア (OSS) を用いた、スケールアウトでの Moodle サイト負荷分散システムの構築を提案し、その実装実験を通して提案の負荷分散システムの有効性を検証する。

A REPORT FOR MOODLE SITE SCALABILITY BASED ON OSS

Yue WANG[†] Kahori OGASHIWA[†]
Joji KARIYA^{††} and Kakuji OGAWARA^{†††}

As an open source learning management system, Moodle has evolved into one of the most widely-used educational tools in the world. While Moodle can be made to perform very well with a few tens of concurrent users on an all-in-one server [1], it is necessary to scale Moodle site along with performance for more than 100 concurrent users. This report shows how to implement a load-balanced Moodle installation with multiple web servers to achieve horizontal scalability, based on open source software. Validity of the proposed method is confirmed by experimental performance testing.

1. はじめに

近年多くの大学では、オープンソースの e-learning システム Moodle を導入し、Moodle を活用した授業やオンライン試験 (小テスト) を行うケースが増えている。1 台のサーバで構築された Moodle サイトは、サーバの性能によるが、小規模 (同時アクセスユーザ数が数十名程度) の場合が快適に稼働できるが[1], 同時アクセスユーザ数が 100 以上になれば、Moodle サイトのスケーラビリティ対策が不可欠である。

既存の Moodle システムを機能強化してパフォーマンスを向上させるアプローチには、大きく分けて「スケールアップ (scale up)」と「スケールアウト (scale out)」と 2 つある[2]。前者は直接にサーバの処理能力を向上することである。例えば、ハードウェア (CPU, メモリ, HDD 等) の増強, ソフトウェア (OS, PHP, Web サーバ, DBMS 等) のパフォーマンス最適化や Moodle サイト管理設定などを行う[3][4]。一方後者は、接続されたサーバの台数を増やし、負荷をそれぞれのサーバに振り分けて処理能力を向上させることである。さらに、スケールアウト環境においては 1 台のサーバが故障を起こしても他のサーバで処理を継続できるので、システムの可用性 (availability) も向上させることができる。

本研究では、オープンソースソフトウェア (Open Source Software, OSS) を用いた、スケールアウトでの Moodle サイト負荷分散システムの構築を提案し、その実装実験を通して提案の負荷分散システムの有効性を検証する。

2. Web サーバ負荷分散の実現方法

負荷分散を実現する代表的な方式には、DNS ラウンドロビン, L4 ロードバランサー, L7 ロードバランサーがある[5]。DNS ラウンドロビンは、DNS サーバのゾーン設定により容易に実現できる負荷分散方式であるが、サーバ間の負荷分散の割合調整や分散先サーバの障害検知ができないなど問題点が存在する。一方、L4 ロードバランサー方式で利用される汎用のロードバランサー専用機は、通常高価なもので運用に置ける費用的負担がかかる。本研究では、L7 ロードバランサーを実装した、OSS である「リバースプロキシ兼ロードバランサー (Reverse-proxy and Load-balancer)」POUND[6]による Moodle サイトの負荷分散を手軽に実現する方法について述べる。

[†] 山口大学 大学情報機構 メディア基盤センター
Media and Information Technology Center, Yamaguchi University
^{††} 山口大学 名誉教授
Yamaguchi University
^{†††} 山口大学 理工学研究科
Graduate School of Science and Engineering, Yamaguchi University

3. 負分散システムの構成

図 1 にロードバランサーによる負分散システムの基本構成を示す。ロードバランサーは、クライアントからのリクエスト (①) に対し、サーバ群 (server farm) の中から利用可能なサーバを選択し、そのサーバに処理を割り振る (②)。クライアントから見ると、ロードバランサーの背後のサーバとではなく、仮想サーバとだけ通信しているように見えることができるが、本研究では負分散をメインの目的としているので、一旦振り分けられたクライアントとサーバは、その後ロードバランサーを経由せずに処理を行うことになる (③)。そうすることによって、ロードバランサーに余計な負荷をかけずに済むから、ロードバランサーがシステムのボトルネックにはならない。なお、分離されたサーバ (Server 1, ..., Server n) は、(データ共有のため) 同じデータベースに問い合わせを行い、同じ Moodle データファイルエリアを参照する (④)。

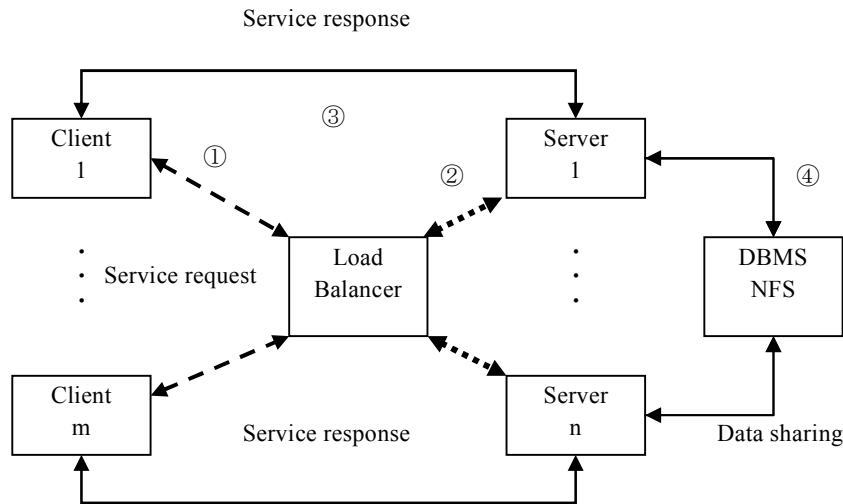


図 1 負分散システムの基本構成

3.1 POUND の設定

POUND[6]は、HTTP リバースプロキシ (Reverse Proxy) 兼ロードバランサーとして、バックエンド・サーバの死活監視を行いながら、接続元 IP アドレスに応じてサーバに割り振ることができる。本研究では、pound-2.5 を使った設定ファイルは次のようになる。

```

## =====<
User      "www"
Group     "www"

Control "/tmp/poundctl"

LogFacility local0
LogLevel 2
TimeOut 30

ListenHTTP
    Address pound.moodle.site.domain
    Port 80

End

Service
    URL ".*"
    Session
        Type IP
        TTL 300
    End
    BackEnd
        Address server1.moodle.site.domain
        Port 8080
        Priority 9
    End
    BackEnd
        Address server2.moodle.site.domain
        Port 8080
        Priority 9
    End
End
## =====>
ここで、2つのバックエンド・サーバ (server1, server2) を指定している。なお、
    
```

パラメータ「Priority」値（1～9）の設定によって負荷分散の割合を調整することができる。（※POUNDの設定についての確認は、コマンド「poundctl -c /tmp/poundctl」で行うことができる。）ただし、（HTTP プロキシ機能上の制約で[6]）POUND とバックエンド・サーバとの通信は HTTP で限定されているが、実際にセキュリティ上、クライアントとサーバの間の通信が Secure HTTP (HTTPS) が望ましい場合は、HTTP リクエストを HTTPS へのリダイレクトを行うようにリアル Web サーバでの対応が必要である。

3.2 HTTPS へのリダイレクト

本研究では、OSS である LIGHTTPD[7]を Web サーバとして利用する。LIGHTTPD の場合は HTTP リクエストを HTTPS へのリダイレクトの設定が次のようになる。

```
## =====<br>server.modules = (<br>    "mod_redirect", ...<br>)<br>}$SERVER["socket"] == "0.0.0.0:8080" {<br>    url.redirect = (<br>        "^/(.*)" => "https://server1.moodle.site.domain/$1"<br>    )<br>}<br>## =====>
```

また、これに合わせて Moodle (1.9) サイトのパラメータ「\$CFG->wwwroot」の設定は次のようにする。

```
$CFG->wwwroot = 'https://server1.moodle.site.domain/moodle';
```

3.3 サーバ間のクロス・ログイン (cross-login) の設定

上述の設定よりサーバの負荷分散を実現できたが、Moodle サイトのデフォルトの設定では、サーバ間の Moodle リソース参照についてまだ1つの問題点が残っている。例えば、クライアント1がリアルサーバ server1 に振り分けられたとする。バックエンド・サーバ間はすべての Moodle データを共有している（図1）ので、server1 の持っている Moodle リソースは、実際に server2 への URL で指定されたものを参照している場合がある。このとき、クライアント1がそのリソースを参照しようとする、（server1 において既にユーザ認証済みにも関わらず）server2 へのログイン認証が必要になるので、クライアント1からのユーザが直接アクセスできない。そこで、負荷分散サーバはすべて同じサブドメインをもっていることを前提として、Moodle クッキーが利用するドメインの設定でこの問題点を解消することができる。具体的に、上述の例に対して、Moodle サイト管理 ([サイト管理] ⇒ [サーバ] ⇒ [セッションハンド

リング]) において「sessioncookiedomain」パラメータを、server1 と server2 の共通のサブドメイン「moodle.site.domain」とすれば良い。この設定より、負荷分散サーバ間の（クッキー認証による）クロス・ログインが実現されるので、認証済みのユーザはすべてのバックエンド・サーバに対して同様にリソースにアクセスすることができる。

4. 実装実験

図2に実装した実験システムの構成を示す。実験システムは、ロードバランサー兼 DBMS・NFS サーバ1台と負荷分散サーバ2台から構成されている。

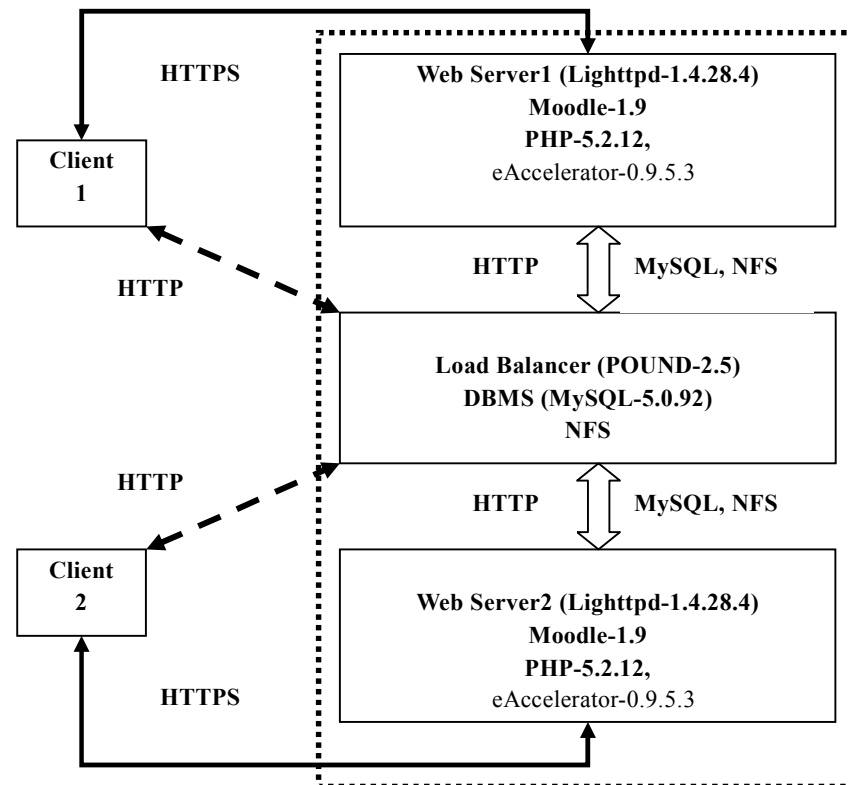


図2 実験システムの構成

4.1 実験環境

4.1.1 ハードウェア環境

実験に使用されているサーバマシン (SUPERMICRO/5015B-NTRB×3) のスペック及び OS は以下の通りである。

- CPU : Core 2 Quad 2.83GHz (4 コア)
- メモリ : 8GB DDR3
- HDD : SATA 2TB×2 (RAID1)
- SSD : SATA 120GB×2 (RAID1)
- RAID Host Adapter : RocketRAID 2210
- OS : FreeBSD 7.3-RELEASE-p2 (amd64)

4.1.2 ソフトウェア環境

実験サーバに使用されているオープンソースソフトウェアは主に以下の通りである。

- Moodle1.9.10+ (Build: 20101208)
- Lighttpd/1.4.28 (ssl)
- Mysql-5.0.92
- PHP 5.2.12 with Suhosin-Patch 0.9.7 (cgi-fcgi) + eAccelerator v0.9.5.3
- Pound-2.5

4.1.3 ネットワーク環境

サーバマシン間は 1000BASE-T の LAN で接続した。クライアント用 HUB はサーバマシンと同じサブネットに 1000BASE-T で接続したが、クライアント PC とクライアント用 HUB は 100BASE-TX の LAN ケーブルで接続した。

4.1.4 ベンチマーク環境

性能評価に用いたベンチマークには OSS である Web アプリのパフォーマンス測定ツール Apache JMeter-2.3.4 を使用した。サーバの負荷テストは、Moodle 小テストシナリオ「ログイン→コース選択→小テスト (全 10 問○×問題)」を用いて行った。JMeter の主なパラメータの設定値は、以下の通りである。Ramp-Up 期間=10 秒、無限ループ (10 分間)、遅延間隔=1 秒、スレッド数=40 または 80 とする (詳細は文献[1]を参照する)。なお、測定は、Moodle サイトのローカルアカウント認証で、1 台のサーバ、または、2 台のサーバに対して、それぞれ、スレッド数が 40 または 80 の負荷を (同時に) かけて行った。

4.2 ベンチマーク結果

実験結果は表 1～4 に示す。また、スレッド=40 と 80 場合の CPU-LOAD-AVERAGE(1 分間)値の変化については図 3～5 に示す。

5. 考察

同時スレッド数が 40 のときのスループットは、1 台の場合に比べると、2 台の場合の方が 82%増になった (表 1, 2)。更に、同じ負荷 (スレッド数=80) に対して、負荷分散サーバ 2 台 (同時スレッド数 40×サーバ 2 台) の場合は、サーバ 1 台 (スレッド数 80) の場合に比べると、CPU-LOAD-AVERAGE(1 分間)と応答時間 90%Line との両方がほぼ半分以下になり、スループットが 19%増になった (表 2, 3)。これらの結果から、同時ユーザ (スレッド) 数が 80 に対して、(2 台のスループットが 1 台より理想の 100%増にはならないが) ほぼ期待通りの負荷分散効果が得られたことがいえる。

表 1. スレッド数 = 40 (× 1 台)

サーバ名	CPU-LOAD-AVERAGE (1 分間)最大値	応答時間 最大値 (秒)	応答時間 90%Line (秒)	スループット (1 秒間)
Server1	2.1	5.1	0.3	33.7

表 2. スレッド数 = 40 (× 2 台)

サーバ名	CPU-LOAD-AVERAGE (1 分間)最大値	応答時間 最大値 (秒)	応答時間 90%Line (秒)	スループット (1 秒間)
Server1	2.4	5.3	0.4	32.6
Server2	3.9	5.4	0.5	28.7
合計				61.3

表 3. スレッド数 = 80 (× 1 台)

サーバ名	CPU-LOAD-AVERAGE (1 分間)最大値	応答時間 最大値 (秒)	応答時間 90%Line (秒)	スループット (1 秒間)
Server1	6.4	5.2	1.0	51.7

表 4. スレッド数 = 80 (× 2 台)

サーバ名	CPU-LOAD-AVERAGE (1 分間)最大値	応答時間 最大値 (秒)	応答時間 90%Line (秒)	スループット (秒)
Server1	5.1	6.0	2.3	38.8
Server2	4.7	5.0	2.0	41.1
合計				79.9

しかし、同時スレッド数が 80 のときのスループットは、1 台の場合に比べると、2 台の場合の方が 55%しか増えなかったため、同時ユーザ数が 160 に対しての (サーバ台数の増加によるスループット増の) 負荷分散効果は、同時ユーザ数が 80 の場合に比べると、27%減少になった。この結果より、本研究の提案した負荷分散システムは、

同時ユーザ数が 160 に十分耐えられると思われるが、ユーザ数の規模が更に大きくなると、負荷分散サーバの台数を増やすだけでは良い効果が得られない可能性がある。

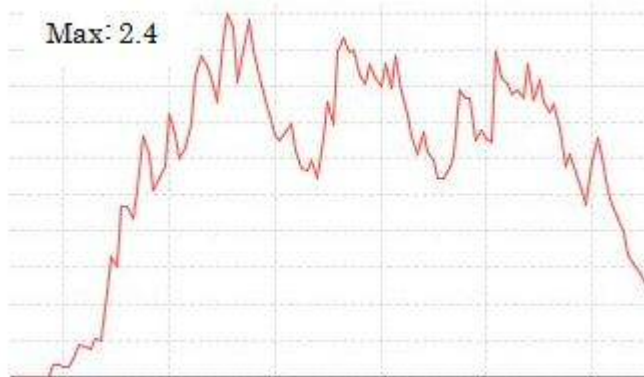


図 3 スレッド数 = 40 (× 2 台) 場合の Server1 の CPU-LOAD-AVERAGE (1 分間)

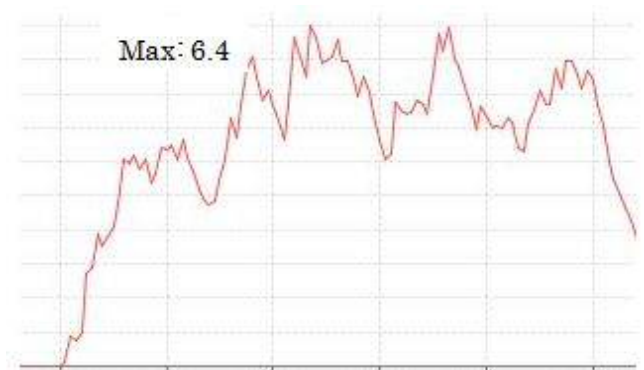


図 4 スレッド数 = 80 (× 1 台) 場合の Server1 の CPU-LOAD-AVERAGE (1 分間)

6. 結論と課題

本研究では、オープンソースソフトウェアを用いた Moodle サイトのスケラビリティ

対策として、負荷分散システムの構築を提案した。また、実装実験を通して提案した負荷分散システムの有効性を検証した。本研究での実装実験の結果より、同時アクセスユーザ数が 160 の大規模オンライン授業に、本研究で提案した Moodle サイトのスケラビリティ対策で対応できることが示唆された。

本研究で提案した Moodle サイト負荷分散システムは、負荷分散サーバが 1 台故障を起こしても他のサーバで処理を継続できるが、「DBMS+NFS」サーバが故障になったら、システム全体は利用不可になってしまう。今後の課題として、このシングルポイント障害 (Single Point Failure) への対策 (例えば、データベースのクラスタ化) などがあげられる。

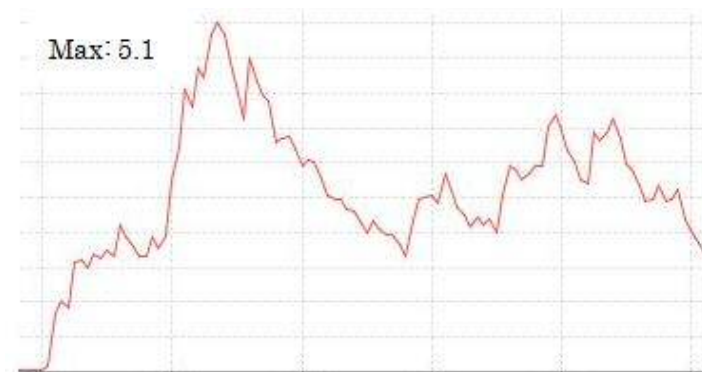


図 5 スレッド数 = 80 (× 2 台) 場合の Server1 の CPU-LOAD-AVERAGE (1 分間)

謝辞 本研究にご協力いただいた山口大学・大学情報機構・メディア基盤センターの皆様へ、謹んで感謝の意を表す。

参考文献

- 1) 王躍, 小柏香穂理, 刈谷丈治, 小河原加久治: Moodle 小テスト時の負荷シミュレーションテスト, 情報処理学会研究報告, Vol.2010-IOT-10 No.11
- 2) <http://lab.klab.org/wiki/サーバ負荷分散概論>
- 3) <http://docs.moodle.org/Performance>
- 4) Alex Buchner: Moodle Administration, pp249-262, PACKT PUBLISHING(2008)
- 5) <http://ossforum.jp/en/node/572>
- 6) <http://www.apsis.ch/pound/>
- 7) <http://www.lighttpd.net/>