

低コストな高可用性と学務システム連携を考慮した Moodle システムの構築

榎田 秀夫[†] 村田 和義[†] 渋谷 雄[†]

[†] 京都工芸繊維大学 情報科学センター
〒 606-8585 京都市左京区松ヶ崎御所海道町
E-mail: †cis-teachers@cis.kit.ac.jp

あらまし 近年、教育機関で e-Learning システムの導入・活用事例が増えてきている。e-Learning システムが本格的に使用されるようになってくると、システムの障害の発生が授業に与える影響が大きな問題となり、可用性の高いシステムが求められる。そこで本稿では、できる限り低運用コストで可用性をあげるように工夫した Moodle システムについて述べる。そのために、(1) 物理的に異なるサーバ PC で構成する、(2) 仮想化技術を用いて、Web フロントエンド部分とバックエンドの DB を別々の OS インスタンス上で動かす、(3) `lsyncd` を用いてファイルの同期化を行う、(4) `pgpool` を用いた DB の同期化、という方法を用いて実現した。さらに、学務情報や認証情報を、学務課やセンター認証システムと連携させることで、教員が e-Learning システムを使い始めるまでの敷居を下げることも実現している。

キーワード Moodle LMS, 仮想計算機 (Virtual Machine), 高可用性, 事務システム連携, 低コスト

The implementation of integrated Moodle pilot system on KIT with low cost and high availability

Hideo MASUDA[†], Kazuyoshi MURATA[†], and Yu SHIBUYA[†]

[†] Center for Information Science, Kyoto Institute of Technology.
Matsugasaki, Sakyo, Kyoto, KYOTO, 606-8585, JAPAN.
E-mail: †cis-teachers@cis.kit.ac.jp

Abstract Recently, e-Learning systems are widely used in many universities. We require that these systems are high availability because a class is suspended if the system has some troubles. In this paper, we present the integrated Moodle pilot system at our university. Our requirements of the new LMS system are Low TCO including hardware and software and High availability. To satisfy the requirements, we choose "Moodle" which run on virtual machines. The key ideas are as follows. (1) Two real server PCs. (2) Four Web servers and Two Database/NFS servers run on different virtual machines. (3) The contents on NFS servers are synchronized by "lsyncd". (4) Two Database servers are synchronized by "pgpool". Moreover, our system has renkei tool with Account database on Information Center and Course informations. These enable that teacher are easy to start using the Moodle system.
Key words Moodle LMS, Virtual Machine, High availability, Low cost.

1. はじめに

近年、教育機関で e-Learning システムの導入・活用事例が増えてきている。本学でも、2003 年に Blackboard 社のシステムを導入し、主に工科系大学教育連携協議会 [1] の中でいくつかの授業科目を e-Learning 用コンテンツ化して提供しているが、積極的に使われているとは言えない状態が続いていた。

その理由としては、以下のような問題があると考えられた。

P1: システムの利用までの手続きが複雑

コース登録や学生登録は、すべて一人の運用管理担当教員に任されている状態となっていたため、使い始めるまでに実施すべき手間が多く、使いたい教員が居ても、タイムリーに対応することが難しい体制であった。さらに、コースが登録されたとしても、学生にどのようにして専用のアカウントを配布する

のかという問題もあった。この為、システムを使おうとするための敷居が高い状態になっていた。

P2: 低性能

導入時は、工科大学教育連携協議会の授業科目を提供することに主眼がおかれていたため、参加人数も多くなく、また非同期型を想定し同時アクセスに対するピーク性能もあまり高く見積もられていなかった。そのため、情報系の演習科目で60人程度が同時にアクセスすると過負荷状態となってしまうものにならない状態となってしまった。

P3: 使い勝手の悪さ

どのようなコンテンツをもって e-Learning とするかは諸説ある。導入時は、講義風景のビデオ (プレゼンツールの表示と、その横で教員が説明している姿) をオンデマンドで流し、別途 PDF ファイルの資料を置く程度を想定していた。しかし、演習科目で課題ファイルを大量にやりとりをしたり、自動採点をする仕組みを動かしたい、といった要望には、当時の Blackboard ではうまく対応できなかった。そのため、専用の Web サイト [2] を動かした方がまし、というような状況となってしまった。

これらの問題を改善し、e-Learning システムを使う講義科目を拡大することを目標に、システムの更新を行った。

2. 要求の整理

前節で挙げた問題を解決する為には、以下に挙げるような要求を満たす必要があると考えられる。

R1: 使いはじめる敷居をさげること

以前のシステムでは、コースの登録や学生登録がすべて手動で行われていた。つまり、使いはじめるためには、管理担当者をお願いをし、学生登録を行い、学生にアカウントの配布を実施する、という段階を経る必要があった。これらの手続きをすることなく、システムにアクセスさえすれば使いはじめられる仕組みを提供することが必須と考えた。

R2: 低コストであること

e-Learning システムを提供し、それが使われることによってどれぐらいの費用対効果が得られるのか、という基本的な情報が十分に存在しない段階において、更新を行う際に多大な費用はかけられない。

R3: 必要な改造・改良が行えること

Blackboard システムは、プロプライエタリなシステムであり、細かい使い勝手に関わる部分の改善要求を出してもすぐに対応できるとは限らず、また対応してもらえなかさえ不明である。他のシステムとの連携などの仕組みを容易に提供するためには、必要な改造・改良を運用担当教員の手で実施できる構成である必要がある。

R4: 可用性は十分であること

e-Learning システムが広く使われるようになると、講義が e-Learning システムに依存することになり、長期間の停止が許されない情勢になることが容易に予想される。運用保守費を十分に確保できるとは限らない為、即日オンサイト保守といった高コストをかけるわけにはいかない。従って、いくつかのコンポーネントが故障していてもサービス自体が継続できる構成で

ある必要がある。

3. 基本設計方針

これらの要求を満たすために、以下のような e-Learning システムの基本設計を行った。

F1: オープンソースソフトウェアで構成する

Moodle [4] と呼ばれる、PHP と DB (PostgreSQL) で構成された LMS システムを採用する。これにより、LAMP/LAPP (Linux Apache MySQL/PostgreSQL PHP) 環境と呼ばれる比較的安価な環境で、システムを構成することができる。また、ソースが公開されているので、細かい使い勝手に関わる部分の改造が、運用担当教員の手でも可能となる。

F2: センターの認証情報と連携させる

本学の情報科学センターでは、全学向けのネットワーク・サーバサービスや教育用 PC 環境などを提供している [3]。その中に、基本認証サーバと呼ばれる LDAP サーバがあり、ほぼすべての教員と学生のアカウントの認証が可能となっている。このサーバと連携させることにより、アカウントの配布作業は情報科学センター側で一括して実施すれば良く、アクセスするための敷居は非常に低くなる。

F3: 学務課の科目コード・履修情報と連携させる

本学の e-Learning システムでは、使いたい講義科目に応じてコースを作成するようになっており、また、利用者は、e-Learning システム上で受講したいと考えるコースに受講登録をして使いはじめる。以前のシステムでは、コース作成は教員からの申し出に従って、運用管理担当教員が手動で作成するため、その作業が終わるまで使いはじめることができない。また、学生は本来の受講登録を学務課に提出しているのにも関わらず、e-Learning システムへの登録作業をしなければならないため、e-Learning システムへの登録を学務課への正式な登録処理と勘違いしたりすることがときどき発生していた。そこで、学務課の持っている開講予定の科目コード情報を元に、それらのすべての科目をあらかじめコース作成しておくこととした。また、学務課に登録された履修登録情報にあわせて、各コースの受講者リストを作成して、それをシステムに反映できるようにした。これにより、教員がシステムを使いたいと考えた際、当該教員のコースは既に準備済みなのですぐに使いつづけることが可能となる。さらに、履修登録情報と同期することで、受講者のみにメッセージを送ったり、クラス名簿の作成を支援することが可能になるため、講義資料を e-Learning システムに載せなくても使える機能を持つことになる。

F4: 仮想計算機技術を用いる

実運用をしながら、必要な改造を実施したりする為には、実運用環境と実験環境を並行稼働させる必要が生じる。また、安全性を高めるために、フロントエンド (apache, PHP) とバックエンド (PostgreSQL) を別々の OS インスタンス上で稼働させることも求められる。そのため、複数のサーバ計算機が必要となるが、物理的に別々の計算機で実現すると、どうしても高コストになってしまう。仮想計算機技術を用いれば、OS インスタンスの作成が比較的容易となり、必要に応じて増減すること

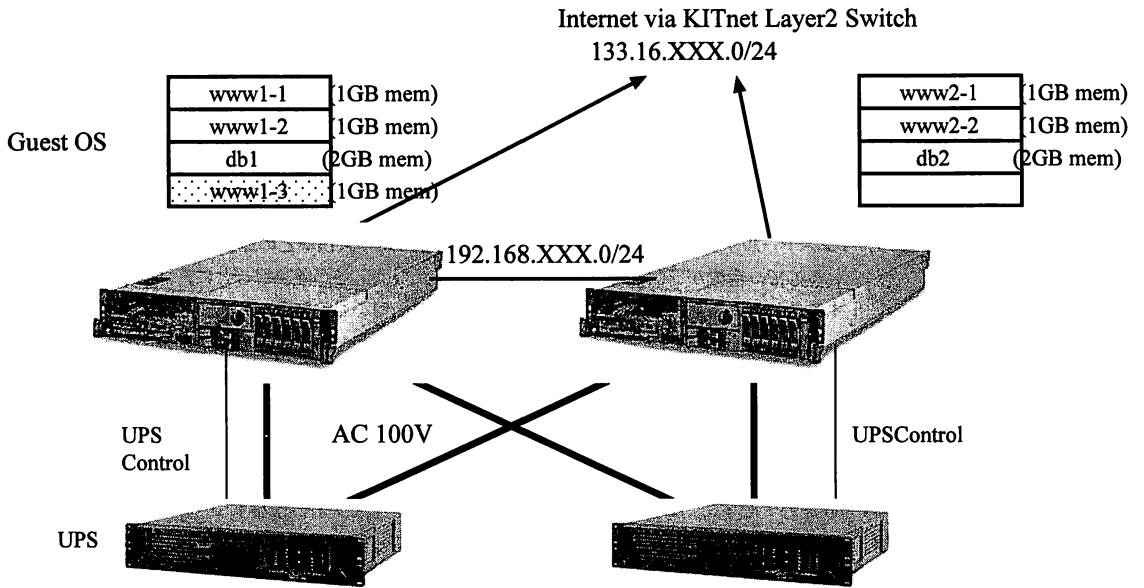


図1 システム構成

表1 導入したシステムのスペック

サーバ PC	IBM x3650 2台 Xeon-D5160 3.0GHz x2, 6GB mem, 300GB SAS x4 (600GB,RAID1+0), 冗長化電源
仮想計算機ソフト	VMware Infrastructure 3 Starter Edition
バックアップ HDD	Buffalo LS-1000 1台
UPS	APC SmartUPS-1500RMJ 2台
KVM IP	AdderLink IP Plathome PShare

が可能となる。さらに、将来性能を増加させる必要が発生した際に、異なるハードウェアを導入したとしても OS 側の設定をほとんど変更することなく追加や移行ができるため、拡張性の面でも優れている。

F5: できるだけ単純な冗長化技術を用いる

あるコンポーネントが故障した場合にもサービスを継続できるようにする為には、コンポーネントの冗長化が有効である。高可用性を実現する為には、クラスタリング技術や負荷分散技術があるが、自動的に failover する機能などは高価であるだけでなく、技術的にもトラブルシュートが難しい、という問題がある。そこで、複数の OS インスタンスの間で、ファイルや DB の中身を同期化する機能のみを使用することとする。そのような技術であれば、同期化機能に問題が発生したとしても同期化機能をバイパスするだけでサービスを継続することが可能になると考えられるため、運用管理者に高度なスキルを求めらなくなる。これは結果として、TCO 削減に寄与すると考えられる。

4. 実装

前節の基本設計方針に従い、システムを実装した。導入したシステムのスペックは表1の通りである。システムの構成概略は図1に示す。

4.1 物理的な構成

電源を二重化したサーバ PC を2台準備し、UPSを含めて対称に構成している。また、HDDはRAID1+0構成とし、HDD障害時にも片肺のHDDだけで中身が参照できる程度に安全な技術を用いている。

それぞれのサーバ PC では、VMware Infrastructure [5] を稼働させている。オープンソースで構築する基本設計に忠実に従うなら、Xen [6] を用いる方がより適しているが、OS稼働という基本的な部分については、サポートの受けられるというメリットを優先している。この仮想化ソフトウェアの上で動作するゲスト OS 上で、Webサーバやデータベースサーバを稼働させている。サーバ PC はデュアルコアの CPU を2基搭載している為、4つのゲスト OS までは CPU 性能が直線的に伸びることが期待できる。

4.2 ゲスト OS の構成

ゲスト OS は、すべて CentOS 5.1 [7] としている。CentOS は、商用の RedHat Linux Enterprise Server の公開されているソースコードをコミュニティの手でリビルドしたものであり、無償で使用できる。構成上、フロントエンドを担う Webサーバ (apache+PHP) の稼働するものと、バックエンドを担うデータベースサーバ (PostgreSQL) の稼働するものがある。また、Moodle にアップロードしたファイルを配置する領域を、すべてのフロントエンドで共通にアクセスできる必要があるため、データベースサーバは、NFS ファイルサーバの機能も持たせて

いる。

Webサーバとデータベースサーバでは、容量の違いとしては基本的にデータベースとファイルサービスの中身(利用者データ)が含まれているかどうかの差しかないと思なせる。従って、利用者データの領域を別ディスクパーティション(/export)とし、OS稼働に関わる部分は、すべてのゲストOSで共通化した。共通部分は25GBとし、利用者データの領域は300GBでスタートしている。これにより、インストールすべきパッケージについては共通化することができるので、構成がシンプルになり、アップデート作業も定型化することが可能となる。

4.3 Webサービスの冗長化

Moodleは、利用者の追跡のためにcookieを使用している。また、Webサーバの多重化への対応の為、cookie情報をデータベース側に持たせてやる方法が提供されている。従って、Webサーバを複数台準備して負荷分散することが容易にできる。本システムでは、高価な負荷分散装置を導入せずに、DNSラウンドロビン機能により確率的に負荷を分散する方法を用いた。この手法は、負荷分散を完全に均質化することはできないが、負荷分散装置というSPF(単一障害点)が不要となる。図1にある通り、本システムでは、Webサーバを1筐体あたり2つ、合計4つの仮想計算機で分散して処理するようにしている。

4.4 利用者データの同期化

データベースサーバは、HDDについてRAID1の冗長化を行っており、単一HDDの障害には無停止で耐えられる構成になっている。しかし、データベースサーバの稼働しているゲストOSがなんらかの理由で停止すると、Webサーバ側が動いていたとしても、結果としてMoodleシステム全体が停止してしまう。本システムでは、1筐体あたり1つ、合計2つの仮想計算機を用意し、一方を通常使用し、他方は常に同期処理をするという構成をとっている。

ファイルサービスについては、lsyncd [8] というツールを使用している。lsyncdは、rsyncとinotifyを使用した仕組みで、あるファイルシステム上のファイルへのアクセスがあった場合に、それをinotifyインタフェースによりリアルタイムに監視し、自動的にrsyncでコピーをキックするものである。単純なファイルのコピーを実行するだけなので、運用にあたって高度なスキルは必要ない。

PostgreSQLについてはpgpool [9] を用いたレプリケーションを採用した。pgpoolは、PostgreSQLに代わってSQL文を受け取り、参照のみのSQLは負荷分散を行い、更新処理を含む場合は、2つのPostgreSQLに処理を依頼し、両方から結果が返ってくることを確認する仕組みである。そのため、もしpgpoolに起因するトラブルが発生した場合でも、pgpoolをバイパスし、直接postgresqlにアクセスするようにするだけで、障害復旧が可能となるため、運用にあたってのスキルレベルがそれほど高くなってもよい。しかし、障害発生時に管理者が手動で対応をする必要がある。

4.5 LDAP認証系との連携

本学の情報科学センターで提供しているLDAP基本認証サーバは、学部、大学院のすべての学生と、教員のアカウントが登

録されている。しかし、LDAPのスキーマが独自形式であるために、Moodle側で単にLDAP認証の設定をするだけでは一部の情報が連携できないことが分かった。そこで、Moodle側のLDAP認証モジュールに手を入れて対応した。具体的には、利用者の種別情報が、businessCategoryというフィールドに入っていることを元にいくつかのフィールドに入れるべき値を計算してから格納するようにした。

4.6 学務課のコース情報との連携

本学の学務課では、各講義科目ごとに、開講曜日・時限、主たる担当教員の情報をもつ時間割マスタと、学生ごとにどの科目を受講しているのかという受講者マスタというデータを管理している。これらのデータは、学務情報システムで管理されているが、このシステムと直接連携することは、作りこみや安全性を考えると避けるべきと考えた。また、これらの情報は、一度作成されれば大きく変更されることはない。そこで、時間割マスタと受講者マスタの情報をCSV形式で本システムの特定のディレクトリにアップロードした上で、コマンドを実行すれば流し込みができるツールとして実現することとした。この程度の作業であれば、学務課の事務職員でも実行可能となるので、運用管理担当教員が直接処理をせずにすむと考えられる。

実際には、学務課の持つ講義に対応した科目コードを、Moodleのコースの短縮名に埋め込むことで一意に識別可能な割当てになるようにしている。

また、時間割マスタには、複数の教員が担当になっている講義であっても一人の教員(主担当教員)だけが登録されている状態になっている。そこで、Moodle側の設定で、コースの教師ロールを持つ利用者は、他の利用者に教師ロールを割り当てることができる権限を与えることとし、共担の講義に対応している。

さらに、受講者マスタは、学生それぞれが、学務課に履修届けを提出し、それを学務課側で集計した後に完成するため、開講初回のコース登録には間に合わない。これに対しては、初回については登録キーを口頭などで伝える方法で学生に自己登録させ、受講者マスタが完成した時点で同期をとることで対応している。実際には、コースごとに受講者マスタにしか存在しない学生については、Moodle側のコースに追加、Moodle側のコースにしか存在しない学生については学生ロールからゲストロールに変更する、という処理を行っている。

4.7 年度更新

e-Learningシステムを使いつづける場合、同一コースを次年度、次々年度とそのまま使いつづけてしまうと、年度の切り替わり時に前年度の受講者の情報を削除しなければならなくなる。しかし、単純に削除してしまえば、前年度の状況との比較ができなくなり情報の蓄積の効果が無くなってしまふ。そこで、本システムでは、同じ科目コードを持つ講義であっても、年度ごとに異なるコースを作成することとした。具体的には、Moodleのコース短縮名に年度を示すコードを追加してコース登録を実施している。

新年度を迎えたときには、新年度用のコースに、旧年度のコースの内容をインポートしてから必要に応じて追加変更をす

表 2 Moodle に追加した機能 (抜粋)

小テスト串刺し表示機能	小テストの評定の際、回答の一覧を一ページで概観する
提出課題の串刺し機能	課題の評定の際、回答の一覧 (オンラインテキストの場合のみ) を一ページで概観する
メ切り課題の採点支援機能	課題の評定の際、メ切りに間に合ったか間に合っていないかで指定した採点をつける
外部システム認証連携	指定された Moodle の内部変数と秘密鍵を接続してハッシュした値を提示する

るようにした。

4.8 外部システムとの認証連携

文献 [2] のように、受領した課題についてプログラム処理をするような仕組みは、Moodle 上でモジュールとして実装しなおす手もあるが、プログラム処理自体の負荷を考慮すると、別の機器で処理をする仕組みと連携する構成の方が望ましい場合がある。Moodle には、外部の Web ページのリンクを呼び出す際に、Moodle 上の内部変数の内容を GET メソッドに付与する仕組みを持っている。しかし、その際には、秘密情報として共通パスワードとアクセス元の IP アドレスの接続に md5 ハッシュをかけたものを使用しているため、proxy 経由のアクセスなどでは別のユーザがハッシュ後のデータを再利用できてしまうため、認証としては非常に弱い。

そこで、Moodle で認証したユーザ名やアクセス元の IP アドレスなど、リソースを配置する側で指定した内部変数のリストを元に接続したものを秘密情報として送り出す仕組みを実装した。これにより、より安全な認証連携が実現できた。

5. 考 察

約半年間の運用状況を踏まえ、本システムについて考察する。

C1: 多数のコースがある場合の Moodle の挙動

本学の 2008 年度の開講講義科目を流し込んだところ、約 1600 のコースが作成された。Moodle は、ログイン直後にコースの一覧を作成するが、その際にデータベース上のコーステーブルを一通りアクセスする。このため、数十コース程度では顕在化しないが、千を超えるコースがある状況では、データベースサーバが過負荷状態に陥ってしまった。これには、データベースの作業メモリの設定を大幅に増加させることで対応できている。

C2: 複数科目コードに跨る講義への対応

講義に対する科目コードに対応させる形で、Moodle 上にコースを作成することにしたが、複数の課程に跨る講義では、課程ごとに科目コードが別々になっている。従って、講義内容が同じコースを複数存在させる必要があるが、コンテンツの更新が大変になってしまう。現在は、これらのコースを子コースとして設定したメタコース機能を用いることで、対応しているが、メタコースの作成は手動対応となってしまっている。学務課と相談して、メタコースに相当する科目コードを付与してもらう、といった方策を検討中である。

C3: レポート機能の弱さ

以前の Blackboard に比べて、受講生が見る環境はそれほど遜

色がないが、教員が参照するレポート機能が弱いという指摘があった。生データはデータベース上にあるため、表示用のモジュールなどの調査や開発が必要と考えられる。

C4: 著作権処理の問題

システムの問題ではないが、e-Learning 化の推進に避けて通れない問題として著作権処理がある。著作権法では、一定の例外的な場合には著作権者に無許諾で著作物を使用できる場合が規定されている。著作権法 35 条では、学校その他教育機関における複製については、一定の条件を満たせば無許諾で複製できること (権利制限) が規定されており、本規定を元に比較的気軽に著作物を利用して講義が行われている。しかしながら、その講義を e-Learning のサーバに蓄積して使用すると、これらの権利制限の範囲外となり、送信可能化権や公衆送信権の問題をクリアにしなければ無許諾で使用することができない。このため、例えば本学の芸術系や意匠系のコースの講義では、実際の作品を元にして議論することが多く、安易に e-Learning コンテンツにすることができない。講義に使用する著作物の権利関係をクリアするためには、相当の知識と手間暇をかけなければならないため、e-Learning 推進の足枷となっている。

6. ま と め

本稿では、できる限り低運用コストで、かつ、ユーザ認証やコース情報を他のシステムと連携して動作する Moodle システムの設計と実装について述べた。

今後の課題としては、データベース部分の負荷分散をはかり、より多くの同時アクセスに対しても快適に動作するようにする方策の検討や、演習系ではない講義へのサポートモジュールの拡充、特に携帯端末への対応などが挙げられる。

謝 辞

本研究の一部は、京都工芸繊維大学 平成 19 年度教育研究推進経費 (教育関係) の「本学における e-ラーニングの全学的環境整備と提供科目の拡大」による。

文 献

- [1] 工科大学教育連携協議会の活動, IT 教育支援協議会ニュースレター, No.1, 3.1 (2003).
- [2] 布目 淳, 福澤 理行, 平田 博章: プログラミング実習におけるコード評価のための e-ラーニングバックエンドシステムの開発, 情報学会 IOT 研究会 2008-05-ICM-11 (2008).
- [3] 京都工芸繊維大学における新情報教育システムについて, CIEC 2006 年 PC カンファレンス, pp.173-176 (2006).
- [4] Moodle, <http://www.moodle.org/>.
- [5] VMware, <http://www.vmware.com/>.
- [6] Xen, <http://www.xensource.com/>.
- [7] CentOS: The Community ENTerprise Operating System, <http://www.centos.org/>.
- [8] lsyncd - Live Syncing Daemon, http://www.pri.univie.ac.at/index.php?c=show&CEWebS_what=Lsyncd.
- [9] pgpool, <http://pgpool.projects.postgresql.org/pgpool-ja.html>.