

ディスクレス PC クラスタを用いたネットワーク システム検証環境の構築

近堂 徹[†] 田島 浩一[†] 岸場 清悟[†] 西村 浩二[†]
相原 玲二[†] 前田 香織^{††}

[†]広島大学情報メディア教育研究センター

^{††}広島市立大学大学院情報科学研究科

あらまし 近年、ネットワーク基盤は我々の社会生活において必要不可欠なものとなり、安定かつ持続的なサービス提供が要求されている。故に、ネットワーク機器やシステムを実環境で運用しようとする場合、目的や要求条件に見合う性能を保持しているかどうかを予め検証できることが望ましい。本稿では、大学等の教育研究機関で整備されている教育用端末や遊休資源を利活用することを念頭におき、PC 端末をディスクレスシステムによりクラスタ化することで多様なアクセスパターンを実現するネットワークシステム検証環境の構築と、それを用いた検証評価事例について紹介する。

Implementation of a performance evaluation system by a diskless PC cluster for network systems

Tohru KONDO[†] Koichi TASHIMA[†] Seigo KISHIBA[†] Kouji NISHIMURA[†]
Reiji AIBARA[†] Kaori MAEDA^{††}

[†]Information Media Center, Hiroshima University

^{††}Graduate School of Information Sciences, Hiroshima City University

Abstract Sustainable operation of the network infrastructure which can provide various services is indispensable factor in our social life. If we use a network equipment or a system in real environment, it is necessary to test the performance previously for stable operation. In this paper, we show a performance evaluation system which can generate various access patterns by a pc cluster, and report some of performance evaluations. We consider a utilization of idle resources by a diskless environment using educational terminals of university.

1 はじめに

ネットワークの広帯域化や PC 端末の低価格化・高性能化に伴い、インターネット上では多種多様なサービスが運用されるようになってきており、我々の日常生活においてネットワークサービスは必要不可欠なものとなっている。さらには、そのサービスを支えるネットワーク基盤技術も安定かつ持続性を求められるようになり、その重要性はますます高まっている。このような観点から、何らかのネットワーク機器やシステムを実際に導入しようとする

場合、目的や要求条件に見合う性能を保持しているかどうかを予め評価し、運用の安定性を検討できることが望ましい。

この検討の対象には、ネットワークスイッチやファイアウォール、負荷分散装置などといったネットワーク機器から、サーバ上で動作する CGI プログラム、Web アプリケーション、といったソフトウェアシステムまで、幅広い範囲で考えられる。しかし、利用するクライアント数やセッション数、帯域などの利用体系の違いによって、その挙動が変わるため、理論的な数

値評価だけでは不十分な場合も多く、実際の運用フェーズに入ってから現象を調査する例も少なくない。そのため、実機を利用した環境での事前検証が必要となる。

実機をベースとしたネットワーク機器やサービスの検証手法として、既に様々な手法が確立され、既に実用化されているものも数多く存在する。ストリームアプリケーションや帯域性能評価を目的としたトラフィックジェネレータとしては **Smartbit[1]**や **Iperf[2]**が存在する。これらは高精度のトラフィックを発生させることで対象機器のスループット計測を行うことが可能だが、特定のトラフィックしか取り扱うことができず適用範囲も限定される。また実インターネット環境を模倣するために、**StarBED** を用いたサーバ負荷試験のフレームワーク[3]が提案されている。これは、700 台規模の高性能サーバ群を利用し、多様なアクセスパターンを再現することでサーバ負荷の検証環境も実現するものである。**StarBED** は検証環境として有効であるが、このような設備を自環境に構築するのは非常に困難な側面もある。

一方、大学や研究機関には計算機資源が多数存在する。教育機関の情報演習室などの共用施設には教育用端末が整備されており、昼間は学生が授業等で利用するが、夜間・休日などの閉室時には停止している場合がほとんどである。

そこで本稿では、大学等の研究機関で整備されている教育用端末や遊休資源を利活用するべく、ディスクレス Linux を用いた PC クラスタを構築することで、容易でかつ様々な用途に利用できるネットワークシステム検証環境を実現するためのシステムについて考える。加えて幾つかの検証事例を示すことで、その有効性について述べる。

2 PC クラスタによるネットワークシステム検証環境

本章では、構築したシステムの概要およびその基本的性能について示す。

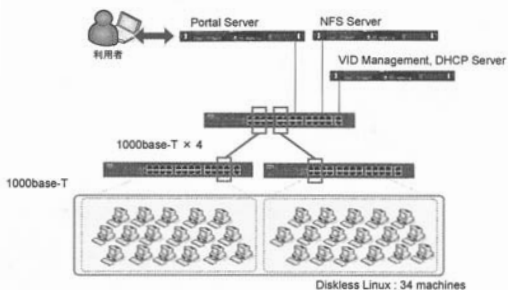


図 1 システム構成図

表 1 使用機器のスペック

	管理サーバ Portal, NFS, VID	実行ノード
CPU	P4-3.06GHz	P4-2.66GHz
Memory	1024Mbyte	
NIC	1000base-T	
OS	Redhat Linux 9	
Kernel	2.4.20-20.9	

2.1 システム構成

本研究で構築したプロトタイプシステムを図 1、使用した機器のスペックを表 1 に示す。本システムはトラフィックを発生させるクライアントノード群（以下、実行ノードとよぶ）と管理サーバ群で構成される。今回の試作システムでは実行ノード 34 台でクラスタを構成した。各端末の接続には 1000base-T を利用し、スイッチ間は 1000base-T を 4 回線利用して 4Gbps にて接続している。これにより、http や ftp のような TCP トラフィックや UDP を利用したストリームトラフィックなどの多様なトラフィックパターンを模倣するのに効果的な構成となっている。

各実行ノードは管理コストを考慮しディスクレス Linux で構成している。ノードのブート時に共通のカーネルイメージを取得して起動し、root ディレクトリ、ホームディレクトリやアプリケーションディレクトリは NFS サーバをマウントするように設定している。これにより、各実行ノードの設定情報や処理結果は NFS サーバに集約され、一元的に管理できる。

なお、今回はディスクレスブート環境として、ミントウェブ社の VID システムを用いているが、Debian/GNULinux3.1r4 をベースに PXE ブートによる Linux ディスクレス環境でも同様の機能を有するシステムが実現できることを確認している。

次にミドルウェアについて述べる。本システムでは、管理運用の効率化を図るために Score を利用した。実際のノード間通信には rsh(Remote-shell)を利用して行っており、ポータルサーバから一元的に各クライアントの挙動を制御することができ、さらに台数の増加に対するスケーラビリティも容易に確保することができる。また、このようなクラスタリングツールを利用せず、リモートシェルを同時実行させるためのラッパープログラムである dsh(distributed-shell)を用いても同様の機能を提供できることを確認している。

このように、本システムの利点は各実行ノードがディスクレス環境で動作する点にある。これによって、端末のハードディスク領域を変更することなく活用することができるため、大学等に存在する遊休資源を用いることでも同様のシステムを構築することが可能になる。

2.2 実行処理概要

次に実際の処理概要について述べる。本システムは、用途に応じて「プログラマブル設定」と「簡易設定」の2種類の設定方法を用意している。以下それぞれについて説明する。

「プログラマブル設定」は、各実行ノードで行わせるべき処理をユーザ自身がスクリプト形式で記述し、それを指定したホストに対して管理サーバから呼び出す方式である。実際の実行手順について、あるWebサーバに対してSSL暗号化を用いたhttps同時接続を実現する場合を図2に示す。

まず始めにユーザは、各実行ノードがマウントしているディレクトリに実行ノードが処理する内容を記述したスクリプトファイルを保

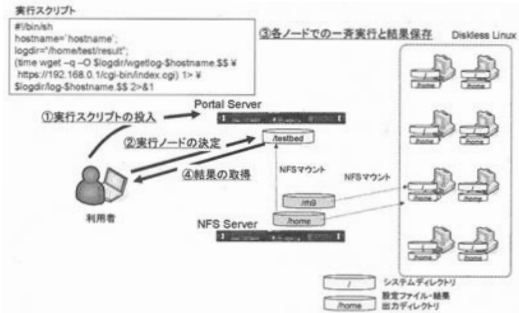


図2 プログラマブル設定における実行手順

存する。スクリプトファイルは、通常のUNIXコマンドをシェルスクリプトにて記述する。

次に、各実行ノードの挙動をポータルサーバ上で制御する。ポータルサーバ上では、スクリプトファイルで記述した処理をどのノードで起動させるかを指定する。この方法として、rsh-allコマンドにより対象とする実行ノードを直接指定して実行させる方法のほか、引数で与えるノード数から自動的に実行ノードリストを生成しrsh-allコマンドを実行することができる。

このように、実行ノード処理の記述とポータルサーバ上での操作の2ステップの操作を経て、本システムを利用した評価実験を行うことができる。コマンドは通常のUNIXコマンドをそのまま利用することができるため、UNIX経験者であれば操作性を損なうことなく本システムを利用することができる。また、perlやexpectなどの言語、自作プログラムを利用すればより高度な処理も行うことが可能となる。

もうひとつのモードである「簡易設定」は、図3に示す設定でプログラムを起動することで、各ノードに自動的にプロセスを振り分けで同時実行してくれるものである。トラフィックの生成パターンは限られるが、パラメータを入力するのみで一斉アクセスが実現できるため、各ノード共通の設定の場合はこちらが容易に利用できる。

```

Stream 1 {
  NODE="THE NUMBER OF NODES"      : stream name
  SESSION="THE NUMBER OF SESSIONS" : the number of nodes
  TRAFFIC="UDP/TCP/ICMP/HTTP"     : the number of sessions in each node
  BANDWIDTH="Mbps"                : traffic type
  TIME="SECOND"                   : if you use UDP, you may specify the bandwidth (default is 1Mbps)
  DESTINATION="Address or URL"    : if you use UDP/TCP/ICMP
  LOG=" LOGFILE-NAME"             : if you use UDP/TCP/ICMP
}

```

図3 簡易設定モードにおけるコンフィグ項目

なお予備実験にて、各ノードから特定ホストへ一斉アクセスを行う場合のアクセス時間精度を測定した結果、340セッション（各ホスト10セッション×34台）の場合でも1秒以下で全アクセスが生成可能であることが分かっている。

3 評価実験事例

本章では、本評価システムを用いて行った評価実験事例についていくつか紹介する。

3.1 ネットワーク利用者認証装置に関する評価事例

耐セッション性能に関する評価事例として、ゲートウェイ型のネットワーク利用者認証装置（以下、認証ゲートウェイ）の評価事例を取り上げる。認証ゲートウェイは、公衆ネットワークにおける持込端末などを対象とし、ネットワーク内部から外部へのアクセスに対する利用者認証機能を提供する機器である。利用開始にあたり利用者の認証処理が行われるが、大学などの演習室での利用や大規模な会議時などには一斉認証処理による処理負荷の増大が考えられるため、本評価では同時認証接続時の処理性能についての検証を行った[4]。

実験環境を図4に示し、それぞれの仕様を表2に示す。本稿では、評価対象機器としてネットスプリング社が販売するMicroFEREC（ファームウェアバージョン 1.5）[5]を用いる。認証はWebブラウザによる認証方式を採用、認証データについては、アカウント情報を学内に設置してあるLDAPサーバに100名分登録した。認証装置からはRADIUSプロトコルでRADIUSサー

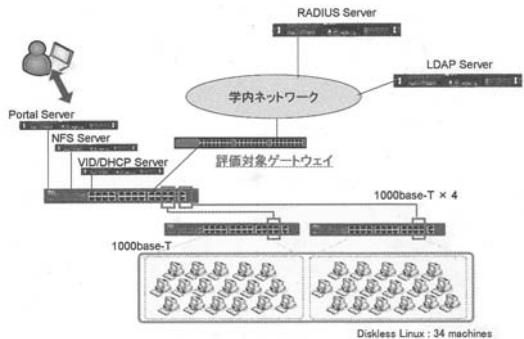


図4 利用者認証装置の評価環境

表2 評価実験における使用機器

サーバ	ハードスเปック / OS / ソフトウェア
LDAPサーバ	Ultra Sparc IIIi 750MHz 512MB / Solaris8 / iPlanet (LDAP)
RADIUSサーバ	Ultra Sparc Ili 440MHz 512MB / Solaris 10 / FreeRADIUS, OpenLDAP

バに認証問い合わせを行い、LDAP内の登録情報との照合により認証処理が行われる。

同時生成セッション数に対する認証可能セッション数と認証時間の結果を図5、図6に示す。横軸は生成セッション数を表し、縦軸は成功セッション数（図5）、認証成功した全セッションの最小/平均/最大認証時間（図6）を表す。図5の結果より、SSL通信の有無に関わらずセッション数の増加に従い徐々に性能が低下しているのが分かる。

このことから、SSL暗号化処理以外の要因でボトルネックが生じていることが分かる。また、図6の結果より、同時認証セッション数の増加

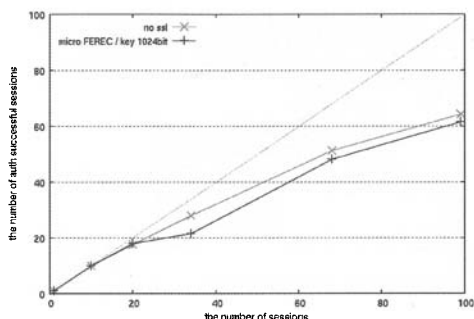


図5 認証可能セッション数

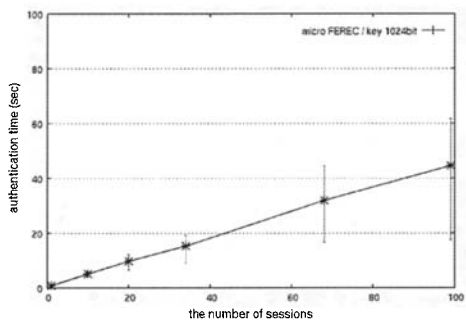


図6 https(鍵長 1024bit)における認証時間

によって認証時間が2倍程度に増加していることが分かり、本環境では100セッションのhttps(1024bit)の同時接続で最大60秒程度、認証に時間を要していることが分かった。

これらの結果より、機器の性能を定量的に把握することが可能となり、設置の要求条件とその性能から、必要台数や設定パラメータを検討することが可能となった。

3.2 ネットワークエミュレータの性能評価事例

もう1つの評価事例として、ネットワークエミュレータの性能評価について述べる。ネットワークエミュレータはIPネットワーク上で発生するパケット損や遅延等の障害を再現することを可能にし、インターネット上で動作するアプリケーションの振舞いを確認するために効果的なシステムである。故に、それらのア

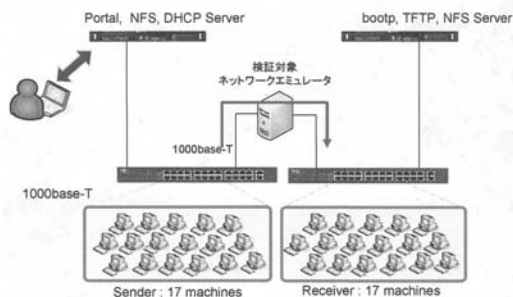


図7 ネットワークエミュレータの性能評価環境

プリケーション(ストリーム)に対してエミュレーション動作が適切に行われるか検証するために、実際に多種多様なストリームを生成し評価するための仕組みが必要となる。

システム構成図を図7に示す。ディスクレスLinuxを送信側ノードと受信側ノードに分割し、それぞれのノードを1対1で対応付けるようにしている。管理サーバからの制御命令に応じて、lperf を利用し、ノード毎に異なる帯域/パケットサイズ/トランスポートプロトコルなどの特性を持つストリームを同時に送出することが可能である。無論、これらの構成は、検証要求に応じて変更することが可能である。今回は評価対象として、ソフトウェアで動作するネットワークエミュレータのひとつであるLenet[6]を利用した。

本稿では、エミュレータ機能のひとつであるパケット損失生成機能についての評価を示す。Lenet の設定パラメータをパケット損失率1%に固定した状態で表3に示すような特性の異なる広帯域ストリームと低帯域ストリームを混在して伝送させた場合のストリーム間でのエミュレーション性能について評価する。なおエミュレータノードとしてはPentium4 -2.6GHz, 512MByteのLinuxPCを利用し、転送パケット数は各ストリーム200,000パケットとした。

表4にパケット損失に関する性能を示す。結果は、各ストリームの平均値を代表値としてい

る。また、分散は同一特性のストリームにおいて算出したものである。結果から、平均パケット損失の生成性能については、設定値から増減はあるものの分散は小さいことから、同一特性ストリーム内での損失率のばらつきは小さいと考えることができる。即ち、各ストリームに対して適切にパケット損失がエミュレートできているといえる。またストリームの増加とパケット損失率の再現性は相関関係がなく、転送ストリームが増加に対してもエミュレーション機能が動作していることが確認できる。

このように、利用するエミュレータの性能について前もって検証することができる点において、この検証環境は効果的に利用できるといえる。本システムを用いた検証実験としては、上記に示した他に、セッション管理サーバの耐性評価やネットワーク認証スイッチの性能評価なども実施しており、様々なシステム検証に利用できることを確認している。

表 3 生成ストリームの特性

ストリームタイプ	帯域 (bps)	パケットサイズ (Byte)
St1	UDP 10M	1200
St2	UDP 128k	180

表 4 パケット損失のエミュレート結果

送信ストリーム数	内訳	平均パケット損失率 (%)	分散 (% ²)
1	St1 x 1	1.0015	—
2	St1 x 1	1.0079	—
	St2 x 1	1.0010	—
4	St1 x 2	0.9910	2.50×10^{-12}
	St2 x 2	1.0047	0

4 まとめ

本稿では、PC 端末をディスクレスシステムによりクラスタ化することで、多様なアクセスパターンを実現するネットワークシステム検

証環境の構築と、それを用いた検証評価事例について述べた。本システムの利点のひとつは各実行ノードがディスクレス環境で動作する点にある。これによって、端末のハードディスク領域を変更することなく活用することができるため、大学等に存在する遊休資源を用いることで同様のシステムを構築することが可能になると考えられる。また、利用方法として簡易設定とプログラマブルな設定を利用可能とし、あらゆるトラフィックパターンを再現できることが可能となる。

今後の課題としては、GUI アプリケーションなどのより高度な処理への対応や、仮想化技術を利用した、行ノード数拡大への対応などを行っていく必要がある。

謝辞

本研究に際し、評価実験事例にご協力頂いた広島市立大学大学院情報科学研究科井上博之講師、広島市立大学情報科学部情報メディア工学科日山雅之君に感謝致します。本研究の一部は文部科学省科学研究費（基盤研究(B) 19300019)の支援を受けて実施しています。ここに記して感謝の意を表します。

参考文献

- [1] Spirent Communications, Smartbit
<http://www.spirent.com>
- [2] A.Tirumala, F.Qin, J.Dugan, J.Ferguson, and L. Gibbs, "Iperf: The TCP/UDP bandwidth measurement tool", <http://dast.nlanr.net/Projects/Iperf/1999>.
- [3] 野中 雄太, 知念 賢一, 宇多 仁, 宮地 利幸, 篠田 陽一, "StarBED を用いたサーバ負荷試験の実現", DICO2007 シンポジウム論文集, pp.199—204, 2007
- [4] 近堂 徹, 田島 浩一, 岸場 清吾, 西村 浩二, 相原 玲二, "PC クラスタによる認証スイッチの認証性能評価システム", 情報研究報告 2007-DSM-47(5), pp.25-30, 2007 年 9 月.
- [5] (株)ネットスプリング, MicroFEREC,
<http://www.ferec.jp>
- [6] Kaori Maeda, Masahide Ishino, Eitaro Kohno, "A Userland Network Emulator with Replay and Capture", The 2007 Internet Symposium on Applications and the Internet (SAINT2007), CD (8pages), 2007