規模に応じたインターネット サーバー構築・運用ノウハウ

民田雅人
松井証券株式会社

このチュートリアル

- 大規模なマシン
  - 例えば、ISPのNOCで運用するサーバー
- 小規模なマシン
  - 例えば、OCNエコノミーで運用するサーバー
- 適正なシステムを適正なコストで運用する判断が行えればよい
- AT互換機とSunのWSを例に

規模とコスト

- 小規模なシステム
  - システムが安上がり
  - 手間もあまりかからない
- 大規模なシステム
  - お金がかかる
  - 運用にさまざまなノウハウが必要になる
- ラージシステムをスモールコストで運用できれば理想的

速いことは良いことだ

- CPUが速い
  - 計算処理が短時間
- HDが速い
  - データの読み書きが短時間
  - アプリケーションの起動が短時間
- ネットワークが速い(太い)
  - データ転送が短時間

スピードとコスト

- イニシャルコスト
  - 概して、速いものは高く、遅いものは安い
- ランニングコスト
  - 速いシステム
    - 手間がかからないので人件費が安くなる(かも?)
  - 遅いシステム
    - 手間がかかるので人件費はかえって高くなる(?)

インタネットサーバーの例

- WWWサーバー
- ネームサーバー
- メールサーバー
- NEWSサーバー
- WEBキャッシュサーバー
- IRCサーバー

システム構成

システム構成のポイント

- CPUの種類とスピード
  - PC-ATを例にすると
    486マシンからPentium-II Xeonまで色々
- メモリ容量
  - 16M程度から512M、あるいはそれ以上
- ハードディスク
  - どの容量のディスクを何台？
- Network I/F
  - 10Mbps or 100Mbps

CPU選択のポイント

- アーキテクチャ
  - x86, SPARC, PowerPC, Alpha….
- クロック
- キャッシュ
- メモリ

CPUアーキテクチャ

x86	PC-AT互換機
SPARC	Sunとその互換機
PowerPC	Macintosh, IBM?
MIPS	SGI
Alpha	COMPAQ
PA-RISC	HP, HITACHI

AT互換機のCPU

- i486 25〜100MHz
- Pentium 75〜200MHz
- Pentium MMX 166〜266MHz
- Pentium-Pro 150〜200MHz
- Pentium-II 233〜450MHz
- Pentium-II Xeon 400〜450MHz
  - SPECint95で0.3〜15

SPARC

- MicroSPARC(LX, Classic) 50MHz
- MicroSPARC-II(SS5)70〜110MHz
- TurboSPARC (SS5)　　170MHz
- UltraSPARC (Ultra1)　　167MHz
- UltraSPARC 1+(Ultra2)　200MHz
- UltraSPARC II (Ultra2, EP450)
  250〜333MHz

CPU Clock Speed

- 最も単純な指標
- クロックスピードと処理能力は密接な関係
- 100MHzのCPU vs 200MHzのCPU
  - 倍速くなるとは限らない
  - 倍以上に速くなることもある

CPUの能力の違い

- FreeBSDのカーネルコンパイル
  トータル時間での比較
  - Pentium-II/333　　2分程度
  - PentiumPro/150　　4〜5分程度
  - Pentium/120　　10分程度
  - 486DX2/66　　20〜40分程度
    - 注: FreeBSDのバージョンやシステム構成が違うため単純に比較できない。

Cache Memory(1)

- CPUとメモリのバス速度の差を吸収する
- 1次キャッシュ
  - CPUに内蔵されている
  - 4K 〜 64Kbyte
- 2次キャッシュ
  - 多くはCPUの外部
  - 128〜2Mbyte

Cache Memory(2)

- サイズ
  - キャッシュメモリそのものの量
- キャッシュバスのスピード
  - CPUクロックとアクセススピードの関係？
- キャッシュ可能エリア
  - 主記憶サイズとの兼ね合い

キャッシュサイズ(x86)

- Pentium-Pro
  - 1次8k/8k 2次256k〜1M
- Pentium-II
  - 〜 300MHz　1次 16k/16k　2次 512k
  - 300MHz超　1次 16k/16k　2次 1M
- AMD K6
  - 1次 32k/32k　2次はMBの構成による

キャッシュサイズ(SPARC)

- UltraSPARC-IIi (Ultra5/Ultra10等)
  - 1次　16K/16K
  - 2次　270MHz　256K
    300MHz　512K
    333MHz　2M
- UltraSPARC-II
  - 1次　16k/16k
  - 2次　250MHz 1MB　300MHz 2MB

*Pentium-II*は速いか？

- Pentium-II 233〜333MHz(66MHz)
  - Cache Clock = 1/2 CPU Clock,
    Area ≦ 512KB
- Pentium-Pro 〜200MHz
  - Cache Clock = CPU Clock, Area ≦4GB
- **Cacheにヒットする限り** P6-200**は速い**
  - **計算なら**Pentium-II
  - **大規模なデータを扱う場合**Pentium-Pro

メモリバス

- CPUとメモリのインターフェース
- メモリバスのクロックとバス幅で決まる

バス幅	クロック	転送量
32bit	33MHz	132MByte/sec
64bit	66MHz	528MByte/sec
64bit	100MHz	800MByte/sec
128bit	100MHz	1600MByte/sec

実際のメモリの挙動

- 計算通りの転送レートは得られない
  - 最初のアクセスに時間がかかる
  - 1クロックで転送できるとは限らない
  - 4-2-2-2　　最初の1ワードに4クロック
残りの3ワードは6クロック
  - 5-1-1-1　　最初の1ワードに5クロック
残りの3ワードは3クロック

メモリ容量

- 必要にして十分なメモリを用意する
  - 少ないとパフォーマンスに影響する
  - 多すぎるメモリは単なる無駄
    - WS用のメモリは安いとはいいがたい
- 次のことを頭にいれる
  - 稼動するプロセスの使うメモリ
  - OSの利用するメモリ
  - ディスクバッファ

ハードディスク

- 磁気円盤が高速回転し
  磁気ヘッドで読み書きを行う
- 連続したアクセスは速い
- ランダムなアクセスはヘッドが動くため遅い
  - インターネットサーバーのボトルネックの要因
- サーバー用にはSCSIが一般的

実際のHDの例

- IBM DDRS シリーズのカタログより
  - DDRS-34560 and DDRS-39130
- スペック
  - 109-171 Mbits/sec Media data rate
  - Rotational speed 7200 rpm
  - Sustained data rate 8.3-13.3 MB/s
  - Average seek time 7.5 ms
  - Average latency 4.33 ms

SCSIコントローラ

- DMA方式
  - コマンドを送ればコントローラが転送
  - CPUは次の作業へ進める
- WRITE　コマンド送れば完了
- READ　結果を待つ必要あり
- 複数のコントローラ
  - 同時書き込みが可能になる

Network I/F

- あまり選択の余地がない
- Ethernet
  - 10M or 100M
  - 良いSwitchと組み合わせてfull-duplex
- FDDI
  - 高価なのがネック
- 最近のCPUは100Mbpsを使いきる能力

Operating System

OSのチューニング

- 与えられたまま使っていては負け
  - 不要なdaemon類は動かないようにする
    - Solarisは多くのdaemon processが動く
    - FreeBSDの場合、不要なものはあまり動かない
  - 変更できるカーネルなら余計なドライバを削る
    - BSD, Linux等…
- メモリの節約、プロセステーブルの節約
  - システム全体のスループットの向上

Solaris 2.6の標準状態(1/2)

- ps -efaの結果

```
     UID   PID  PPID  C    STIME TTY      TIME CMD
    root     0     0  0   Oct 13 ?        0:01 sched
    root     1     0  0   Oct 13 ?        0:00 /etc/init -
    root     2     0  0   Oct 13 ?        0:00 pageout
    root     3     0  0   Oct 13 ?       13:12 fsflush
    root   195     1  0   Oct 13 ?        0:00 /usr/lib/sendmail -bd -q1h
    root   167     1  0   Oct 13 ?        0:03 /usr/sbin/cron
    root   277     1  0   Oct 13 ?        0:00 /usr/lib/saf/sac -t 300
    root    75     1  0   Oct 13 ?        0:00 /usr/sbin/aspppd -d 1
    root    96     1  0   Oct 13 ?        0:00 /usr/sbin/in.rdisc -s
    root   106     1  0   Oct 13 ?        0:00 /usr/sbin/rpcbind
    root   133     1  0   Oct 13 ?        0:00 /usr/sbin/inetd -s
    root   108     1  0   Oct 13 ?        0:00 /usr/sbin/keyserv
    root   153     1  0   Oct 13 ?        0:00 /usr/sbin/syslogd -n -z 12
    root   138     1  0   Oct 13 ?        0:00 /usr/lib/nfs/statd
    root   140     1  0   Oct 13 ?        0:00 /usr/lib/nfs/lockd
    root   173     1  0   Oct 13 ?        0:00 /usr/sbin/nscd
    root   183     1  0   Oct 13 ?        0:00 /usr/lib/lpsched
    root   280     1  0   Oct 13 ?        0:00 /usr/dt/bin/dtlogin -daemo
```

Solaris 2.6の標準状態(2/2)

```
    root   212     1  0   Oct 13 ?        0:02 /usr/sbin/vold
    root   218   216  0   Oct 13 ?        0:00 /usr/sbin/ccv -f
    root   205     1  0   Oct 13 ?        0:00 /usr/lib/utmpd
    root   216     1  0   Oct 13 ?        0:00 /usr/sbin/cssd
    root   217   216  0   Oct 13 ?        0:00 /usr/sbin/cs00
    root   219   216  0   Oct 13 ?        0:00 /usr/sbin/kkcv -f
    root   221     1  0   Oct 13 ?        0:00 /usr/lib/locale/ja/wnn/dpkeyserv
    root   225     1  0   Oct 13 ?        0:00 /usr/lib/locale/ja/wnn/jserver
    root   226   225  0   Oct 13 ?        0:01 /usr/lib/locale/ja/wnn/jserver_m
    root   281   277  0   Oct 13 ?        0:00 /usr/lib/saf/ttymon
    root   278     1  0   Oct 13 console  0:00 /usr/lib/saf/ttymon -g -h -p xxxxx console login:  -T
sun -d /
    root   260     1  0   Oct 13 ?        0:00 /usr/lib/snmp/snmpdx -y -c /etc/snmp/conf
    root   270     1  0   Oct 13 ?        0:00 /usr/lib/dmi/snmpXdmid -s xxxxxx
    root   282   260  0   Oct 13 ?        0:00 mibiisa -p 32788
    root   269     1  0   Oct 13 ?        0:00 /usr/lib/dmi/dmispd
    root 11379     1  0   Nov 06 ?        0:00 /usr/openwin/bin/fbconsole -d :0
    root 11376   280  0   Nov 06 ?        0:02 /usr/openwin/bin/Xsun :0 -nobanner -auth /var/dt/A:0-
8Iv0z_
    root 11301   133  0   Nov 06 ?        0:00 /usr/dt/bin/rpc.ttdbserverd
    root 11393 11377  0   Nov 06 ?        0:01 dtgreet -display :0
    root 11377   280  0   Nov 06 ?        0:00 /usr/dt/bin/dtlogin -daemon
```

停止したサービス

- インターネットサーバーに OpenWindowsは不要
  - 必要ならば手動で起動することにして、セッションマネージャ類を止める
- 仮名漢字変換関連のdaemonを止める
  - Sol 2.6の場合、適当にインストールするとWnnとATOKのサーバーが動く

停止したサービス(2)

- NFS, RPC, NISは利用しない
  - inetd.confからrpc関連をコメントアウト
  - rpcbind(いわゆるportmap)を停止
- その他
  - vold, snmp, lpsched等
- 場合によってはsendmailも止める
  - sendmail -q30m として-bd を抜く方法もある

Solaris 2.6のチューニング後

- ps -efaの結果

```
     UID   PID  PPID  C    STIME TTY      TIME CMD
    root     0     0  0   Sep 26 ?        0:00 sched
    root     1     0  0   Sep 26 ?        0:18 /etc/init -
    root     2     0  0   Sep 26 ?        0:00 pageout
    root     3     0  1   Sep 26 ?      390:47 fsflush
    root   146     1  0   Sep 26 ?        0:00 /usr/lib/sendmail -bd -q1h
    root   214     1  0   Sep 26 ?        0:00 /usr/lib/saf/sac -t 300
    root   104     1  0   Sep 26 ?        0:18 /usr/sbin/inetd -s
    root    94     1  0   Sep 26 ?        1:10 /usr/sbin/in.named
    root   109     1  0   Sep 26 ?        0:21 /usr/sbin/syslogd -n -z 12
    root   126     1  0   Sep 26 ?        1:02 /usr/lib/inet/xntpd
    root   136     1  0   Sep 26 ?       13:51 /usr/sbin/nscd
    root   130     1  0   Sep 26 ?        0:16 /usr/sbin/cron
    root   156     1  0   Sep 26 ?        0:01 /usr/lib/utmpd
    root 19855     1  0   Oct 30 console  0:00 /usr/lib/saf/ttymon -g -h -p xxxx console login:
-T sun -d /d
    root   221   214  0   Sep 26 ?        0:00 /usr/lib/saf/ttymon
```

- Sol 2.xは24MmemのSPARC IPCでも使える

サービスの止めすぎに注意

- 不要だと思って止めたところで.....
  - OpenWindowsを起動したら、
    起動するまでに異様な時間がかかる
  - Netscape Enterprise Serverが起動しない
    - apacheは動くのに ...
  - あたりをつけて、1つづつサービスを動かして
    確認する(しかない)
- 止めるとパフォーマンスを落とすサービス

FreeBSDのGENERICカーネル

- インストールがだいたいうまくいくような、最大公約数的カーネル
  - SCSIドライバが10種類以上
  - ネットワークドライバも10種類以上
  - CD-ROM用ドライバがいろいろ
- カーネル内部のテーブルも小さい
  - 1人のユーザで使う分には、ほぼ間に合う
- とりあえず使うにはなんとかなる程度

リコンフィグ

- 不要なファイルシステム、ドライバを削る
  - NFS, MSDOSFS, CD9660… ed0, aic0, nca0…
- テーブルを必要な大きさに増やす
  - maxusers, NMBCLUSTERS, FD_SETSIZE…
- LINT(/sys/i386/conf)を眺めながらGENERICを修正する
- その他、各種パラメータのチューニング

FreeBSDのカーネル

- インストール直後はGENERICカーネル
- カーネルのリコンフィグを行う

FreeBSD 2.2.7-RELEASEでGENERICカーネルとリコンフィグ後のカーネル

```
$ ls -l kernel.GENERIC kernel
-rwxr-xr-x  1 root  wheel  1564800 Aug  5 03:57 kernel.GENERIC
-r-xr-xr-x  1 root  wheel   764284 Sep 29 08:45 kernel

$ size kernel.GENERIC kernel
text    data    bss     dec     hex
1294336 81920   91112   1467368 1663e8  kernel.GENERIC
589824  53248   51600   694672  a9990   kernel
```

Solarisの場合

- /etc/systemのチューニング
  - FDやpty, 共有メモリなどの変更
  - AnswerBook, Solaris FAQを参考に…
    - 変更後はリブートが必要
- nddコマンドと/dev/tcpによるtcpパラメータのチューニング
  - /usr/sbin/ndd /dev/tcp ?
- 変更が必要になるのは、特別な場合

Server Software

WWWサーバー

- HTTPのリクエストを処理するサーバー
  - インターネットでもっとも稼動台数が多い?
- HTTP処理
  1. TCP接続
  2. クライアントからのリクエスト
  3. サーバーのレスポンス
  4. TCP切断
  - TCP Connectionの断続

WWWサーバー
レスポンスまでの処理

- リクエストの解析
  - どのページ?、CGI?、URLのmapping?
- ページデータの読み出しやCGIの実行
- ログの生成
  - (必要なら)IPアドレスからFQDNへのネームサーバーへの問い合わせ
- レスポンス

WWWサーバー
ボトルネックの要因(1)

- コンテンツデータの読み出し
  - Disk I/O
- cgiプログラムの実行
  - CPUパフォーマンス
  - メモリ不足によるスワップ
  - プログラムの実行が遅い
    - C vs Perl ?

WWWサーバー
ボトルネックの要因(2)

- ログの書き出し
  - 100万hit/dayの場合、10行/秒を越える
    - ヘビーなサーバーではログファイルも無視できない
  - DNSを調べてFQDNで記録する場合の問題
    - ログの記録が遅くなる
- ネットワークロケーション
  - サーバー側はISPのハウジングサービス

wwwサーバー
1requestの負荷

- リクエスト数×処理時間 を恒に考慮する
- 1リクエストに 1秒かかり、メモリが1M
  - 10リクエスト/秒なら、メモリは10M
  - 100リクエスト/秒なら、メモリは100M
- CPUが10倍速なら、処理時間は 0.1秒
  - 100リクエスト/秒でも、メモリは10Mですむ
- ネットワークの転送スピードも考慮
  - クライアントが遅いと転送時間は変わらない

WWW Server Software

- CERN　リクエスト毎にfork
- NCSA httpd　高機能でCERNより高速
- Apache forkしたプロセスにfd passing
  NCSA httpdも採用
- phttpd　forkの代わりにthreadを利用
- thttpd　select を使ったループで処理

CERN Server

- WEB Serverのサンプルインプリメント
- WEB Proxyサーバーとキャッシュ
  - 兼用サーバーが簡単に…
- 1リクエスト毎にfork
  - リクエストを受けた後にfork
  - UNIX向きの処理方法だが、forkの処理は重くパフォーマンスの低下
- 最近はあまり使われてない?

NCSA httpd

- 高機能のWEBサーバー
  - URLのリダイレクト
  - アクセス制限
  - 柔軟な設定
  - CERNよりは高速
- 機能が高いため一時広く利用される

Apache WEB サーバー

- 正確な動作を重視したインプリメント
  - スピードよりも正確さを優先
  - しかしながら十分高速
- 予めforkしたプロセスを用意
  - リクエストに応じて子プロセスを順に利用
- バーチャルホスト対応
- 現在世界でもっとも稼動台数の多いWEB

phttpd & thttpd

- phttpd
  - thread を利用し、速いらしい。
    - Netscape Enterprise Serverも同様の構造
  - 移植性に難あり
    - ほとんど Solaris 専用
- thttpd
  - 複数コネクションを selectを使って同時に処理
  - かなり高速だが、基本機能のみ

NEWSサーバー

- トラフィック　90万通/day 18〜20GByte/day
  - 現状もっともヘビーなサーバー
- 他のサーバーとの記事の交換
- エンドユーザー向けのサービス
  - 記事の購読、投稿
- コントロール処理

NEWSサーバー
記事の受信

- 他のサーバーからの受信
  - 記事の重複の検索
  - スプールへの書き込み
  - 送信するためのファイルの生成
    - いずれもDisk I/Oの発生
- full feedのボリュームを受けるのは大変
  - T1では足りない

NEWSサーバー
記事の送信

- 他のサーバーへの記事の送り込み
  - 記事の送信ファイルの読み込み
  - スプールから読み取った内容の送信
    - 配送先が増えるにつれて、ディスクI/Oの増大

NEWSサーバー
ユーザーの購読用

- ニュースグループと記事番号の管理
- overview 情報
- history情報からスプールの記事へ対応
  - 対象とするユーザー数
- cancel処理

従来のINN (before 1.x)

- スプール形式にufs (UNIX File System)
- 記事番号のファイル名
- ニュースグループのディレクトリ
- news.software.nntpの350番
  - /var/spool/news/news/software/nntp/350
- cancelが重い
  - ufsがボトルネック

ufsの問題

- ファイルを作る
  - 同じファイル名が存在しているかどうかを
    リニアサーチ
- ファイルを消す
  - 目的のファイルが見つかるまでリニアサーチ
- ディレクトリに大量のファイルがある場合の
  パフォーマンス低下を招く

Diablo

- 元々配送専用サーバー
  - 当初ユーザーが読む機構は無かった
- キャンセル処理をしない
- activeやnewsgroupsファイルが無い
  - 購読用ならある
- INN 1.xの5〜10倍のパフォーマンス
- news spoolは独自の形式

現在のINN (version 2.x)

- CNFS (Cyclic News File System)
- 巨大なファイル内部に記事を格納
- 先頭から記事を置き、
  最後まで使ったらまた先頭へ
- history には記事のファイル中の
  offsetを記録
- cancelが劇的改善

Mailサーバー

- 2種類のMailサーバー
  - メーリングリスト用メールサーバー
  - PCユーザーなどのPOP&SMTPサーバー
- それぞれ、チューニングポイントが違う
  - より速く大量に
  - より多くのユーザーを1台のマシンで

メーリングリストサーバー

- 多くの宛先にメールを送る
  - できる限り高速に
- 配送が遅いと議論がかみ合わない
  - sendmailは1アドレスづつ順に配る
- qmail
  - 小さなプログラムで次々配送
- sendmail + WIDE patch + smtpfeed
  - selectを利用し同時に複数の宛先へ配送

sendmailの特徴

- 逐次配送
  - 前の配送が終わらないと次を配送しない
    - 配送先がたまたまアンリーチだったりすると、その後の宛先全部が遅れる
- 同一MX先の相乗り
  - user1@foo.co.jp user2@sh.foo.co.jpが同じMXなら1通で配送

qmailの特徴

- 複数の小さなプログラムを組み合わせ用途別に処理
  - qmail-smtpd, qmail-inject, qmail-remote…
- security holeを発生しないような構造
  - security holeの発見に$1,000の賞金
- 軽くて高速
- 同一ドメインであっても1通づつ配送

sendmail + WIDE Patch + smtpfeed

- smtpfeedが実際の配送を行う
  - sendmailの外部メーラ
  - sendmailとは LMTPで通信
    - WIDE patch が必要
- MXの相乗り
- select を使って複数に同時配送
  - きわめて高速
  - DX4で 1600アドレスを3分以内に90%配送

POP,SMTPサーバー

- SMTPについてはメールサーバーと同様
  - SPAM対策はちゃんとする(おまけ)
- qpopper
  - もっとも普及しているpopサーバー
- 同一ディレクトリに大量ファイルの問題
  - ufsボトルネック問題

qpopperの注意点

- qpopperの挙動
  - リクエストがあるとメールボックスを1行づつテンポラリファイルにコピー
  - 接続終了時に、元のファイルに書き戻す
- テンポラリファイルを置くディスクを別にするように改造する

OSのインストール

- 不要なものまでインストールしない
  - Solarisの場合、CDE環境無しでも良い?
- パーティションの切り方
  - 『swapサイズはメモリの倍とる』は過去の話
    - メモリ512Mでswap 1Gなどというのは愚の骨頂
  - 十分なメモリを用意したらswapは0でよい
    - OSによってはswap 0が不可なものもあるので注意

運用開始後の監視

- 正常に動作しているか
- ボトルネックはないか？
- vmstat, iostat, netstat, ps などでチェック
  - あるマシンの正常運用状態
  - 破綻状態

Real Example

news.nspixp.wide.ad.jp [1996/10]

- IXでのHUBとなるニュースサーバー
  - WIDE Projectのnspixpの一環
  - 当時のトラフィックを支えてそこそこ速いこと
- スペック
  - P5-133, Mem 128MB
  - HDは 2G × 2 + 4G
    - 2Gはシステムとhistory、4Gはスプール
    - SCSIは2チャンネル

news.nspixp.wide.ad.jp日記 (1)

- ソフトウェアはdiabloを利用
- CPUの余裕が少なくなったためP6-200へ
- Etherを 10Mbps→100Mbps に変更
  - 100Mのつもりが、Etherスイッチの設定ミス
- mount optionにnoatimeを追加
  - ちょっとだけパフォーマンス向上
  - async optionも試してみたが効果なし.
    おそらく メモリ不足 (この時点では128M)

news.nspixp.wide.ad.jp日記 (2)

- diablo を shmを使うように修正
  - configのミスで使われていなかった
- incoming 18.57GB/日の記録
  - outgoing 43.17GBと減る
- メモリを128MB → 256MB
  - スプールのexpire時間が半分に短縮

news.nspixp2.wide.ad.jp [1998/1]

- Pentium-II 333
  - スピードではなく入手しやすさと、発熱量
- 384MB メモリ
  - MBの限界 (128MB DIMM x 3)
- スプールのI/Oを稼ぐ
  - 9G×2 を ccd でストライプする
- 100Mbps ネットワーク
  - nspixp2の事情によりFDDI

*news.nspixp2.wide.ad.jp*の実力

- 31のISPと接続
- トラフィック
  - incoming　55〜80万通　18〜23GB
  - outgoing　800〜 1000万通　270〜330GB
- 最大トラフィック
  - 12/6　in 25.7GB　out 361GB
- ピーク時はFDDI帯域を使い切る

sh.janog.gr.jp [1998/8]

- JANOGのメール、WWWサーバー
- ハードウェア
  - CPU　DX4ODP100
  - メモリー　16M
  - Hard Disk　SCSI 2G ✕ 1
  - MB　ISA/VL
- お金をかけずに作るのが目標
  - ベースは、ゴミになる直前を拾ってきた(^^;

sh.janog.gr.jp (2)

- Software構成
  - OS　FreeBSD 2.2.7 RELEASE
  - www　thttpd 2.04
  - mail　sendmail 8.9.1a + WIDE patch 3.1W + smtpfeed 0.90
- メモリ16Mで現状十分
  - 定常的なprocessが増えたりしない
  - ログインしたときのシェルの使うメモリが…

COTURA AERO 4/33C [1998/11]

- COMPAQのサブノートパソコン[1994?]
  - CPU　SL486SX/33MHz
  - MEM　12M
  - HD　IDE 840M (元々は 250M)
    - 極めて非力(^^;
- FreeBSD 2.2.7 with PAO
  - PAOでPCMCIA Ethernet Card

COTURA AERO 4/33C (2)

- 小規模なWEBとProxyサーバー
  - WWWサーバー　thttpd
  - proxy サーバー　squid
- squidはこの規模のマシンには重過ぎる
  - 5M以上のメモリ消費
  - アクセスの無い時も10〜20%のCPUを消費
- 結局apache + mod_proxy
  - thttpdも無駄なのでやめる

*Solaris 2.x*での例

- Ultra1で512Mmemなマシンがあり
  1Gのメモリが要求された
  - 6~70M規模のプロセスが数個常駐
- Sun純正メモリは定価で200万円近い
  - 実売は110万円ぐらい??
- UltraSPARC 167MHzで
  メモリを1Gも積んで大丈夫？
  - システムバランスの問題

*Solaris 2.x*での例
*SPARCengine Ultra AXi*の導入

- SUNのOEM用プロダクトである SPARCengine Ultra AXiを利用
  - ATXのボードに300MHz UltraSPARC-IIi
  - CPU＋ボード、1Gメモリ、4.3G HD x 2、2nd Ether I/F、VIDEO、19inch 4Uなケース
  - モニタ無し、CD-ROMドライブもなし
- PCを組み立てられない人、Sunに詳しくない人にはお勧めしない！

高速化のためのTIPS(1)

- システムのボトルネックを探す
- 8-2の法則
  - 8割の処理が2割の部分で行われている
  - その2割の部分を改善すれば、
    全体では大きく性能アップ
- インターネットサーバーでは、多くの場合CPUよりディスクアクセスにある
  - 見つけにくい場合も多い

高速化のためのTIPS(2)

- 速いHDを選ぶ
- HDのアクセス(head seek)を減らす
  - メモリを十分に積んでバッファを効かせる
  - 物理的に複数台のディスクを利用し、1台あたりのアクセスを減らす
    - ストライプは効果的
  - 容量ぎりぎりまで使わない
- HDの転送レートを上げる
  - コントローラを複数用意する

高速化のためのTIPS(3)

- 人間を鍛える
  - 普段は遅いマシンを使う
  - 速いマシンでは気がつかない差が遅いマシンだと気がつきやすい
  - 条件の違うマシンをいろいろ用意して同じことをやってみる
  - 結果の違いはどこからくるのかを見極める
- ノウハウは聞いただけでは身につきにくい

システムを作るときのポイント

- 目的をよく見極める
  - スピード?、コスト?、省電力?、省スペース?
- 予算は絶対に無視できない
  - 余ってるものがあるなら使ってみる
- 置けなくなったら負け
  - 必要とするスペースと電力も考慮する

まとめ

- 限界のように見えるマシンでも、能力を活用していないことが多い
  - ボトルネックをさがす
  - リソースは無駄なく有効に活用する
- CPUとネットワークは速く、ディスクは遅い
  - バランスの良いシステムを構築する
- チューニングは当たり前のことの積み重ね
  - 一つ一つ細かくチェックする

おしまい

```
===== Fri Dec 11 12:00:01 JST 1998 =====

  [ vmstat -w 5 ]

 procs    memory      page                    disks       faults       cpu
 r b w    avm    fre  flt  re  pi  po  fr  sr s0 s1 s2   in   sy  cs us sy id
1466 0419312 50468  106  40  35   0  22  41 33 19 19   75   29   7 12 16 72
2733 0426192 54636 1648  33 128   0 209 2190 89 62 62 8660 9366 2516 44 53  3
1426 0382264 58956 1863  27 105   0 175 1891 94 63 41 8691 6972 2348 43 56  0
 273 0384560 51708 1833  25  99   0 172    0 89 87 40 7922 6759 2100 40 48 12
 8 3 0387088 55276 1907  16  97   0 132 1283 53 68 39 7762 6433 2049 37 49 13
 7 2 0354468 60744 1282  28  85   0 159 1464 39 43 90 7249 4632 1733 35 42 24
 820 0378480 55040 1512  74  80   0 192    0 56 44 76 8188 5447 1974 38 49 13
12 6 0365884 57776 2402 145  83   1 183 1752 43 44 63 8248 8163 2260 45 52  3
1323 0405904 52764 2566  24  90   0 190    0 55 48 69 8460 8653 2121 46 53  1
 518 0375112 56268 2416  14 112   1 308 2348 54 93 52 7181 7935 2175 39 45 16
 4 2 0381444 51620 2552  45 102   0 352    0 23 74 59 7004 7929 2132 32 46 22
14 6 0337732 52160 2333  72 113   1 258 2209 33 59 71 7630 7690 2107 35 47 18
 342 0362280 58128 2273  15 130   0 589  716 59 83 94 6978 7249 1999 31 44 25
 2 7 0379584 51344 2295  10 161   0 396    0 54 106 65 6878 6858 2006 30 43 27
 0 4 0324636 54988 2045   5 113   0 233 1989 11 61 75 5919 6197 1593 25 35 40
 411 0338692 54376 1541  13  83   0 431    0 16 38 79 4941 5205 1505 21 29 50
 4 2 0332128 56488 1320  11 112   0 244 3191  9 51 94 4212 4387 1149 17 26 57
 051 0398720 53476 1736  20 114   0 433    0 28 52 87 5580 5883 1638 27 32 41
```

```
===== Mon Dec 7 12:00:02 JST 1998 =====

  [ vmstat -w 5 ]

 procs    memory      page                    disks       faults      cpu
 r b w    avm    fre  flt  re  pi  po  fr  sr s0 s1 s2   in   sy  cs us sy id
14142 01017612 49996   85  40  34   0  21   6 33 18 19    0  150 183 11 16 73
 0174 0994096 52544 1197  28 148   0 123 5488 85 71 61 3581 7211 1111 13 23 64
 5169 01038264 56440 1487  34 145   0 129 2597 97 71 63 4010 5305 1283 19 24 57
 1166 01025588 53208 1417  44 141   0  74 517 98 73 68 4870 5062 1342 19 30 51
 3136 01159784 51064 1410  30 150   0  94 3547 78 73 82 4003 5010 1093 17 26 58
 0152 01065944 55804 1344  30 138   2  27 3003 100 71 83 4818 4263 1308 20 29 52
 1156 0963764 52688 1262  72 125   0  39 3137 76 72 81 4816 4828 1309 17 29 54
 2126 0900560 50812 1165  23 123   0  25 402 78 69 81 4337 4182 1025 15 27 57
 3135 0890728 55984 1020  29 119   1  22 4992 67 71 64 4203 3976 1142 15 25 60
 0151 0994264 55104  858  37 116   0  22 989 62 73 82 3920 3422 1000 14 22 65
 2130 0945984 51656 1098  20 140   2  25 3227 72 73 83 4355 3837 964 16 25 59
 1157 0966072 56576  973   7 128   1  14 2999 73 67 76 4557 3397 1091 13 28 59
 4126 0910024 52700 1073  47 136   5  20 5971 40 70 80 5231 4178 1403 19 31 50
 1135 0969764 55812 1373  83 122   1  33 4563 74 70 89 4962 4723 1368 21 27 52
 3133 0940864 53868 1047  36 127   4  44 2103 70 80 78 4591 3982 1105 15 28 57
 2158 0979052 56912 1081  35 140   3   8 7942 42 76 78 4680 3388 898 15 27 57
 1130 0861736 53536 1449  13 135   0  13 1406 37 71 77 4215 4360 905 16 24 60
 1144 0888524 50360 1214  18 142   0   9 2153 53 69 84 4922 3859 1008 16 27 56
```

```
===== Fri Dec 11 12:00:01 JST 1998 =====

  [ iostat -w 5 -c 20 ]

     tty          sd0           sd1           sd2           cpu
 tin tout sps tps msps  sps tps msps  sps tps msps  us ni sy in id
   0    3   3  33  0.0  -12  19  0.0    8  19  0.0  12  0 10  6 72
   0 01544  89  0.0 2122  63  0.0 1858  61  0.0  44  0 36 17  3
   0 01727  94  0.0 2495  63  0.0 1166  42  0.0  44  0 34 21  0
   0 01465  89  0.0 2727  89  0.0 1243  39  0.0  39  0 31 16 14
   0  0 887  52  0.0 2186  64  0.0 1328  41  0.0  39  0 32 18 12
   0  0 704  37  0.0 1059  41  0.0 2468  88  0.0  35  0 27 14 24
   0  0 921  57  0.0 1353  47  0.0 2247  78  0.0  38  0 32 17 12
   0  0 786  42  0.0 1222  43  0.0 1893  64  0.0  45  0 32 20  3
   0  0 971  57  0.0 1591  49  0.0 2364  68  0.0  45  0 34 19  1
   0  0 925  53  0.0 2454  93  0.0 1562  52  0.0  39  0 29 16 16
   0  0 401  24  0.0 2349  74  0.0 1963  61  0.0  31  0 30 17 22
   0  0 631  34  0.0 1951  58  0.0 2478  71  0.0  35  0 27 19 18
   0  0 969  60  0.0 2437  84  0.0 2837  94  0.0  31  0 28 16 26
   0  0 795  45  0.0 3357 103  0.0 2280  72  0.0  28  0 25 15 31
   0  0 236  12  0.0 1522  49  0.0 1730  61  0.0  25  0 22 14 39
   0  0 262  14  0.0 1372  45  0.0 2746  94  0.0  22  0 18 12 48
   0  0 144   8  0.0 1569  49  0.0 2804  90  0.0  16  0 17  9 58
   0  0 572  34  0.0 1642  53  0.0 2688  86  0.0  28  0 19 12 40
   0  0 576  36  0.0 2719  98  0.0 1409  54  0.0  18  0 17  7 59
   0  0 598  32  0.0 2824  87  0.0 1868  56  0.0  14  0 12  8 66
```

```
===== Mon Dec 7 12:00:02 JST 1998 =====

  [ iostat -w 5 -c 20 ]

      tty          sd0           sd1           sd2           cpu
 tin tout sps tps msps  sps tps msps  sps tps msps  us ni sy in id
   0    2  -9  33  0.0    7  18  0.0    2  19  0.0  11  0 10  6 73
   0 01352  85  0.0 2092  71  0.0 1546  60  0.0  13  0 14  9 64
   0 01611  96  0.0 2093  71  0.0 2472  62  0.0  19  0 15  8 57
   0 01604  99  0.0 2021  73  0.0 2505  69  0.0  20  0 19 11 50
   0 01480  78  0.0 2263  73  0.0 2233  81  0.0  16  0 18  8 58
   0 01605 100  0.0 2469  72  0.0 1941  83  0.0  20  0 17 11 52
   0 01343  77  0.0 2176  71  0.0 2951  79  0.0  17  0 19 12 53
   0 01410  78  0.0 2440  69  0.0 2423  84  0.0  16  0 15 12 58
   0 01239  68  0.0 2364  72  0.0 2153  63  0.0  15  0 16 10 59
   0 01140  61  0.0 2099  72  0.0 2723  82  0.0  14  0 13  8 65
   0 01397  74  0.0 2276  73  0.0 2647  83  0.0  17  0 16  9 58
   0 01157  71  0.0 2582  67  0.0 2415  76  0.0  14  0 15 13 59
   0 0 764  43  0.0 2348  70  0.0 2815  80  0.0  19  0 16 14 51
   0 01334  74  0.0 2631  71  0.0 2677  89  0.0  21  0 16 11 52
   0 01213  70  0.0 2429  80  0.0 2761  78  0.0  16  0 18  9 57
   0 0 610  40  0.0 2470  76  0.0 2596  78  0.0  15  0 15 12 58
   0 0 736  40  0.0 2760  71  0.0 2439  77  0.0  17  0 16  9 59
   0 0 825  50  0.0 2457  69  0.0 2503  84  0.0  16  0 17 10 57
   0 01061  62  0.0 2321  75  0.0 2479  75  0.0  16  0 16 13 55
   0 0 622  33  0.0 2512  66  0.0 2421  82  0.0  14  0 14  8 65
```

```
===== Fri Dec 11 12:00:01 JST 1998 =====

  [ netstat -i 5 ]
            input        (Total)           output
   packets  errs      bytes    packets  errs      bytes colls
     20082     0    3373662      37924     0   55867653     0
     20722     0    5664223      37142     0   53656333     0
     16878     0    4299650      29939     0   42283387     0
     19047     0    3341410      35891     0   53336774     0
     15966     0    3035188      30484     0   45189333     0
     18290     0    3308978      35457     0   52559059     0
     18393     0    3843862      35621     0   53247724     0
     19775     0    6655625      33588     0   48640327     0
     15371     0    2860193      28470     0   42038780     0
     15311     0    4058512      26478     0   38251201     0
     16498     0    3612798      30225     0   43265481     0
     14474     0    3466973      24921     0   36412454     0
     14151     0    2933800      25889     0   37775740     0
     11760     0    1595203      21797     0   31713464     0
     10497     0    3290418      16663     0   23976990     0
      8116     0    3254345      12845     0   17885429     0
     11487     0    2773769      20162     0   29070078     0
      8713     0    2010270      16695     0   23845998     0
      6156     0    1176552      10684     0   14655538     0
      6554     0    1132713      11446     0   16033995     0
```

```
===== Mon Dec 7 12:00:02 JST 1998 =====

  [ netstat -i 5 ]
            input        (Total)           output
   packets  errs      bytes    packets  errs      bytes colls
      6798     0    1422026      11369     0   14650591     0
      8192     0    2946134      12415     0   15053805     0
      9836     0    3419121      15567     0   20326622     0
      7854     0    2196073      12615     0   16494031     0
      9911     0    3239666      16342     0   22895964     0
     10288     0    5611769      14252     0   17515807     0
      8445     0    3039962      13335     0   17190700     0
      8738     0    3885862      13123     0   16607982     0
      8125     0    3656387      11772     0   14778315     0
      8221     0    2035852      14070     0   18827103     0
      9399     0    4086644      14294     0   18336174     0
     11755     0    5306250      17632     0   22554671     0
     10210     0    4199156      15888     0   20613130     0
     10487     0    5512400      14646     0   18317779     0
      8764     0    2153825      15115     0   19965130     0
      8241     0    2852509      13234     0   16745072     0
     10230     0    3442973      17124     0   23023510     0
      9565     0    3496506      15870     0   20661694     0
      8326     0    3895470      12855     0   16254892     0
      9811     0    3362258      15845     0   21133382     0
```