スケーラビリティーを考慮したインターネットサーバーの構築

東京大学情報基盤センター
Internet Week 2000 Tutorial @ Osaka
19 Dec 2000
安東　孝二
<chutzpah@ecc.u-tokyo.ac.jp>

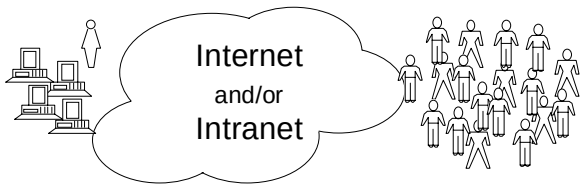
目次

Part1　概論

- インターネットサーバーとは
- インターネットサーバーを取り巻く環境
- インターネットサーバーに求められるもの
- インターネットサーバーのソリューションのために

インターネットサーバーとは

- インターネット技術を用いてサービスを行うサーバー
  - インターネット技術≒TCP／IP技術？



インターネットサーバーを取り巻く環境

- 家庭用高速ネットワーク
- 常時接続ネットワーク
- クライアントの高性能化
- マルチメディアコンテンツ
- ユーザー数の激増
- e-commerce, net-banking,etc.

家庭用高速ネットワーク

- xDSL
  - ADSL、SDSL　～10Mbps
  - VDSL　～50Mbps
- CATV
  - 数Mbps
- FTTH？

常時接続ネットワーク

- xDSL
- CATV
- FLET‘S　ISDN

クライアントの高性能化

2000年11月初旬

- 1GHzのCPUが4万円以下
  - 1年前は同価格帯で500から600MHz
- HDDの1GBあたりの単価が300円台
  - 1年前は800から900円台
- 128MBメモリが6，000円台
  - 1年前は2万円台

マルチメディアコンテンツの普及

- テキスト以外のコンテンツサービス
  - 動画ストリーム
    - Realplayer
    - CU-SeeMe, NetMeeting
  - MPEG
    - Napster MP3トラフィックの増大

ユーザー数の激増

- 従来型のインターネットユーザーの増加
- 携帯電話サービスを通じた新しいユーザーの増加
  - i-mode,Ezweb,J-sky
- PC以外のアプライアンスの出現

e-commerce, net-banking,etc.

- インターネットを通じた商取引の普及
  - 株売買、オークション、etc.
  - QoSへ対する厳しい要請
    - **“eight-second rule”**
    - **security**
    - **performance**
    - **availability**

ユーザーに見せてはいけないメッセージの例

- An error has occurred! Our server have reached the maximum number of simultaneous users. Please try again later.
- エラーが発生しました。このサーバーは同時使用ユーザー数が限界に達しました。しばらくしてアクセスしてください。

インターネットサーバーに求められるもの1

- 速さ
  - “eight-second rule”に勝つ
  - 最低限の性能を保証する
- 安さ
  - TCOを低く押さえる
    - Capital equipment
    - Network cost
    - Operational cost

インターネットサーバーに求められるもの2

- 信頼
  - **security**
    - 暗号化
    - 認証
  - **fault-tolerance**
    - 多重防護
  - **fool-proof**

インターネットサーバーのソリューションのために

- 戦略的なsite planning！
  - performance modeling
  - （既存のシステムがある場合は）system analysis
  - workload prediction and/or forecasting
  - cost modeling
  - capacity planning
    - (capacity)=(vertical) x (horizontal)
      - » vertical scaling up
      - » horizontal scaling up

ここまでの要約

- システムソリューションにおいて、スケーラビリティは避けては通れない命題
  - クライアントサイドの急激な高性能化
  - コンテンツの大容量化、ユーザーの増加
  - 社会からのインフラとしての要求

Part2　スケーラビリティ

- スケーラビリティとは
- スケーラビリティ確保のための提案と注意

スケーラビリティとは

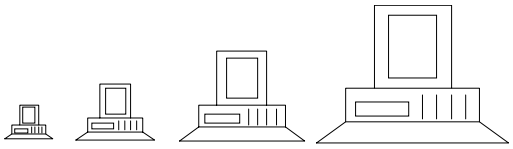
- 規模適応性
  - ○○○○の規模適応性
- 同一のサービスを、増大するリクエストに応じた任意の規模で、要求性能が保証されるように、提供できる余地がある
- 別の要求として、当然コストやリスクを最小限にすることが求められる
- 要求に応じた規模への適応が可能なこと
  - 「追従可能」
  - scale up and/or scale down
  - インターネットサーバーにおいては、通常は増大する規模への適応が求められる

スケーラビリティの確保のための提案と注意

- 単一ホストレベル
  - vertical scaling
- システムレベル
  - horizontal scaling
  - クラスタ技術,etc.
- vertical & horizontal
  - Quality & Quantity

ホストレベルのスケーラビリティ1

- vertical scaling 単体の能力を上げる



ホストレベルの スケーラビリティ2

- ホスト単体としてのスケーラビリティとは
- コンピュータコンポーネント別の可能性
- 各種アクセラレータの可能性

ホスト単体としての スケーラビリティ

- 管理者の予想に応じた挙動
  - ゆっくりでもしっかり動く
    - 挙動がリニアに推移してほしい
  - 限界が見えることも重要
- 拡張性が期待できること

コンピュータコンポーネント（プロセッサ）

- マルチプロセッサへの適応
  - 3個以上のプロセッサを利用できるマシンは高価
  - 基本的にリニアにはスケールしない
- 一部機種では、ダイナミックアサインが可能

コンピュータコンポーネント（メモリ）

- スロットが限られているため、基本的にメモリモジュール依存
- 一部機種ではダイナミックにメモリを増減することが可能

コンピュータコンポーネント（ストレージ）

- HDDについてはハードウェアRAIDの導入が効果的
  - 安価なIDE RAIDから高価なSANストレージまで選択肢は広い
  - コントローラチップの信頼性とキャッシュメモリの量に注意
- テープ装置についてはHDDに比べソリューションが少ない、もしくは高価

コンピュータコンポーネント（ネットワーク）

- ～1Gbps
  - GbEはバススピードの制限やドライバの出来によって、性能に多くのばらつき
    - PCIの場合、rev.3(64bit,66MHz)で初めてGbEを最大限に利用できる
    - TCP/IPスタックの出来によっては500Mbpsくらいしかでない

コンピュータコンポーネント（ネットワーク）

- trunking
  - CPUに負荷
  - NICの故障の問題

コンピュータコンポーネント（OS）

- パーティションサイズ、ファイルサイズなどの制限
- クラスタリング可能かどうか
- 各種ドライバーの有無

コンピュータコンポーネント（アプリケーション）

- WEBサーバーについて
  - マルチスレッド
  - ボトルネックになりうるCGIなど工夫が可能か
- メールサーバーについて
  - マルチスレッド

特にUNIXで気をつけたいこと

- forkだめ
- swapだめ
- inetdだめ

アプリケーションのちょっとした工夫

- たとえば。。。。。
  - 大量のcgiを処理するには
    - cgi専用サーバ
    - cgi処理専用プロセス（あらかじめ埋めておく）

各種アクセラレータ1

- WEBアクセラレータボード
  - Akamba社　Velobahn
- WEBアクセラレータ
  - CacheFlowなど多数

各種アクセラレータ2

- SSLアクセラレータボード
  - Sun Microsystems社　Sun Crypto Accelerator I
  - nCipher社　nFast
  - Compaq社　AXL 200 SSLアクセラレータ

各種アクセラレータ3

- SSLアクセラレータ
  - Intel社 Netstructure
  - F5 Networks社　Big IP
  - nCipher社　nFast
  - Alteon iSD-SSLアクセラレータ

各種アクセラレータ4

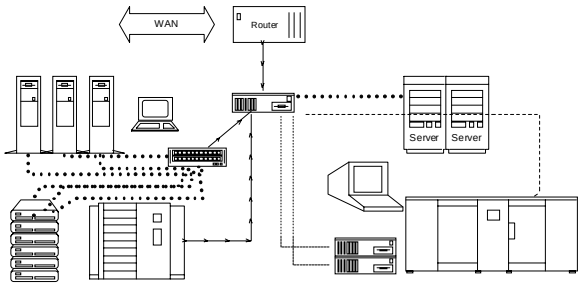
- メールサーバーのアクセラレータ
  - 特にH/Wアクセラレータはない？
  - あえて言えば、SSD(Solid State Disk)の利用など？

ソフトウェアでaccelerate できることも！

- sendmailをsmtpfeedで高速化
  - 効果がない場合もある

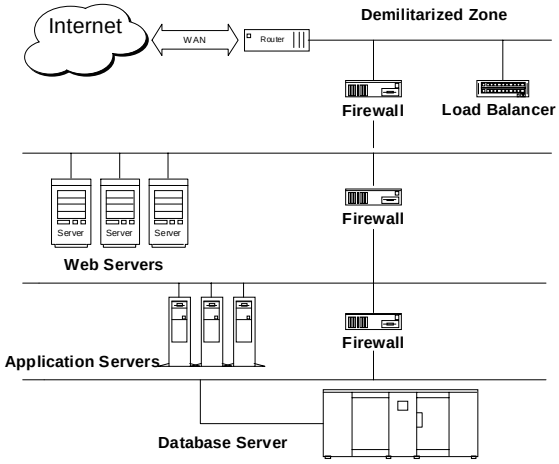
システムレベルのスケーラビリティ

- horizontal scaling　複数のマシンで全体性能を上げる



WEBサーバーのhorizontal scaling1

- 通常のe-Commerce WEBシステム構成



WEBサーバーのhorizontal scaling2

- DNS based method
  - DNS round robinを利用
  - DNSの情報がキャッシュされることがあるが、大域的には分散可能
  - explicitに個々のサーバーを指定されてしまう恐れ
  - 書き込みを伴う場合は、同期もしくは、共有ストレージの排他制御が必要

WEBサーバーのhorizontal scaling3

- Server-based method
  - URL redirectを用いる
  - redirectをするサーバーには最初のセッションが必ず集中
  - workloadに応じた効率的な分散が可能
  - 書き込みを伴う場合は、同期もしくは、共有ストレージの排他制御が必要

WEBサーバーのhorizontal scaling4

- dispatcher-based method
  - L4 SwitchでのdispatchL4 Switchでのdispatch
  - 連続性を求めるセッションに対し、SSL,Cookieなどの対処が必要
  - 書き込みを伴う場合は、同期もしくは、共有ストレージの排他制御が必要

メールサーバーの horizontal scaling(SMTP)1

- DNS based method
  - DNS round robinを利用（MXレコード）
  - DNSの情報がキャッシュされることがあるが、大域的には分散可能
  - explicitに個々のサーバーを指定されてしまう恐れ
  - 共有ストレージの排他制御

メールサーバーの horizontal scaling(SMTP)2

- Server based method
  - Mirapoint社 Message Routerでのルーティング
  - Sun|Netscape alliance iPlanet Messaging Serverでのルーティング
- dispatcher based method
  - L4スイッチでのdispatch
  - 共有ストレージの排他制御

メールサーバーの horizontal scaling(POP,IMAP)1

- DNS based method
  - DNS round robinを利用
  - DNSの情報がキャッシュされることがあるが、大域的には分散可能
  - explicitに個々のサーバーを指定されてしまう恐れ
  - 共有ストレージの排他制御

メールサーバーの horizontal scaling(POP,IMAP)2

- Server based method
  - Mirapoint社 Message Routerでのルーティング
  - Sun|Netscape alliance iPlanet Messaging Serverでのルーティング

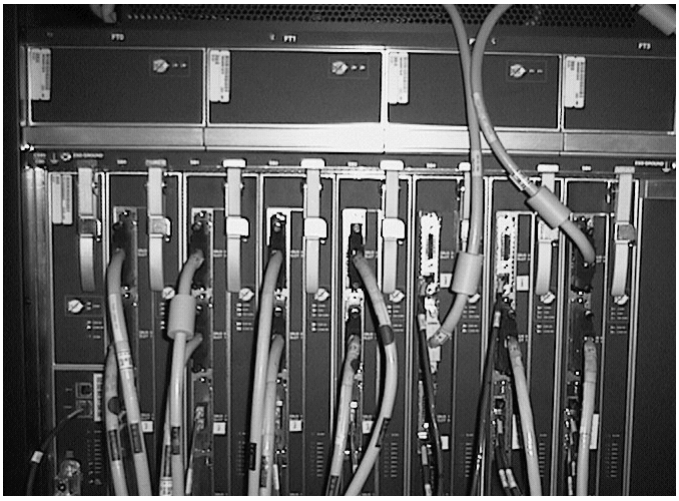
メールサーバーの horizontal scaling(POP,IMAP)3

- dispatcher based method
  - L4スイッチでのdispatch
  - 共有ストレージの排他制御

参考実例

- Sun Enterprise 10000のスケーラビリティ
- Mirapoint Message Routerの仕組み
- Intel Netstructure の仕組み

Sun Enterprise 10000の スケーラビリティ



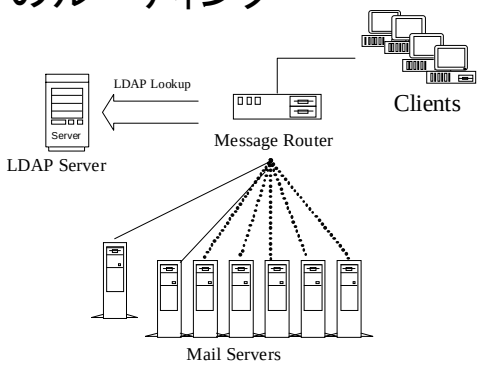
Sun Enterprise 10000の スケーラビリティ

- 最大16枚のシステムボードのクラスター
- ダイナミックなドメイン構成も可能
- 各種デバイスのホットスワップが可能
- Solarisが稼動

Mirapoint Message Routerの 仕組み

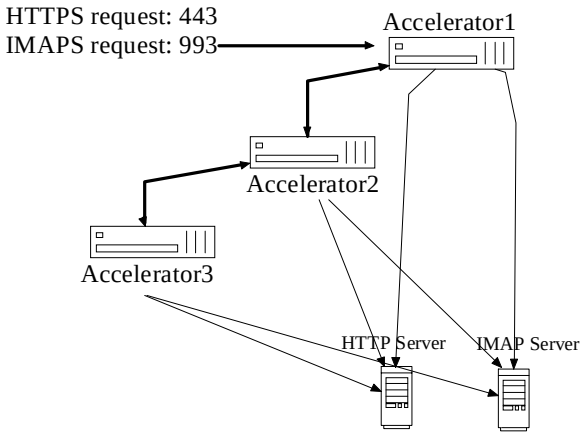
Mirapoint Message Routerの仕組み

- LDAP情報によるSMTPおよびPOP,IMAPセッションのルーティング



SSLアクセラレータの仕組み1

- **Intel Netstructure 7110 e-Commerce Acceleratorの場合**



SSLアクセラレータの仕組み2

- SSLのencode/decodeとdst portの変換を行う
- 性能を保証するために、２００セッション以上は受けない
- あふれたセッションは控えているアクセラレータが行う（カスケードして運用）

Part3　システムプランニング

- vertical scaling versus horizontal scaling
- スケーラビリティーを考慮したシステムプランニングのためには
- 最近の流行

vertical scaling(scaling up) と horizontal scaling(scaling out)

- 水平方向のスケーラビリティはオペレーションコストを増大させる副作用を持つ
- 垂直方向のスケーラビリティは一般にコストパフォーマンスが悪い
  - 垂直・水平両方向のスケーラビリティについて対費用効果を考慮するべきである
- 一般にネットワークでは水平方向の方が垂直方向より効果的

スケーラビリティーを考慮したシステムプラニングのためには

- 水平方向のスケーラビリティの確保
  - Capacity Planning
  - Quantitative Approach
- オペレーションコストを減少させる努力
  - Intuition
  - Ad hoc procedure

スケーラビリティーを考慮したシステムプラニングのためには

- 各段階でスケーラビリティを意識する
  - 目標設定
  - システム設計
  - パフォーマンスの測定
  - クライアントの動きを分析
  - システムの負荷を分析
  - 性能向上のためのモデリング
  - モデルにおけるパラメタの抽出
  - システム負荷の低減
  - パフォーマンスの予想

最近の流行

- クラスタリング
  - 各社ワークステーション
  - LinuxなどPC UNIXでも
- WEB
  - 攻めのコンテンツ配布
    - Akamaizer
    - HydraGPS

Part4　まとめ1

- スケーラビリティの確保は今日のインターネットサーバーに欠かせない要素である
- システム構成の中で、どのくらいのスケーラビリティを持たせたいか検討する
- スケーラビリティの確保のためには、垂直方向と水平方向の2つの側面があるので、それぞれに応じた組み合わせを考える

まとめ2

- 正解はない。後からの評価で決まる
- 予測－実現－検証の絶え間ないサイクル
- ある種の「匠の技」が必要なことも