

キャッシュサーバ運用技術

Internet Week 2000 Tutorial @ Osaka

12/19/2000

鍋島 公章

nabe@kosho.org

目次

- 概論
- 構築と運用
- WWW以外への適用

Part 1 概論

- プロトコル概要
- キャッシュの基本
- プロトコル詳細
- キャッシュサーバのセキュリティ
- ヒット率の考察
- リバースプロキシ
- 透過型キャッシュ
- I-CAP
- ベンチマーク

プロトコル概要

- Hypertext Transfer Protocol (HTTP)
 - WWW用コンテンツ転送プロトコル
 - HTTP/0.9
 - HTTP/1.0
 - RFC1945
 - HTTP/1.1
 - RFC2068
 - ポート番号：80
- Uniform Resource Identifier (URI)
 - WWWにおけるコンテンツの識別名
- Hyper Text Markup Language (HTML)
 - WWWのコンテンツ記述言語

プロトコル概要 (基本1)

- メッセージ種類
 - 要求メッセージ
 - 応答メッセージ
- http://foo.com:80/index.html



Internet Week 2000 Tutorial (12/19/2000)

5

プロトコル概要 (基本2)

- メッセージフォーマット
 - リクエスト (メソッド+URI)/ステータス行
 - ヘッダ
 - 一般ヘッダ
 - 要求ヘッダ
 - 応答ヘッダ
 - エンティティヘッダ
 - メッセージボディ

Internet Week 2000 Tutorial (12/19/2000)

6

プロトコル概要 (要求メソッド)

- GET
 - オブジェクトの取得
- HEAD
 - オブジェクト情報(ヘッダ)の取得
- POST
 - 情報の引渡し
- OPTIONS
 - 使用可能なメソッドの一覧
- TRACE
 - ループバックチェック
- PUT
 - オブジェクトの保存
- DELETE
 - オブジェクトの削除
- PURGE (Squid拡張)
 - キャッシュ内オブジェクトの削除

プロトコル概要 (応答コード抜粋)

- 100番台：通知
- 200番台：処理成功
 - 200: 処理成功
- 300番台：転送要求
 - 301: オブジェクト移動(恒久)
 - 302: オブジェクト移動(臨時)
 - 304: オブジェクト無変更
- 400番台：クライアント側エラー
 - 400: リクエスト不正
 - 401: 認証不正
 - 403: アクセス権不正
 - 404: オブジェクトNot Found
 - 407: プロキシ認証必要
 - 412: マッチング条件失敗
- 500番台：サーバ側エラー
 - 500: サーバ内エラー
 - 503: サービスダウン
 - 504: ゲートウェイタイムアウト

プロトコル概要 (サンプル1)

- 要求メッセージ
 - GET / HTTP/1.1
 - Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint, */*
 - Accept-Language: ja
 - Accept-Encoding: gzip, deflate
 - User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows 98; Win 9x 4.90)
 - Host: cache.jp.apan.net
 - Connection: Keep-Alive

プロトコル概要 (サンプル2)

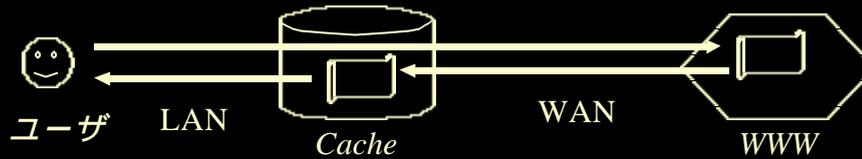
- 応答メッセージ
 - HTTP/1.1 200 OK
 - Date: Thu, 26 Oct 2000 16:51:49 GMT
 - Server: Apache/1.3.14 (Unix)
 - Last-Modified: Thu, 18 Mar 1999 05:31:05 GMT
 - ETag: "f012-491-36f08f99"
 - Accept-Ranges: bytes
 - Content-Length: 1169
 - Keep-Alive: timeout=15, max=100
 - Connection: Keep-Alive
 - Content-Type: text/html

 - <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
 - <html> <head>
 - <title>APAN Tokyo Root Server</title>

キャッシュの基本 (しくみ)

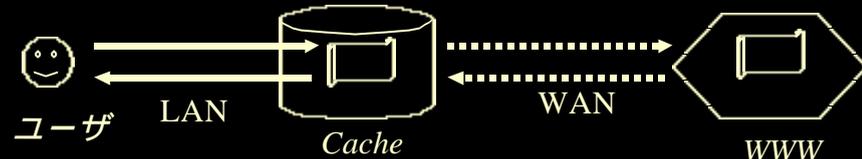
- リクエストの中継

- 中継したコンテンツを貯め込む



- 二回目以降のリクエスト

- 貯め込んだコンテンツを使う



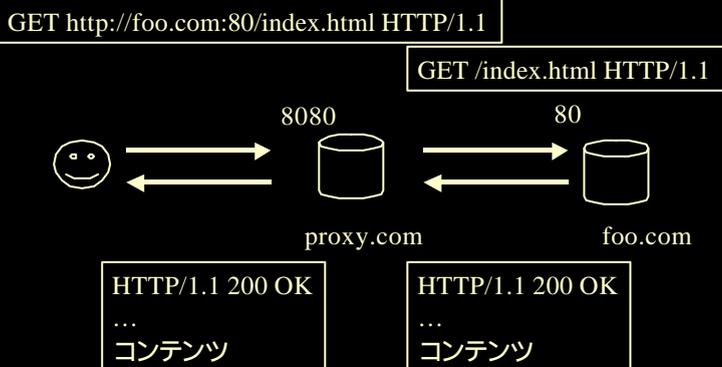
Internet Week 2000 Tutorial (12/19/2000)

11

キャッシュの基本 (しくみ)

- Proxyリクエストとダイレクト・リクエスト

- http://foo.com:80/index.html
- proxy.com:8080



Internet Week 2000 Tutorial (12/19/2000)

12

キャッシュの基本 (しくみ)

- 保存オブジェクト
 - オブジェクト+オブジェクト情報
 - 例 HTMLファイル+HTTPヘッダ情報

```
Date: Mon, 28 Aug 2000 07:30:21 GMT^M
Server: Apache/1.3.12 (Unix) (Red Hat/Linux) mod_perl/1.21^M
Last-Modified: Thu, 24 Aug 2000 06:10:35 GMT^M
ETag: "7ace1-30a-39a4bc5b"^M
Accept-Ranges: bytes^M
Content-Length: 778^M
Keep-Alive: timeout=15, max=100^M
Connection: Keep-Alive^M
Content-Type: text/html^M
^M
<HTML>
<HEAD> </HEAD>

<BODY>
```

Internet Week 2000 Tutorial (12/19/2000)

13

キャッシュの基本 (効用)

- 近くのキャッシュサーバ上にコンテンツがある
 - レスポンスの向上
- 同じコンテンツが、回線上を流れない
 - 回線の有効利用
 - トラフィックの抑制にはならない場合もある (レスポンスが向上した分、アクセスが増える)
- 同じリクエストが、WWWサーバに届かない
 - WWWサーバの負荷の低減



Internet Week 2000 Tutorial (12/19/2000)

14

キャッシュの基本 (実際)

- 回線の有効利用
 - インターラクティブなコンテンツの増加によるヒット率低下
 - 処理トラフィックの増加によるヒット率向上
- レスポンス向上
 - HTTPの平均速度は上がっている
 - 低速PPPユーザだと、キャッシュにヒットしてもレスポンス向上効果が少ない
 - WWW チャット, 掲示板サービス
 - リクエストの中継処理の分レスポンスが低下する
- キャッシュサーバの不正利用
 - 外部のユーザに勝手に使われる
 - JPCERT/CCへの報告多数

プロトコル詳細(キャッシュ関連ヘッダ)

- 要求ヘッダ
 - If-Modified-Since
 - Max-Forwards
 - Proxy-Authorization
 - If-Match
 - If-None-Match
 - If-Range
 - (X-Forwarded-for)
 - (Forwarded)
- 一般ヘッダ
 - Via
 - Pragma
 - Date
 - Cache-Control
- エンティティヘッダ
 - Expires
 - Last-Modified
 - ETag
- 応答ヘッダ
 - Age
 - (X-Cache)
 - (X-Cache-Lookup)

プロトコル詳細(Cache-controlヘッダ)

- Cache-control
 - 要求時
 - キャッシュ許可 (no-cache, no-store)
 - 有効期限 (max-age, max-stale, min-fresh)
 - リクエスト (only-if-cached)
 - 応答時
 - キャッシュ許可 (public, private, no-cache, no-store)
 - 変更許可 (no-transform)
 - 有効期限 (max-age)
 - 最新性のチェック (must-revalidate, proxy-revalidate)

プロトコル詳細(ヘッダの生成)

- HTML文書内でmetaタグ内に記述
 - HTTP-Equiv
 - `<meta http-equiv="cache-control" content="no-cache">`
 - 実際にタグは生成されない
 - ブラウザで処理される
 - プロキシでは無視される！
- .metaファイルの使用
 - Apache (CERN httpd準拠)
 - srm.conf
 - MetaFiles on
 - test.html .web/test.html.meta
 - Cache-Control: no-cache

プロトコル詳細 (1)

- 最新性関連 (応答)
 - Date (一般ヘッダ)
 - メッセージが生成された日時
 - Last-Modified (エンティティヘッダ)
 - オブジェクトの更新時間
 - Expires (エンティティヘッダ)
 - オブジェクトの有効期間
 - AGE (応答ヘッダ)
 - 無チェック時間
 - キャッシュされたオブジェクトの最新性を最後にチェックしてからの経過時間 (キャッシュ内で最新性のチェックなしで存在していた時間)
 - Cache-control : max-age (応答時)
 - 無チェック時間の最大値の指定

プロトコル詳細 (2)

- キャッシュされたオブジェクト使用時の最新性チェック指定
 - Cache-Control : proxy-revalidate (応答時)
 - プロキシでキャッシュされた場合のみ最新性をチェックする
 - Cache-Control : must-revalidate (応答時)
 - 必ず最新性をチェックする (ブラウザ内キャッシュの場合もチェック)
- 最新性チェック要求
 - HEADメソッド
 - オブジェクトの情報 (主に、Last-Modified、ETagヘッダ) だけを取得
 - If-Modified-Since (要求ヘッダ)
 - 最新情報取得リクエスト
 - GETメソッド + If-Modified-Sinceヘッダ (IMSリクエスト)
 - オブジェクト (200)
 - 変更無し (304)

プロトコル詳細 (3)

- 最新性関連 (要求時)
 - Cache-Control : max-age (要求時)
 - 無チェック時間が指定時間以内のオブジェクトを受け付ける
 - Cache-Control : max-stale (要求時)
 - Expireしてから指定時間以内のオブジェクトを受け付ける
 - Cache-Control : min-fresh (要求時)
 - 最新性が指定時間以上保持されるオブジェクトを受け付ける

プロトコル詳細 (4)

- エンティティタグ関連
 - ETag (エンティティヘッダ)
 - エンティティのバージョン毎にユニークな値を持つ
 - If-Match (要求ヘッダ)
 - Etagが同じ場合、メソッドを実行
 - 異なる場合、412 (返答コード) が返る
 - If-None-Match (要求ヘッダ)
 - Etagが異なる場合、メソッドを実行
 - 同じ場合、412 (返答コード) が返る
 - If-Range (要求ヘッダ)
 - Etagが同じ場合、Range指定した部分エンティティを返す
 - Etagが異なる場合、全エンティティを返す

プロトコル詳細 (5)

- キャッシュ不可オブジェクト
 - ユーザ認証付オブジェクト
 - クッキー付オブジェクト
- キャッシュ制御
 - Cache-control : no-cache (要求、応答時)
 - キャッシュ禁止
 - Cache-control : no-store (要求、応答時)
 - 一時的なコピーも禁止
 - Cache-control : public (応答時)
 - パブリックコンテンツ、共用キャッシュ可能
 - Cache-control : private (応答時)
 - プライベートコンテンツ、プライベートキャッシュ可能

プロトコル詳細 (6)

- キャッシュ制御
 - Pragma: no-cache (一般ヘッダ、HTTP/1.0)
 - 要求：
 - 必ずWWWサーバからオブジェクトを取得
 - 応答：
 - キャッシュに保存禁止

プロトコル詳細 (7)

- ブラウザのリロードの実装
 - Netscape Communicator 4.75
 - Reload: IMS + Pragma: no-cache
 - Shift + Reload: Pragma: no-cache
 - Microsoft Internet Explorer 5.5
 - Proxyなし
 - Reload: IMS
 - Control + Reload: 新規GET
 - Proxyあり
 - Reload: IMS + Pragma: no-cache
 - Control + Reload: Pragma: no-cache
 - (IMS = If-Modified-Since)

プロトコル詳細 (8)

- Proxy中継
 - Via(一般ヘッダ)
 - 中継プロキシ名
 - 例: Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1)
 - Max-Forwards(要求ヘッダ)
 - TRACEメソッド使用時の最大中継数
 - TRACEメソッド
 - 最後にリクエストを受けたサーバがリクエストに付けられたヘッダを返す
 - X-Forwarded-for, Forwarded (要求ヘッダ)
 - アクセス元IPアドレス

プロトコル詳細 (9)

- その他
 - Cache-control : no-transform (応答時)
 - オブジェクトの変更不可
 - Cache-control : only-if-cached (要求時)
 - キャッシュにヒットした場合のみオブジェクトを返す
 - Cache-control : proxy-authorization (要求時)
 - プロキシ認証コード
- Cache Digest用 (Squid拡張)
 - X-Cache (応答時)
 - オブジェクトがキャッシュから取り出されたか
 - X-Cache-Lookup (応答時)
 - キャッシュがオブジェクトを持っているか

Internet Week 2000 Tutorial (12/19/2000)

27

キャッシュサーバのセキュリティ(1)

- WWWはインタラクティブに
 - もはや、情報を受信するだけではない
 - 商取引
 - WWWショッピング
 - 偽造・盗難クレジットカードの使用
 - コミュニケーション
 - E-Mail, NetNewsの送受信
 - 掲示板, チャットサービス
 - いやがらせ, スパミング, ...
 - ポートスキャンの踏台
- 発信者のIPアドレス = ユーザ特定の重要な方法

Internet Week 2000 Tutorial (12/19/2000)

28

キャッシュサーバのセキュリティ(2)

- 匿名装置

- キャッシュサーバ
 - ユーザの代理人としてリクエストを発行する
 - サーバ側にはキャッシュサーバのアドレスが残る
- 匿名装置としてのキャッシュサーバの使用
 - 他の組織のキャッシュを使いプライバシーを守る
 - インターネットの常識?



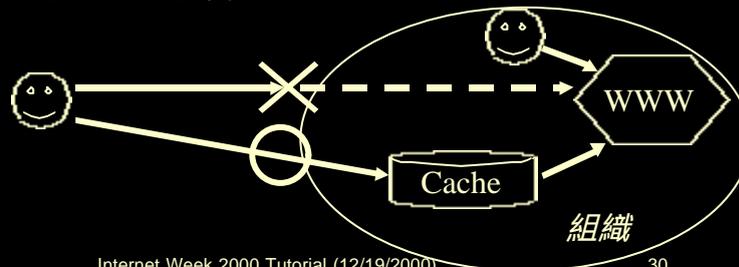
Internet Week 2000 Tutorial (12/19/2000)

29

キャッシュサーバのセキュリティ(3)

- セキュリティホール

- ファイアウォールとしてのキャッシュサーバ
- アクセス制限のミス
 - 外部から内部に入るためのセキュリティホール
 - イン트라ネット上の社内情報にアクセス可能
- 特権ポート以外は外部からのアクセスを許すサイトの場合、一台のキャッシュサーバが命取り



Internet Week 2000 Tutorial (12/19/2000)

30

キャッシュサーバのセキュリティ(4)

- 外部からアクセス可能なキャッシュサーバを見つけ出すのは簡単
 - ポートスキャン
 - キャッシュサーバ探し専用ツール
 - WWWのアクセスログから特定
 - Viaヘッダ等によりキャッシュサーバの情報はWWWサーバのログに残る
 - 公開状態のキャッシュサーバのリストも出回る
- ユーザの意識は低い
 - 「インターネットはオープン」という言葉の誤解
 - アクセス可能なキャッシュサーバは誰でも使って良い、と思っている

キャッシュサーバのセキュリティ(5)

- 対策
 - 組織外部からのリクエストを拒否する
 - 一部プロダクトでは、キャッシュサーバでアクセス制限不可
 - ルータでアクセス制限する
 - Apache serverのProxy Moduleも注意
 - 80ポートでProxyが動いている可能性がある
 - Reverse Proxy専用マシンも注意
 - Forwarding Proxyとして動いている可能性がある
 - Proxy Checker
 - 外部からのアクセス制限のチェック
 - <http://cache.jp.apan.net/proxy-checker/>
 - アクセスログを保管する

キャッシュサーバのセキュリティ(6)

- デフォルトのSquid.conf (ACL定義)
 - `acl all src 0.0.0.0/0.0.0.0`
 - `acl manager proto cache_object`
 - `acl localhost src 127.0.0.1/255.255.255.255`
 - `acl SSL_ports port 443 563`
 - `acl Safe_ports port 80 21 443 563 70 210 1025-65535`
 - `acl Safe_ports port 280 # http-mgmt`
 - `acl Safe_ports port 488 # gss-http`
 - `acl Safe_ports port 591 # filemaker`
 - `acl Safe_ports port 777 # multiling http`
 - `acl CONNECT method CONNECT`

キャッシュサーバのセキュリティ(7)

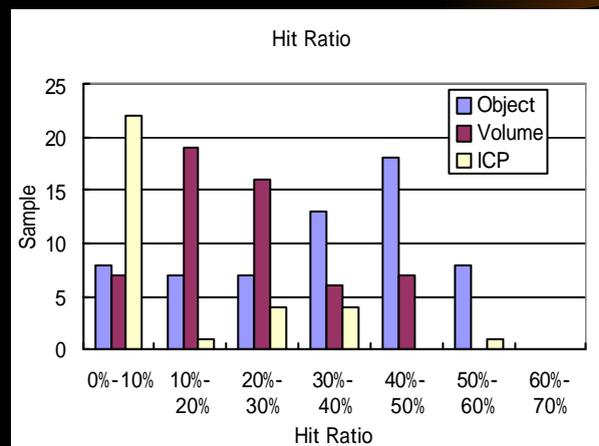
- デフォルトのSquid.conf (アクセス権定義)
 - `http_access allow manager localhost`
 - `http_access deny manager`
 - `http_access deny !Safe_ports`
 - `http_access deny CONNECT !SSL_ports`
 - `http_access deny all`
 - `icp_access allow all`
 - `miss_access allow all`

ヒット率の考察(1)

- アンケート形式のヒット率調査
 - 調査期間
 - 98年11月～12月
 - 報告者数
 - 44
 - キャッシュサーバ数
 - 61
 - <http://cache.jp.apan.net/survey98/>

ヒット率の考察(2)

- 結果



ヒット率の考察(3)

- ヒット率
 - Object (リクエスト) ベース: 40%程度が中心
 - Volume (トラフィック) ベース: 20%程度が中心
 - IJ4U 36%, 17% (700 Kbps), 某ISP 30%, 30%
- 傾向
 - ユーザの種類, クライアントの種類によってヒット率の傾向は異なる
 - ユーザの種類 (企業, 学校, ISP)
 - 企業 > 学校 > ISP
 - クライアントの種類 (ブラウザ, 他のキャッシュ, 混合)
 - ブラウザ > 他のキャッシュ

ヒット率の考察(4)

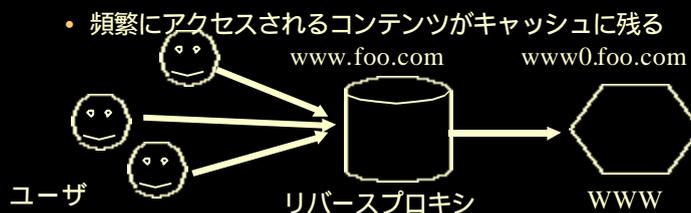
- アンケート結果の内訳
 - 使用ソフトウェアはSquid
 - もはやベストプロダクトではない
 - 最大100万リクエスト/日 (数M bps程度)
- 最近の製品を使うとヒット率は高くなる
 - 処理トラフィック (処理トラフィックが増えるとヒット率は向上する)
 - 10Mbps以上の処理が可能
 - ヒット率を高める機構

ヒット率の考察(5)

- ヒット率を高める機構
 - アルゴリズムの改良
 - コンテンツの置換えアルゴリズム
 - ユーザのReloadリクエストへの対応
 - コンテンツの自動更新
 - 人気コンテンツの定期的アップデート

リバースプロキシ (基本1)

- 特定のWWWサーバ用のキャッシュ
 - WWWサーバの代理にリクエストを受け付ける



- 役割
 - WWWサーバの高速化
 - 軽量ミラーサーバ

リバースプロキシ (高速化)

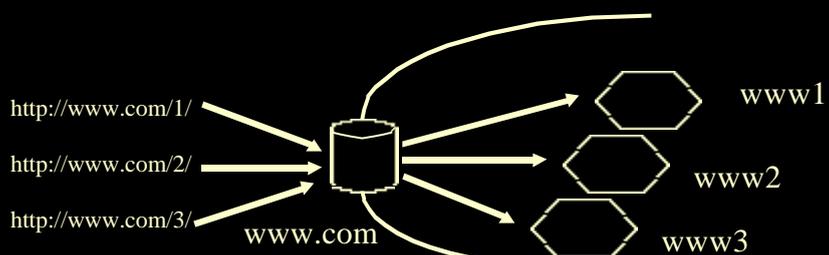
- WWWサーバの高速化
 - WWWサーバでも同様の機構を実現可能だが、現状では、リバースプロキシを使うのが一般的
 - WWWサーバの一つのボトルネックはディスク処理
 - 人気コンテンツをメモリ上に持つ
 - ディスクIO処理の回数が減る
 - 人気コンテンツへのリクエストが高速化される
 - リクエスト処理の高速化
 - シングルスレッド等
 - 動的なコンテンツ (データベースアクセスCGI等) のキャッシュ
 - システム設計に注意が必要
 - サーバ負荷を大幅に低減

リバースプロキシ (軽量ミラーサーバ)

- 一種のミラーサーバ
 - 頻繁にアクセスされるコンテンツだけが、リバースプロキシ上にキャッシュされる
 - 頻繁にリクエストされるコンテンツは全体の一部
 - 必要なディスク領域の節約
 - WWWサーバからの情報の更新は必要ない
 - コンテンツはユーザのリクエストにより自動的にキャッシュ上に貯まる
 - ユーザのReloadリクエスト等により自動的に更新される
 - 発展形として、WWWサーバから更新情報をリバースプロキシにプッシュする機構もある
 - ディストリビューションサービス

リバースプロキシ (その他用途)

- 内部サーバの保護
 - ファイアーウォール上にリバースプロキシを配置
 - 外部からのHTTPリクエストだけを、内部のWWWサーバに中継
- 複数のサーバを一台にまとめる
 - URL書換え



Internet Week 2000 Tutorial (12/19/2000)

43

透過型キャッシュ (しくみ)

- Proxy型(従来型)
 - ユーザは、明示的にキャッシュサーバを設定

ブラウザ設定:
Proxy Server:
cache.foo.com:3128



キャッシュ



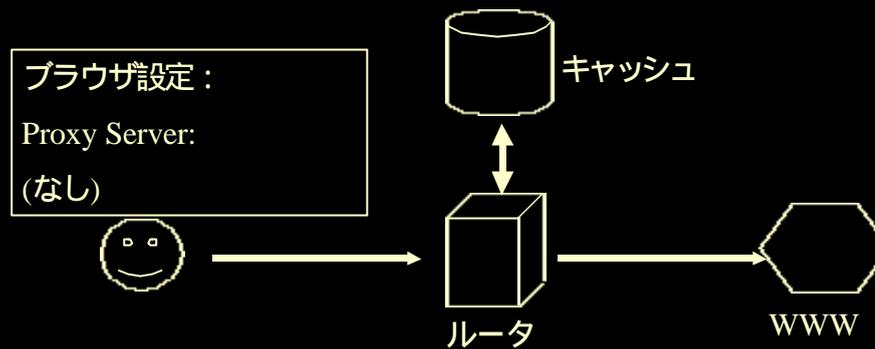
WWW

Internet Week 2000 Tutorial (12/19/2000)

44

透過型キャッシュ(しくみ)

- 透過型
 - ルータ等が強制的に80番向けパケットをキャッシュに送り込む(リクエストのハイジャック)
 - リバースプロキシとしての利用も広まる



Internet Week 2000 Tutorial (12/19/2000)

45

透過型キャッシュ(利点)

- ユーザの設定が不必要
- 全トラフィックをキャッシュ可能
 - トラフィック抑制効果が大きい
 - 通常だと、使わせ方に悩む
- キャッシュしたいものだけ選別可能
 - 経路別
 - 国際線(リンクコスト高い)はキャッシュする
 - 国内線(リンクコスト安い)はキャッシュしない
 - オブジェクト別
 - イメージファイルはキャッシュする
 - HTMLファイルはキャッシュしない

Internet Week 2000 Tutorial (12/19/2000)

46

透過型キャッシュ (問題点)

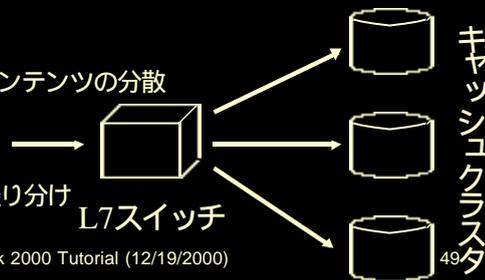
- ユーザは強制的にキャッシュを使用させられる
 - 安定性, プライバシがより重要に
- 特別なルーティング
 - ルータの負荷が上がる
- 強制的に多段キャッシュとなる可能性
 - 最適な運用形態
- 80番ポートを使うWWW以外のアプリケーション
 - 世の中には変なシステムが存在する
- マルチホーム時のトラフィック操作が難しくなる
 - WWWリクエストのソースアドレスが書き換わる

透過型キャッシュ (現状)

- 米国
 - 段々人気が無くなってきた
 - 特に大手ISP
 - あまり効果が上がらない?
- 日本
 - ユーザサイトでは, ぼちぼち導入済み
 - ISPによる使用も始まった
 - 大手ISPも導入直前 (と言われて、はや2年...)
- iDC
 - リバースプロキシとしての利用

透過型キャッシュ(L7スイッチ)

- Layer7スイッチ
 - プロトコル内部の情報(Layer7)による振り分け
 - Layer4スイッチ
 - Layer4 (TCP) コネクションの振り分け・変換
- キャッシュサーバ単体でも透過型が可能
 - 比較的大規模な構成ではL4-7スイッチの使用が主流
- L7スイッチの機能
 - クラスタリング
 - ハッシュを使ったコンテンツの分散
 - フェイルオーバー
 - 透過型キャッシュ機能
 - オブジェクト名による振り分け



Internet Week 2000 Tutorial (12/19/2000)

I-CAP(1)

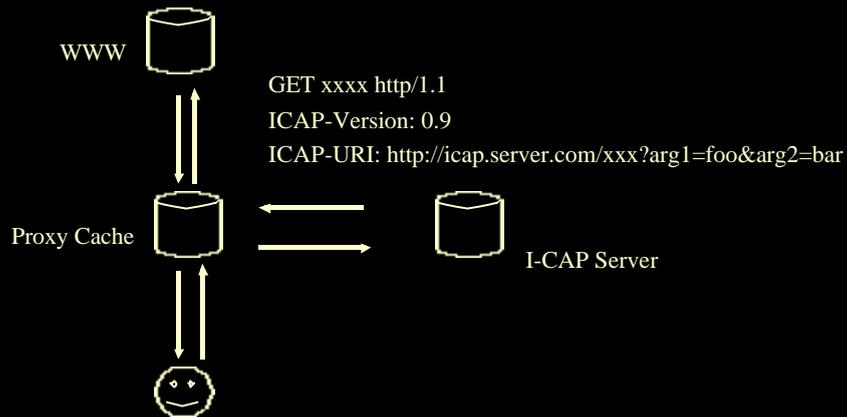
- Internet Content Adaptation Protocol (I-CAP)
 - プロキシの機能拡張
 - 翻訳、デバイス別、フィルタリング、圧縮等
 - 拡張機能サーバ(I-CAPサーバ)とプロキシサーバの標準プロトコル
 - 機能呼び出しのプロトコルとしてHTTP準拠のプロトコルを使用
 - モード
 - Request Modification
 - Request Satisfaction
 - Response Modification

Internet Week 2000 Tutorial (12/19/2000)

50

I-CAP (2)

- 例



Internet Week 2000 Tutorial (12/19/2000)

51

ベンチマーク

- WWWサーバ用のベンチマークは使えない
 - 小数のデータに対するリクエスト
 - リクエストの局所性
 - ヒット率が高くなり、あまりディスクを使わない
- キャッシュサーバ用のベンチマーク
 - 適度なヒット率を保つリクエストパターンを生成
- Web Polygraph (IRCACHE)
 - 複数のクライアントとサーバ
 - 定期的にベンチマーク大会を開催
 - ただし、主要ベンダは参加せず
 - 毎回、コストパフォーマンスが向上

Internet Week 2000 Tutorial (12/19/2000)

52

Part 2 構築と運用

- 運用心得
- 運用トレンド
- 協調運用
- Squid + FreeBSD
- ハードウェアの選定
- OSのインストール
- コンパイル
- Squid.conf
- 実行準備
- アクセスLOG
- 監視
- ユーザへの使いわせ方
- 協調運用
- クラスタリング
- リバースプロキシ
- 実システム

運用心得

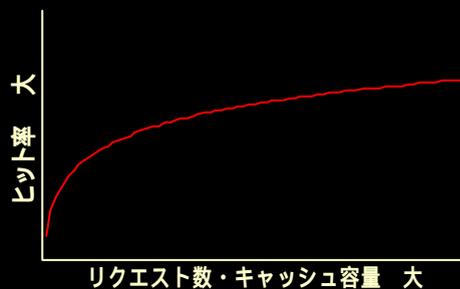
- ユーザリクエストの中継
 - ルータの管理と同じ
- プライバシーの保護
 - ユーザのプライバシー(アクセス履歴)を握る

運用トレンド

- アクセス制限は必須
 - 悪用の広がり
- キャッシュサーバの協調運用は難しい
 - 協調関係は最小にする
- サーバ負荷に注意
 - 高負荷のキャッシュはレイテンシを増加させる
- フリーソフト以外のプロダクトが低価格化
 - ある程度以上のトラフィックになるとSquidは管理コストが高い

協調運用 (ヒット率の上げ方)

- キャッシュが処理するリクエストを増やす
- キャッシュ領域を増やす
- ただし, 対数に比例



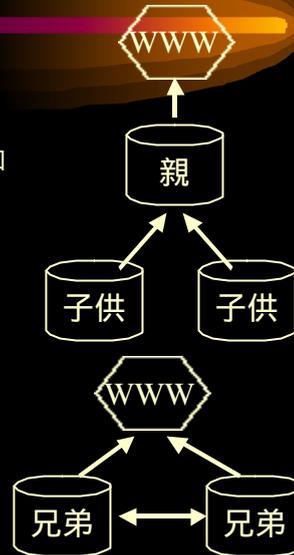
協調運用(関係)

- 親子関係

- 子供はミスした全リクエストを親に送る
 - 親キャッシュで処理するリクエストが増加
 - 親キャッシュのヒット率が上がる
 - 系全体のヒット率が上がる

- 兄弟関係

- コンテンツがある場合のみ、リクエストを送る
 - 見かけ上のキャッシュ領域が増加
 - ヒット率が上がる

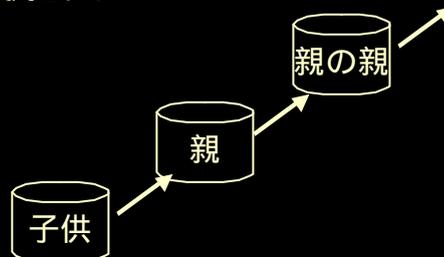


Internet Week 2000 Tutorial (12/19/2000)

57

協調運用(親子関係)

- 多段接続は避ける
 - 1つのキャッシュを経由する毎に0.5秒程度 + ICP待ち時間(キャッシュ間のRTT+ICP処理時間)の時間が余計にかかる
 - 3段程度 / 約2秒が限度
- 速い経路を持つキャッシュが有効
 - 親キャッシュの経路が使用される
- ICPを使うのが無難
 - 親のダウンを検出



Internet Week 2000 Tutorial (12/19/2000)

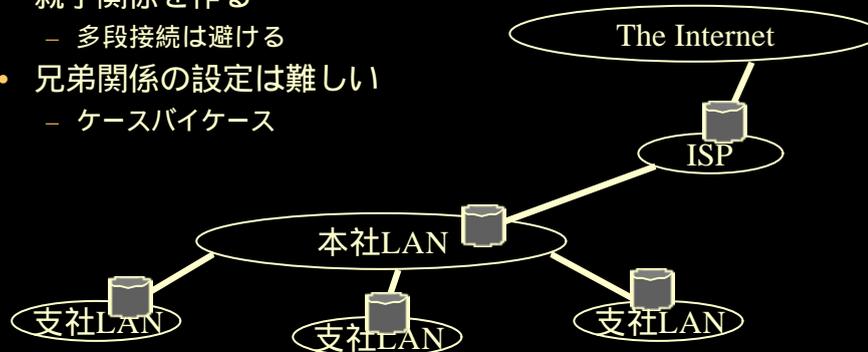
58

協調運用 (兄弟関係)

- 下手な協調関係はレスポンスを低下させる
 - 効果のある兄弟サーバを選ぶ
 - RTTが小さい(最重要)
 - 一番大きいRTTに引きずられる
 - 太い線で接続されている
 - ヒット率が高い

協調運用 (キャッシュの配置)

- 対外リンクの手前に1システムが基本
 - サイト(同一LAN内)に1システム
 - 組織単位等では置かない
- 親子関係を作る
 - 多段接続は避ける
- 兄弟関係の設定は難しい
 - ケースバイケース

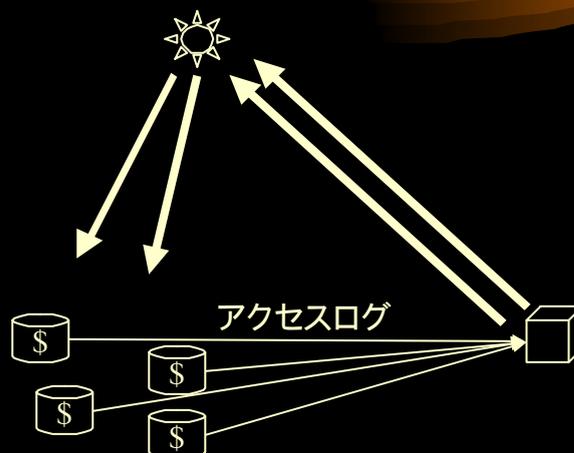


協調運用（衛星システムの利用1）

- ヒット率を上げる方法
 - 処理トラフィックを増やす
 - オブジェクト保持時間が短くなる、しかし、ホットコンテンツがキャッシュされる効果の方が大きい
 - 中小サイト(ISP)では難しい
 - 衛星経由でホットコンテンツをブロードキャスト
 - 参加サイトのLOGを収集
 - アクセスされたオブジェクトを衛星経由で各参加サイトへブロードキャスト（実際にはマルチキャストプロトコルを使用）
 - 仮想的に処理トラフィックを増やしたのと同等の効果
 - 効果
 - 10～20%程度のヒット率向上（パンフレットより）

協調運用（衛星システムの利用2）

- 概要



Squid + FreeBSD

- ターゲット
 - 1M bps程度まで,
 - 使い古したPC+FreeBSD+Squidで十分対応可能
 - 256 MB程度のメモリは載せる
 - 5M bps程度
 - 30万円程度のPC
 - 512MB程度のメモリ
 - ディスクに投資
 - 15M bps以上
 - 単体のSquid + FreeBSDでは荷が重い
 - クラスタリング

ハードウェアの選定(1)

- ディスク性能が一番効く
 - 高性能ディスクを複数台接続
 - 出来れば一万回転以上の高性能SCSI
 - UDMA 66 IDEディスク
 - SCSIの場合6台程度までは向上する
 - あまりに多いとSCSIバスが飽和する
 - シーク待ち時間を減らす
 - RAIDの場合 RAID-0 (ストライピング)

ハードウェアの選定(2)

- できるだけ多くのメモリをのせる
 - 20GBのオブジェクト=1M個程度のオブジェクト
 - オブジェクトのインデックスサイズ=50MB程度
 - オンメモリキャッシュ領域として使う
 - ディスクIOを減らす
- CPUはそれほど重要ではない
 - そこそこのCPUで十分
 - Squidの場合, 基本的にマルチCPUは意味がない
 - マルチスレッド化
 - diskd (2.4.Devel)

OSのインストール(パーティション)

- パーティション
 - キャッシュ領域は別パーティションにする
 - noatime オプション付きでマウント
 - アクセス時間を書き込まない
 - LOG領域は大き目にする
 - スワップ領域を大きくしても意味がない
 - スワップを使い始めるとSquidの性能は著しく低下する

OSのインストール(Log領域1)

- Logファイルの大きさ

- access.log
 - ICP + TCPリクエスト
 - 130 byte / 行 程度
 - gzip -9
 - 1/5程度に圧縮可能
 - 例: 10万リクエスト
 - 生log: 13M
 - 圧縮log: 2.5M
- store.log
 - オブジェクトのディスクIOのlog
 - 通常, 不必要
 - access.logと同程度
- cache.log
 - エラーメッセージ
 - 通常, 無視出来る大きさ
 - エラー時に増えるが大したことはない

OSのインストール(Log領域2)

- Logファイル用領域

- 一日に必要な量の最低5倍程度の領域を確保する
 - logの保存失敗
 - 急なリクエストの増加
- 例: 10万リクエスト/日 (access.logのみ)
 - store.logを残す場合, 同程度の領域がさらに必要
 - 10万行 * 130 = 約 13 MB/日
 - Log領域: 60 MB程度/日を用意する
 - 保存用: 2.5 MB程度/日

OSのインストール(時計)

- 時計あわせ
 - オブジェクトの新鮮さの判断
 - (オブジェクトの取得時刻-オブジェクトの生成時刻) *
 - 時計の遅れ
 - オブジェクトがキャッシュされない
 - 未来のファイルをキャッシュ
 - 時計の進み
 - 古いオブジェクトを返す
 - NTP等を設定する

OSのインストール(カーネル 1)

- 使用可能リソースを増やす
 - プロセスあたりの使用可能ファイルディスクリプタ数
 - MAXFILES ($2 * (20 + 16 * \text{MAXUSERS})$)
 - デフォルト 1064
 - リクエストの中継には3つのファイルディスクリプタを使用
 - 同時に300コネクション程度しか処理できない
 - ディスクリプタを使い切ると、レスポンスが急激に低下
 - コネクション開始の待ち行列に入る
 - MBUF領域
 - NMBCLUSTERS ($512 + \text{MAXUSERS} * 16$)
 - デフォルト 1024
 - 処理するコネクション数が増えると間に合わない
 - OSが落ちる

OSのインストール(カーネル2)

- カーネルのコンフィギュレーションファイルの編集
 - cp /sys/i386/conf/GENERIC CACHE
 - maxusers 32 -> 256
 - options MAXFILES = 8232
 - options NMBCLUSTERS = 10240
- カーネルのコンパイル
 - config CACHE
 - cd ../../compile/CACHE
 - make clean
 - make depend
 - make
 - cp kernel /kernel

OSのインストール(カーネル3)

- Squid-2.4 (Development Versions)
 - Diskd
 - ディレクトリ単位に管理プロセス
 - マルチプロセッサの効果あり
 - 第二回Bake-offではパフォーマンスが4倍まで向上
 - async-ioの発展形
- Linux
 - ReiserFS
 - 高速
 - B-ツリーによるオブジェクト割当

コンパイル

- コンパイル
 - tar -xzf squid-2.2.STABLE5-src.tar.gz
 - cd squid-2.2.STABLE5
 - ./configure
 - make
 - make install

Squid.conf (アクセス制限 1)

- Squid.conf (デフォルトクラス定義)
 - acl all src 0.0.0.0/0.0.0.0
 - acl manager proto cache_object
 - acl localhost src 127.0.0.1/255.255.255.255
 - acl SSL_ports port 443 563
 - acl Safe_ports port 80 21 443 563 70 210 1025-65535
 - acl Safe_ports port 280 # http-mgmt
 - acl Safe_ports port 488 # gss-http
 - acl Safe_ports port 591 # filemaker
 - acl Safe_ports port 777 # multiling http
 - acl CONNECT method CONNECT

Squid.conf (アクセス制限 2)

- Squid.conf(デフォルトアクセス制限)
 - `http_access allow manager localhost`
 - `http_access deny manager`
 - `http_access deny !Safe_ports`
 - `http_access deny CONNECT !SSL_ports`
 - `http_access deny all`
 - `icp_access allow all`
 - `miss_access allow all`

Squid.conf (アクセス制限 3)

- クラス設定
 - `acl users src 127.0.0.1/255.255.255 123.456.0.0/255.255.0.0`
- リクエスト制限
 - `http_access deny !users`
- コメントアウト可能
 - `http_access deny !Safe_ports`
 - ポートスキャン, 不正利用の禁止

Squid.conf(キャッシュ領域)

- ディスクキャッシュの領域の設定
 - 実際のパーティションの大きさの8割程度を割振る
 - Cache_dir /usr/local/squid/cache 800 16 256
- メモリキャッシュの領域の設定
 - スワップを使わない程度に大きくする
 - 一般には実メモリの1/3程度
 - 例: ATRC 実メモリ 512MB キャッシュメモリ 192MB
 - 100MB程度のFreeメモリ
 - cache_mem 80 MB
- 最大オブジェクトサイズ
 - 例: maximum_object_size 128000

Squid.conf(その他必須項目)

- Mailアドレス (Mail aliasも設定する)
 - ftp_user squid@your.cache.com
 - cache_mgr root@your.cache.com
- DNSキャッシュクライアント数
 - dns_children 5 -> 32

Squid.conf(ヒント1)

- 負荷を下げる
 - abortしたリクエストを強制終了させる
 - quick_abort_(min|max|pct)
 - 最大オブジェクトサイズを小さくする
 - リクエストあたりのディスクIOを減らす
 - maximum_object_size
- ヒット率を上げる
 - オブジェクトの新鮮さの判断を甘くする
 - refresh_pattern
 - 強制ReloadをIMSに書換え
 - reload_into_ims on

Squid.conf(ヒント2)

- セキュリティ
 - リクエストの中継情報を非表示
 - 内部ホストの情報を外部に出さない
 - forwarded_for off
- ファイアーウォール内キャッシュ
 - 必ずファイアーウォール経由でアクセス
 - acl local dstdomain your.com
 - always_direct allow local
 - never_direct allow all
 - cache_peer no-query

Squid.conf(ヒント3)

- 最短経路の選択

- --enable-icmp (コンパイル時)
- query_icmp on
- make install-pinger
- NetDB
 - WWWサーバまでのRTT
 - リクエスト時にICMPをWWWサーバに送る
 - RTTの足し込み(親キャッシュ経由)



実行準備

- ディレクトリ設定

- Ownerをnobody.nogrupに変更
 - /usr/local/squid/log
 - /usr/local/squid/cache
- キャッシュ用領域の初期化
 - /usr/local/squid/bin/squid -z

- 実行ファイル

- /usr/local/etc/rc.d/squid.sh
 - /usr/local/squid/bin/RunCache &

アクセスLog(解析, 保存)

- Calamari
 - オプション
 - -a 詳細な表示
 - -r -1 全クライアント情報表示
 - スクリプト
 - \$\$SQUID/bin/squid -k rotate
 - sleep 120
 - mv \$\$SQUID/log/cache.log.0 \$ARC/cache.\$DAY
 - calamari < \$ARC/cache.\$DAY > \$STAT/stat.\$DAY

アクセスLog(フォーマット)

- フォーマット
時刻 転送時間 クライアントアドレス
キャッシュステータス/httpステータス オブジェクトサイズ
HTTPメソッド URL ユーザ名(ident)
協調ステータス/協調ホスト mime

942289905.074 537 129.60.215.113
TCP_MISS/200 1949
GET http://foo.com/index.html -
FIRST_PARENT_MISS/cache00.jp.apan.net text/html

アクセスLog(キャッシュステータス1)

- 基本
 - TCP_HIT
 - 通常のヒット
 - TCP_MISS
 - 通常ミス
 - TCP_MEM_HIT
 - オンメモリキャッシュにヒット
 - TCP_DENIED
 - アクセス制限違反
 - TCP_NEGATIVE_HIT
 - エラーステータスのキャッシュにヒット
 - HTTP not found (404)等
 - TCP_SWAPFAIL
 - ヒットしたが、実際にはキャッシュ中に存在しない(システムエラー)
 - ERR_CLIENT_ABORT
 - クライアントがリクエストを中断
 - ERR_NO_CLIENTS
 - クライアントが強制終了
 - ERR_READ_ERROR
 - WWWへのリクエストの読み込みエラー
 - ERR_CONNECT_FAIL
 - WWWへコネクションが開けない

Internet Week 2000 Tutorial (12/19/2000)

85

アクセスLog(キャッシュステータス2)

- Squidによる最新性のチェック(IMSリクエストの発行)
 - キャッシュにヒット, しかしオブジェクトが長期間キャッシュに存在
 - (オブジェクト取得時間 - オブジェクト生成時間) *
 - TCP_REFRESH_HIT
 - キャッシュ中オブジェクトは最新
 - キャッシュ中のオブジェクトを返す
 - TCP_REFRESH_MIS
 - オブジェクトをWWWサーバから取得
 - 最新のオブジェクトを返す
 - TCP_REF_FAIL_HIT
 - IMSリクエスト失敗
 - キャッシュ中のオブジェクトを返す

Internet Week 2000 Tutorial (12/19/2000)

86

アクセスLog(キャッシュステータス3)

- クライアントによる最新性のチェック
 - TCP_CLIENT_REFRESH
 - クライアントがno-cache命令を発行
 - 常にWWWからオブジェクトを取得
 - TCP_IMS_HIT
 - クライアントがIMS命令を発行, しかし, キャッシュ中のオブジェクトは最新
 - Not Modifiedを返す
 - TCP_IMS_MISS
 - クライアントがIMS命令を発行, キャッシュ中のオブジェクトが古い事が判明
 - 最新のオブジェクトを返す

Internet Week 2000 Tutorial (12/19/2000)

87

アクセスLog(協調テイタス 1)

- 基本
 - NONE
 - キャッシュにヒット (外部アクセスなし)
 - DIRECT
 - WWWサーバに直接アクセス
 - TIMEOUT_DIRECT
 - ICPのタイムアウトによりWWWサーバに直接アクセス
 - PARENT_HIT
 - 親キャッシュでヒット
 - PARENT_MISS
 - 親キャッシュでミス
 - SIBLING_HIT
 - 兄弟キャッシュでヒット

Internet Week 2000 Tutorial (12/19/2000)

88

アクセスLog(協調テイタス 2)

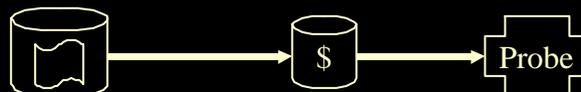
- 最短経路
 - RTT情報なし、configファイル中の最初(FIRST)の親キャッシュ
 - FIRST_PARENT_MISS
 - FIRST_UP_PARENT
 - RTT情報によるアクセス選択
 - SOURCE_FASTEST (ICPIによりダイレクトを選択)
 - CLOSEST_DIRECT (ICMPIによりダイレクトを選択)
 - CLOSEST_PARENT_MISS
 - CLOSEST_PARENT
- その他
 - DEFAULT_PARENT
 - ROUNDROBIN_PARENT

Internet Week 2000 Tutorial (12/19/2000)

89

監視(1)

- リソース不足
 - リクエストの中継時間が長くなる
 - 監視
 - 特定のファイルを取り出す時間を計測する
 - pl-checker (proxy latency checker)
 - 中継速度の計測
 - timeoutすると管理者にMailで連絡
 - 例: pl-checker proxy-server 3128 http://target-url 10 nabe@kosho.org



Internet Week 2000 Tutorial (12/19/2000)

90

監視(2)

- Multi Router Traffic Grapher (MRTG)
 - 本来は、Routerの処理トラフィックを表示するWWWページ作成ツール
 - Squidにも使える

ユーザへの使わせ方(1)

- キャッシュサーバを積極的に使うユーザの割合は減少傾向
 - キャッシュを使っても、あまりレスポンスは上がらない
 - 特に低速PPPユーザ
 - Web Chatや掲示板ではレスポンスが下がる
 - 第三者にリクエストのLOGを管理される
 - 嫌がるユーザも多い

ユーザへの使わせ方(2)

- 強制的に使わせる
 - 80ポートへのリクエストをブロックする
 - キャッシュサーバを使わないと、外部のWWWへアクセス不可能
 - 大学等では行われている
 - 透過型キャッシュ
 - FreeBSD単体
 - IF Filter + Squid ipf-transparentオプション
 - Cisco
 - WCCP 1.0
 - Port Redirection

ユーザへの使わせ方(3)

- PAC (Proxy Auto Config)
 - プロキシサーバの自動設定
 - proxyサーバの情報をJavascriptで記述
 - ドメイン, IPアドレス等によるプロキシサーバの選択
 - バックアップサーバの指定

ユーザへの使わせ方(4)

- WPAD (Web Proxy Auto-Discovery protocol)
 - PACファイルのあるURLの自動認識
 - 使用プロトコル
 - Dynamic Host Configuration Protocol (DHCP)
 - Service Location Protocol (SLP)
 - Well-known ホスト名(WPAD)
 - `http://wpad.your.domain/wpad.dat`
 - DNS レコード (SRV TXT)
 - IE50以上で使用可能
 - LANの設定オプション

協調運用

- 基本設定
 - 親子
 - `cache_peer` 親キャッシュ `parent` TCP-port ICP-port
 - 兄弟
 - `cache_peer` 兄弟キャッシュ `sibling` TCP-port ICP-port
- 詳細設定
 - `cache_peer_domain`
 - `cache_peer_access`
 - `neighbor_type_domain`

クラスタリング

- 一台のサーバではパフォーマンス不足
 - マシンを増やすのが低コスト
 - 2台にすれば, 処理性能は2倍
 - ラウンドロビン
 - DNS
 - PACファイルの利用
 - コンテンツの分散を行うとヒット率の向上が見込める
 - ハッシュによるコンテンツの分散
 - ハッシュを使わないコンテンツの分散

クラスタリング (ハッシュ使用)

- クライアント - サーバ間
 - PACファイルの利用
 - Super Proxy Script
 - $proxyIndex = \text{Mod}(\text{Checksum}(\text{URL}), N)$
 - L4-7スイッチの利用
 - 仮想サーバの設定
 - リクエストの振り分け
- サーバ - サーバ間
 - CARP (Cache Array Routing Protocol)
 - ハッシュ関数によるURLのグループ分け
 - キャッシュサーバ間のコンテンツの分散
 - Proxy Array Membership Table
 - キャッシュサーバのグループの定義

クラスタリング(ハッシュ不使用)

- キャッシュはラウンドロビンでリクエストを受付ける
 - 相互に兄弟関係かつProxy-only オプション
 - 他のキャッシュから得たオブジェクトはローカルに保存しない
 - 大体, 分散できる
 - Cache Digest
 - 定期的にオブジェクトの一覧を交換する
 - かなりラフな分散になる
 - ICPの利用
 - ICPパケットの量は無視できない
 - Multicast ICP

リバースプロキシ

- バーチャルホスティング
 - 一台のホストで複数のWWWサーバの役割を果たす
 - バーチャルIP型
 - 一つのインターフェースにWWWサーバ分のIPアドレスを割り振る
 - WWWサーバデーモンをそれぞれのIPアドレスにバインドさせる



リバースプロキシ

- HOSTヘッダ型

- ホストに付けられたIPアドレスは一つ
- HOSTヘッダ (要求ヘッダ HTTP/1.1) により、WWWのドキュメントのルートディレクトリを変える
- HOSTヘッダを生成しない古いブラウザは対応不可能

- http://foo.com/index.html



Internet Week 2000 Tutorial (12/19/2000)

101

リバースプロキシ

- WWWサーバー台

- httpd_accel_host 実サーバ名
- httpd_accel_port 実サーバポート番号
- キャッシュ不可オブジェクトの指定
 - acl QUERYurlpath_regex /your-cgi-bin ?
 - no_cache deny QUERY
- HOSTヘッダの書き換えも自動的に行われる



Internet Week 2000 Tutorial (12/19/2000)

102

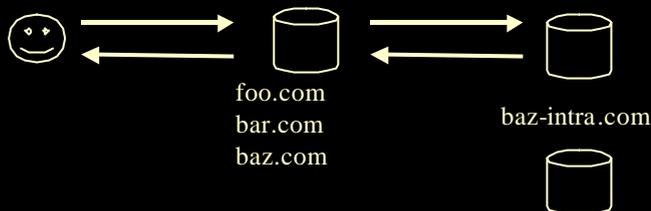
リバースプロキシ

- HOSTヘッダ型バーチャルホスティング (一般)

- HOSTヘッダによるオリジナルホスト指定
- フロントサーバと実サーバの組を設定

foo.com	foo-intra.com	foo-intra.com
bar.com	bar-intra.com	
baz.com	baz-intra.com	

- Squidでは実現不可能



Internet Week 2000 Tutorial (12/19/2000)

103

リバースプロキシ

- HOSTヘッダ型バーチャルホスティング(Squid)

- httpd_accel_host virtual
- httpd_accel_port 実サーバポート番号
- httpd_accel_uses_host_header on
- Squidの制限
 - HOSTヘッダによる切り分け
 - HOSTヘッダ = 実サーバ
 - リバースプロキシ = オリジナルホスト (IPアドレスは同じ)
 - ポート番号で区別する

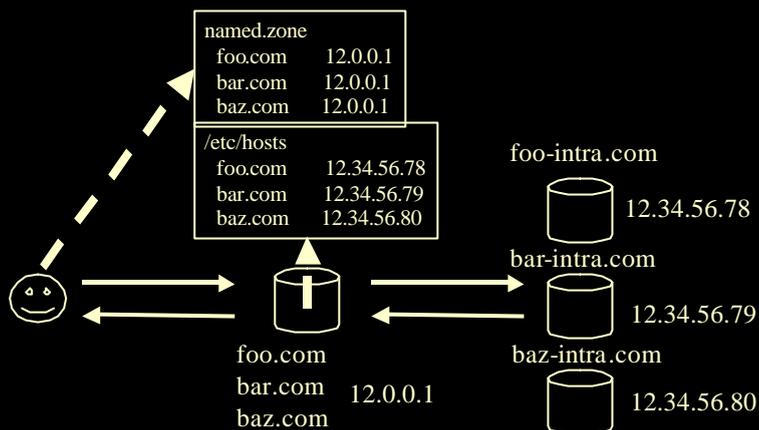
Internet Week 2000 Tutorial (12/19/2000)

104

リバースプロキシ

- HOSTヘッダ型バーチャルホスティング(Squid)

- ローカルなResolve規則



Internet Week 2000 Tutorial (12/19/2000)

105

リバースプロキシ

- バーチャルIP型バーチャルホスティング(Squid)

- `httpd_accel_host virtual`
- `redirect_program /usr/local/squid/bin/rewrite`

- `rewrite`

- `while (<>) {`
- `s@http://203.181.248.20@http://foo-int.com@;`
- `s@http://203.181.248.21@http://bar-int.com@;`
- `print $_;`
- `}`

Internet Week 2000 Tutorial (12/19/2000)

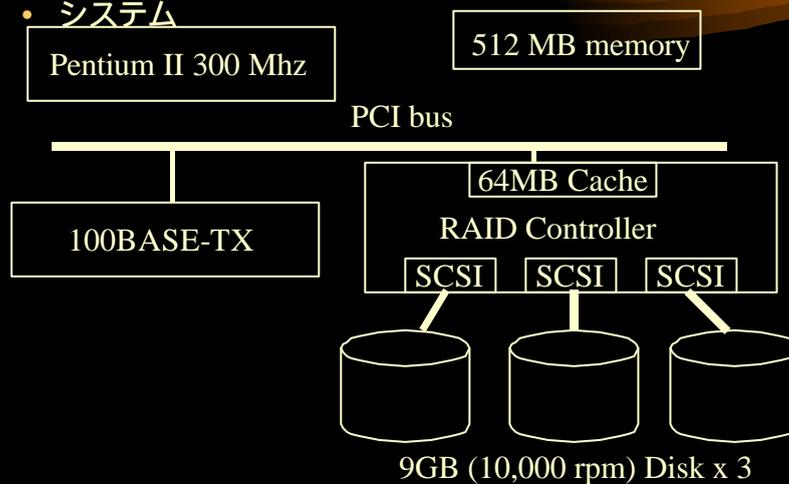
106

実システム(1)

- APAN Tokyo Root Cache Server (ATRC)
 - APAN (Asia-Pacific Advanced Network)
 - Root サーバ
 - 各サイトのキャッシュサーバからのリクエストを受ける
 - ハードウェアスペック
 - Pentium II 300MHz , 512 MB Memory
 - 9.1GB Disk x 3 (Ultra-Wide 10,000 rpm)
 - RAID Controller (PCI to SCSI)
 - DPT PM3343UW/3+64MB Cache
 - 100BASE-TX

実システム(2)

システム



実システム(3)

- 設定
 - ディスク全体をRAID-0 (ストライピング)
 - 3台のディスクを最大限に使う
 - 一台のディスクが故障するとシステム全体の再インストールが必要 (故障する確率は単体のディスクの3倍)
- 運用状況
 - 100万リクエスト/Day
 - 10GB/Day
 - ヒット率(リクエストベース): 20%
 - ヒット率(トラフィックベース): 10%
 - ヒット率(TCPベース): 10%

実システム(4)

- パフォーマンス
 - 通常, スワップは不使用
 - ピーク時
 - 100リクエスト/Sec (瞬間最大: Calamariによる統計)
 - 30リクエスト/Sec (5分間の平均)
 - 3M bps
 - CPU Usage: 50%
 - 同時TCPコネクション: 500
 - 使用ファイルディスクリプタ: 600
 - たまにMBUF領域を使い切りOSダウン

Part 3 WWW以外への適用

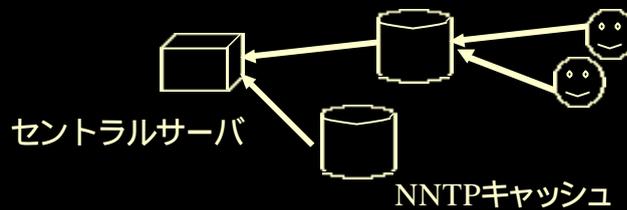
- NEWS
- ストリーミング

NEWS(1)

- 既存のUsenetのモデル
 - グループ単位の配送
 - 実際にユーザが読むかどうか?
 - それぞれのサーバ間で記事のパケットリレー
 - サーバ単位で記事やNews Groupの管理
 - 管理コスト, 処理コストが大きい

NEWS(2)

- NNTPキャッシュの構成
 - セントラルサーバ
 - 既存のNewsサーバ
 - 記事やNews Groupの管理
 - 記事の配送
 - ユーザからのリクエストは直接受けない
 - NNTPキャッシュサーバ
 - 読まれた記事のキャッシュ
 - キャッシュコンテンツの管理
 - News Groupの管理は自動



Internet Week 2000 Tutorial (12/19/2000)

113

NEWS(3)

- メリット
 - ディスク容量, ネットワークトラフィックの節約
 - 管理コストの低減
- 運用例:
 - 複数のNewsサーバを管理
 - 一台をセントラルサーバ, その他をNNTPキャッシュサーバ
 - NNTPキャッシュサーバ間のラウンドロビンも可能
 - 小規模なNewsサーバを管理
 - 他の組織のセントラルサーバを使い, ローカルにはNNTPキャッシュサーバのみ

Internet Week 2000 Tutorial (12/19/2000)

114

ストリーミング(概要)

- 21世紀のメイントラフィック
 - Webトラフィックを超えると見られている
 - VHS程度の品質=1.5 M bpsの帯域が必要 (MPEG-1)
 - 600K bps VHSレベル
 - 1.5M bps DVDレベル
- ネットワークの安定性&太さ&QOSが重要
 - コンテンツをユーザの近くに置く(複製)効果大きい
- キャッシュ技術の次のターゲットはストリーム

ストリーミング(特徴)

- 高価値なコンテンツ
 - 複製管理
 - コピープロテクション
 - アカウンティング
 - 披れき管理
 - ユーザ認証
- 不完全性なコンテンツ転送
 - 不快にならない程度にパケット落ちを許す
 - 配信用にコンテンツを加工
- 連続メディア
 - 部分的・途中からのコンテンツの再生

ストリーミング(種類)

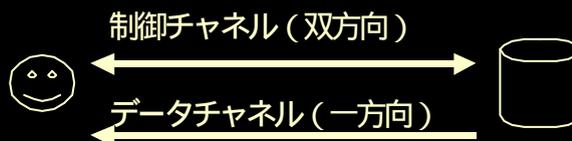
- ダウンロード型
 - コンテンツをHTTP、FTP等で提供
 - ダウンロードしながらの再生も可能
 - 既存のWWW用キャッシュサーバでキャッシュ可能
- オンデマンド型
 - 専用プロトコルを使用
 - ユーザ側に複製を作らせない
 - 専用のキャッシュサーバが必要
- ライブ型
 - キャッシュ不可能
 - バッファリングを工夫しているものはある

Internet Week 2000 Tutorial (12/19/2000)

117

ストリーミング(プロトコル概要)

- チャンネル
 - 制御チャンネル
 - パスワード処理、ストリーム制御(早送り、停止等)、コンテンツ情報
 - 通常TCP
 - データチャンネル
 - コンテンツの配送
 - 通常UDP



Internet Week 2000 Tutorial (12/19/2000)

118

ストリーミング(プロトコル一覧)

• プロトコル

	制御	データ
- スタンダード*	RTSP (TCP)	RTP(TCP,UDP)
- RealNetworks (旧)	PNA (TCP)	PNA(TCP,UDP)
- RealNetworks (新)	RTSP (TCP)	RDT(TCP,UDP)
- Microsoft	MMS	MMS

- * スタンダード準拠
 - RealNetworks, Apple QuickTime, Java Media Framework (JMF), Cisco IP/TV
- HTTP Cloaking
 - HTTPによる上記プロトコルのエンカプセレーション

ストリーミング(プロトコル-1)

- Real Time Streaming Protocol (RTSP)
 - ポート番号 : 554
 - rtsp://...
- Progressive Networks Audio (PNA)
 - ポート番号 : 7070
 - pnm://...
 - Progressive Networks Media
- Microsoft Media Server (MMS)
 - mms://...

ストリーミング(プロトコル2)

- Real-time Transport Protocol (RTP)
- RealNetworks Data Transport (RDT)
- Session Description Protocol (SDP)

ストリーミング(メディアファイル)

- Real Networks用
 - Real Audio Meta File
 - .ram, .rpm
 - Synchronized Multimedia Integration Language (SMIL)
 - .simpl, .smi
 - RealAudio
 - .rm, .ra
 - RealVideo
 - .rm
 - RealPix streaming image markup
 - .rp
 - RealPix ad rotation
 - .rpa
 - RealText streaming text
 - .rt
 - Flash Player file
 - .swf

ストリーミング (RTSP 標準フロー)

- OPTIONS
 - 使用可能なメソッドの一覧を得る
- DESCRIBE
 - メディアの情報を取得
 - 情報の転送はSDP
- SETUP
 - コネクションの確立
- SET_PARAMETER
 - パラメータの設定
- PLAY
 - 再生
- PAUSE
 - 一時停止
- TEARDOWN
 - コネクションの切断

ストリーミング (RTSP その他メソッド)

- その他メソッド
 - ANNOUNCE
 - メディアの情報を変更
 - RECORD
 - 保存
 - REDIRECT
 - 他のサーバにリダイレクト

ストリーミング (HTTP Cloaking-1)

- RTSP/RDT等をHTTPでエンカプセル
 - 主にFireWallを通過させるため
- HTTPコネクションを二つ使う (RealNetworksの場合)
 - 制御、データではない
 - POST
 - クライアント → サーバへのコネクションをHTTP化
 - RTSPリクエスト (バイナリ - 化?)
 - GET
 - サーバ → クライアントへのコネクションをHTTP化
 - RTSPリプライ + コンテンツ
 - ストリーム再生中は、POSTリクエストとGETリクエストが継続する

Internet Week 2000 Tutorial (12/19/2000)

125

ストリーミング (HTTP Cloaking-2)

POST /SmpDsBhgRl HTTP/1.0

GET /SmpDsBhgRl141609c1-a365-11d4-e6c3-0f600d57940a HTTP/1.0

141609c1-a365-11d4-e6c3-0f600d57940a

RTSPリクエスト1

RTSPリクエスト2

RTSPリクエスト3

RTSPリプライ1

RTSPリプライ2

RDTデータ

RTSPリプライ3



Internet Week 2000 Tutorial (12/19/2000)

126

ストリーミング (HTTP Cloaking & Proxy)

- 問題点

- キャッシュ不可能な長時間のコネクション
 - Proxyのリソースの占有
 - Proxyの最大コネクション保持時間によるストリームの切断
- Proxyにおけるコンテンツの分散と相性が悪い
 - ストリームサーバ側でL7SWを使ってクラスタリング
 - 2つのコネクションを同じストリームサーバに割振る
 - ソースIPアドレスによるコネクション割振り
 - PUTとGETを別ProxyにURLハッシュによって分散
 - 異なるIPアドレスからのコネクション

ストリーミング (RTSP/RTP キャッシング)

- RFC2326

- HTTPと同様のキャッシュ管理
 - Cache-Control:
 - SETUP時に、オブジェクトのキャッシュ属性をセットする
- HTTPとの違い
 - オブジェクトのDescriptionもキャッシュする
 - 連続メディア
 - 保持しているオブジェクトを流している時には、そのオブジェクトのDescriptionを返す
 - 通過したストリームだけをキャッシュする

ストリーミング (RTSP/RTP キャッシング2)

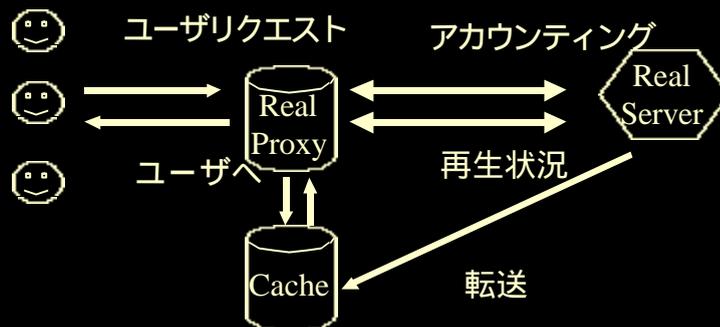
- RFCドラフト
 - Caching Support in Standards-based RTSP/RTP Servers
- メタチャンネル
 - RTPで失われる情報を補う
- リクエストを中継する (キャッシュヒット時も)
 - SETUP
 - TEARDOWN
- Open Issues
 - コピープロテクション
 - アクセス・アカウンティング
 - ユーザ認証

ストリーミング (RealNetworks用キャッシュ 1)

- 特徴
 - アカウンティング
 - 制御チャンネルを素通しさせる
 - キャッシュ上のコンテンツもアカウンティングを行う
 - 配布はセントラルサーバで管理
 - すべてのLOGはストリームサーバにリアルタイムに送られる
 - ストリームサーバへの接続が切れると、コンテンツの配送を中止
 - サーバからキャッシュへの転送は特別のプロトコルを使用
 - オブジェクトの完全な転送
 - コンテンツの途中からの再生も可能
 - キャッシュサーバは一種のファイルシステムとして使う
 - Real Proxy + ファイルシステム

ストリーミング (RealNetworks用キャッシュ2)

- 概要



Internet Week 2000 Tutorial (12/19/2000)

131

終わりに

- ユーザ側キャッシュ
 - アプライアンス化
 - 特別なノウハウはあまりない (頑張っても得られる物は少ない)
- サーバ側キャッシュ (特にContent Distribution Service)
 - コンテンツのコントロールが重要
 - まだノウハウを蓄積している状態
- ご質問等はCDS-MLまで
 - <http://cache.jp.apan.net/CDS-ML/>
 - <http://www.kosho.org/>

Internet Week 2000 Tutorial (12/19/2000)

132

付録

1. HTTPヘッダー一覧
2. HTTP応答コード一覧
3. RTSPサンプル

付録1 (HTTPヘッダ)

- 要求ヘッダ
- 応答、一般ヘッダ
- エンティティヘッダ

要求ヘッダ

- Accept
- Accept-Charset
- Accept-Encoding
- Accept-Language
- Authorization
- From
- Host
- If-Modified-Since
- If-Match
- If-None-Match
- If-Range
- If-Unmodified-Since
- Max-Forwards
- Proxy-Authorization
- Range
- Referer
- User-Agent

応答、一般ヘッダ

- Age
- Location
- Proxy-Authenticate
- Public
- Retry-After
- Server
- Vary
- Warning
- WWW-Authenticate
- Cache-Control
- Connection
- Date
- Pragma
- Transfer-Encoding
- Upgrade
- Via

エンティティヘッダ

- Allow
- Content-Base
- Content-Encoding
- Content-Language
- Content-Length
- Content-Location
- Content-MD5
- Content-Range
- Content-Type
- ETag
- Expires
- Last-Modified

付録2 (HTTP 応答コード)

- 100, 200, 300番台
- 400番台
- 500番台

応答コード(100, 200, 300)

- 100 Continue
- 101 Switching Protocols
- 200 OK
- 201 Created
- 202 Accepted
- 203 Non-Authoritative Information
- 204 No Content
- 205 Reset Content
- 206 Partial Content
- 300 Multiple Choices
- 301 Moved Permanently
- 302 Moved Temporarily
- 303 See Other
- 304 Not Modified
- 305 Use Proxy

応答コード(400)

- 400 Bad Request
- 401 Unauthorized
- 402 Payment Required
- 403 Forbidden
- 404 Not Found
- 405 Method Not Allowed
- 406 Not Acceptable
- 407 Proxy Authentication Required
- 408 Request Time-out
- 409 Conflict
- 410 Gone
- 411 Length Required
- 412 Precondition Failed
- 413 Request Entity Too Large
- 414 Request-URI Too Large
- 415 Unsupported Media Type

応答コード (500)

- 500 Internal Server Error
- 501 Not Implemented
- 502 Bad Gateway
- 503 Service Unavailable
- 504 Gateway Time-out
- 505 HTTP Version not supported

付録3(RTSP サンプル)

- シーケンス
 - OPTIONS
 - DESCRIBE
 - SETUP
 - SET_PARAMETER
 - PAUSE
 - PLAY
 - TEARDOWN

Options

- OPTIONS rtsp://127.0.0.1:554 RTSP/1.0
- CSeq: 1
- User-Agent: RealMedia Player Version 6.0.7.1363 (linux-2.0-libc64386-gcc2.95)
- ClientChallenge: 0e6b6e8223bad4971a5698ecdc1ad1ae
- PlayerStarttime: [16/10/2000:21:10:32 00:00]
- CompanyID: UmmMs41IR3IcEaYTW0gPVQ==
- GUID: 00000000-0000-0000-0000-000000000000
- RegionData: 0
- ClientID: Linux_2.2_6.0.7.1363_play32_RN01_EN_586
- Pragma: initiate-session

- RTSP/1.0 200 OK
- CSeq: 1
- Date: Mon, 16 Oct 2000 12:10:33 GMT
- Server: RealServer Version 6.1.3.970 (linux-2.0-libc6-i386)
- Public: OPTIONS, DESCRIBE, ANNOUNCE, SETUP, GET_PARAMETER, SET_PARAMETER, TEARDOWN
- RealChallenge1: 618931ec3fbd22c2b936676742a5f9c8
- StatsMask: 3

DESCRIBE-1

- DESCRIBE rtsp://127.0.0.1:554/g2video.rm RTSP/1.0
- CSeq: 2
- Accept: application/sdp
- Bandwidth: 57600
- GUID: 00000000-0000-0000-0000-000000000000
- RegionData: 0
- ClientID: Linux_2.2_6.0.7.1363_play32_RN01_EN_586
- SupportsMaximumASMBandwidth: 1
- Language: en-US
- Require: com.real.retain-entity-for-setup

DESCRIBE-2



- RTSP/1.0 200 OK
- CSeq: 2
- Date: Mon, 16 Oct 2000 12:10:33 GMT
- Set-Cookie: cbid=efhggmdicjnkclmeorraqplqrojrkltufkegidlffjjiplpsjonpoqqomspqurfjgehil;path=/;expires=Thu,31-Dec-2037 23:59:59 GMT
- vsrc: http://127.0.0.1:8080/viewsource/template.html?nuyhtgqq7iz69cbuafoBhvt6qyqDreA1v1yeC3d4ngEt5o5g000000
- X-TSPort: 7802
- Last-Modified: Sun, 15 Oct 2000 16:50:36 GMT
- Content-base: rtsp://127.0.0.1:554/g2video.rm/
- ETag: 1812673537-2
- Content-type: application/sdp
- Content-length: 4427

- v=0
- o=- 971628636 971628636 IN IP4 127.0.0.1
- s=Creating Great Video for the Web
- i=RealNetworks RealNetworks 92000

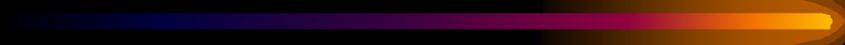
SETUP



- SETUP rtsp://127.0.0.1:554/g2video.rm/streamid=1 RTSP/1.0
- CSeq: 4
- RDTFeatureLevel: 2
- Transport: x-real-rtt/udp;client_port=6970;mode=play
- Session: 1812673537-2

- RTSP/1.0 200 OK
- CSeq: 4
- Date: Mon, 16 Oct 2000 12:10:33 GMT
- Session: 1812673537-2
- Transport: x-real-rtt/udp;client_port=6970;server_port=27156

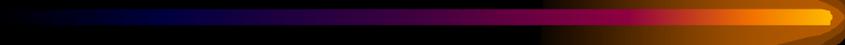
SET_PARAMETER



- SET_PARAMETER rtsp://127.0.0.1:554/g2video.rm RTSP/1.0
- CSeq: 5
- Subscribe: stream=0;rule=2,stream=0;rule=3,stream=1;rule=8,stream=1;rule=9
- Session: 1812673537-2

- RTSP/1.0 200 OK
- CSeq: 5
- Date: Mon, 16 Oct 2000 12:10:51 GMT
- Session: 1812673537-2

PLAY



- PLAY rtsp://127.0.0.1:554/g2video.rm RTSP/1.0
- CSeq: 6
- Session: 1812673537-2
- Range: npt=0-82.896000

- RTSP/1.0 200 OK
- CSeq: 6
- Date: Mon, 16 Oct 2000 12:10:51 GMT
- RTP-Info: url=rtsp://127.0.0.1:554/g2video.rm/streamid=0;seq=0;rtpime=0,
url=rtsp://127.0.0.1:554/g2video.rm/streamid=1;seq=0;rtpime=0

SET_PARAMETER



- SET_PARAMETER * RTSP/1.0
- CSeq: 7
- Ping: Pong

- RTSP/1.0 451 Parameter Not Understood
- CSeq: 7
- Date: Mon, 16 Oct 2000 12:10:51 GMT

SET_PARAMETER



- SET_PARAMETER rtsp://127.0.0.1:554/g2video.rm RTSP/1.0
- CSeq: 8
- SetDeliveryBandwidth: Bandwidth=27787;BackOff=0
- Session: 1812673537-2

- RTSP/1.0 200 OK
- CSeq: 8
- Date: Mon, 16 Oct 2000 12:10:51 GMT
- Session: 1812673537-2

- SET_PARAMETER rtsp://127.0.0.1:554/g2video.rm RTSP/1.0
- CSeq: 9
- SetDeliveryBandwidth: Bandwidth=57600;BackOff=0
- Session: 1812673537-2

- RTSP/1.0 200 OK
- CSeq: 9
- Date: Mon, 16 Oct 2000 12:10:55 GMT
- Session: 1812673537-2

PAUSE

- PAUSE rtsp://127.0.0.1:554/g2video.rm RTSP/1.0
- CSeq: 10
- Session: 1812673537-2

- RTSP/1.0 200 OK
- CSeq: 10
- Date: Mon, 16 Oct 2000 12:11:02 GMT

PLAY

- PLAY rtsp://127.0.0.1:554/g2video.rm RTSP/1.0
- CSeq: 11
- Session: 1812673537-2
- Range: npt=59.468000-82.896000

- RTSP/1.0 200 OK
- CSeq: 11
- Date: Mon, 16 Oct 2000 12:11:08 GMT
- RTP-Info: url=rtsp://127.0.0.1:554/g2video.rm/streamid=0;seq=57;rtpime=26820,
url=rtsp://127.0.0.1:554/g2video.rm/streamid=1;seq=101;rtpime=27000

TEARDOWN



- TEARDOWN rtsp://127.0.0.1:554/g2video.rm RTSP/1.0
- CSeq: 15
- Session: 1812673537-2

- RTSP/1.0 200 OK
- CSeq: 15
- Date: Mon, 16 Oct 2000 12:11:17 GMT