

ストリーミングCDN

ストリーミングシステム(II)

Internet Week 2001 チュートリアル
2001年12月4日、パシフィコ横浜
鍋島 公章

目次

- ◆ストリーミング入門
- ◆CDN入門
- ◆負荷分散
- ◆オンデマンドCDN
- ◆ライブCDN
- ◆動的ストリーミングとCDN

Part 1 ストリーミング入門

ストリーミング入門(1)

◆ 特徴 (WWWとの違い)

- 連続メディア
 - ◆ 途中からの再生を許す
 - 制御チャンネル
 - データチャンネル
- リアルタイム・不完全性
 - ◆ 1分間のコンテンツは1分で転送
 - 欠落データはあきらめる
 - ◆ 人間の感性
 - ある程度の乱れには寛容

ストリーミング入門(2)

◆ プロトコル

- 制御チャンネル
 - ◆ 例: RTSP over TCP
 - ◆ ストリーム制御(再生、停止、早送り、等)
 - ◆ パラメータ設定
 - ◆ コンテンツ属性
- データチャンネル
 - ◆ 例: RTP over UDP
 - ◆ コンテンツの転送
 - 再送要求
 - 配送レート制御

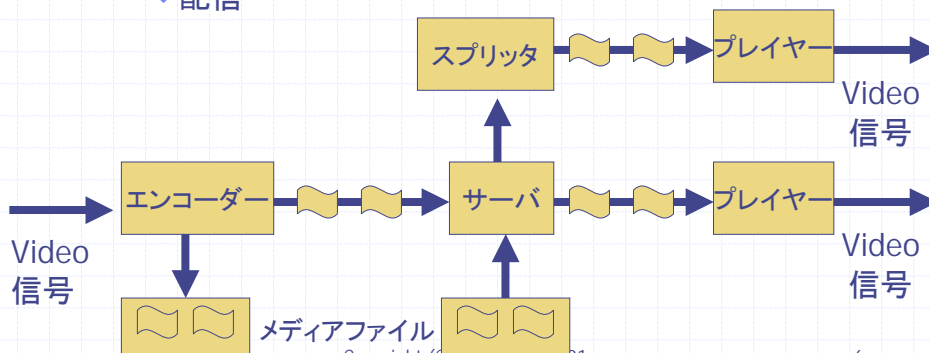


5

ストリーミング入門(3)

◆ システム

- エンコーダー
 - ◆ 圧縮・変換
- サーバ
 - ◆ 配信
- スプリッタ
 - ◆ 中継
- プレイヤー
 - ◆ 伸張・復元



6

ストリーミング入門(4)

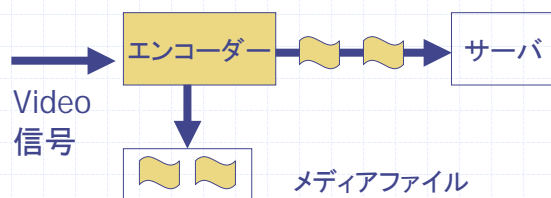
◆ エンコーダー

- Video信号をエンコーディングしたものを、チャンクに分断

- ◆ ライブ
 - サーバに送信
- ◆ アーカイブ
 - ファイルに格納

◆ メディアファイル

- ヘッダ
 - ◆ 属性情報
- データ領域
 - ◆ チャンク化されたストリーミングコンテンツ
 - ◆ チャンクは、そのままストリーミングパケットに載る



Copyright (C) kosho.org 2001

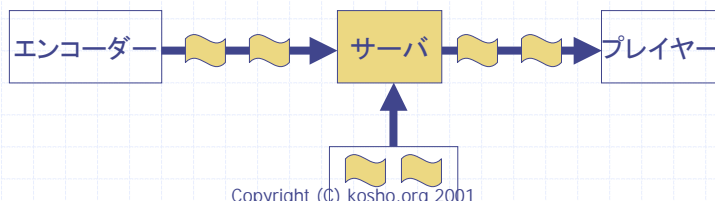
7

ストリーミング入門(5)

◆ サーバ

- シーケンス番号を付加して、チャンクを配送プロトコルで配信

- ◆ オンデマンド
 - ファイル中のチャンクをパケットに載せ送出
- ◆ ライブ
 - 基本的にそのまま中継
 - サーバとして配信を実行



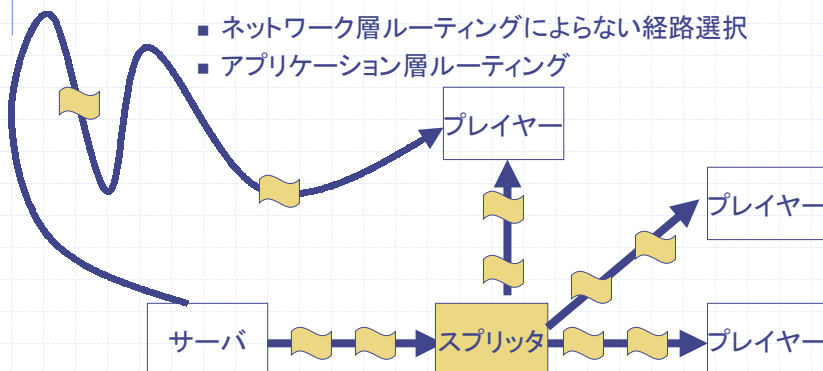
Copyright (C) kosho.org 2001

8

ストリーミング入門(6)

◆ スプリッタ

- ストリーム接続の中継、分岐
 - ◆ 一つの(ライブ)ストリームを、複数のユーザに分岐
 - ◆ 中継ポイントの設定による、経路の選択
 - ネットワーク層ルーティングによらない経路選択
 - アプリケーション層ルーティング



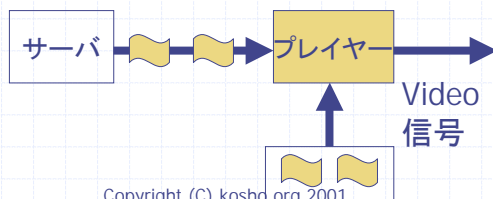
Copyright (C) kosho.org 2001

9

ストリーミング入門(7)

◆ プレイヤー

- Video信号への復元
- バッファリング
 - ◆ パケット到着順の訂正
 - ◆ 欠落パケットの再送要求
 - 再生前に再送要求
 - ◆ 一時的な帯域不足
 - バッファリング中に回復を待つ



Copyright (C) kosho.org 2001

10

Part 2 CDN入門

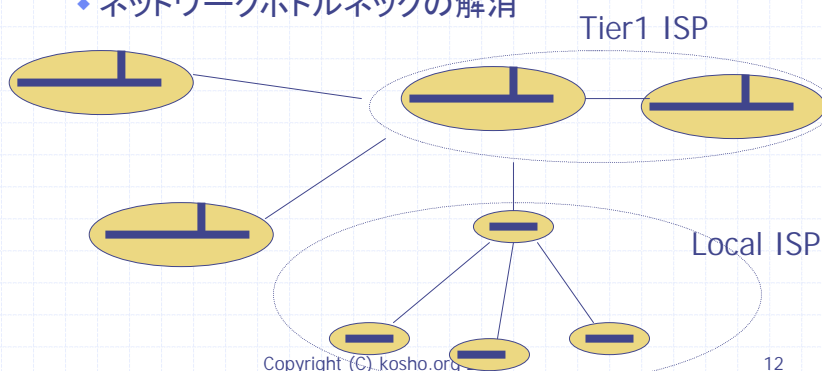
CDN入門(1)

◆ Content Distribution Network (CDN)

- ◆ Content Distribution Service Provider (CDSP)

- 目的

- ◆ ユーザアクセスの高速化
- ◆ ネットワークボトルネックの解消

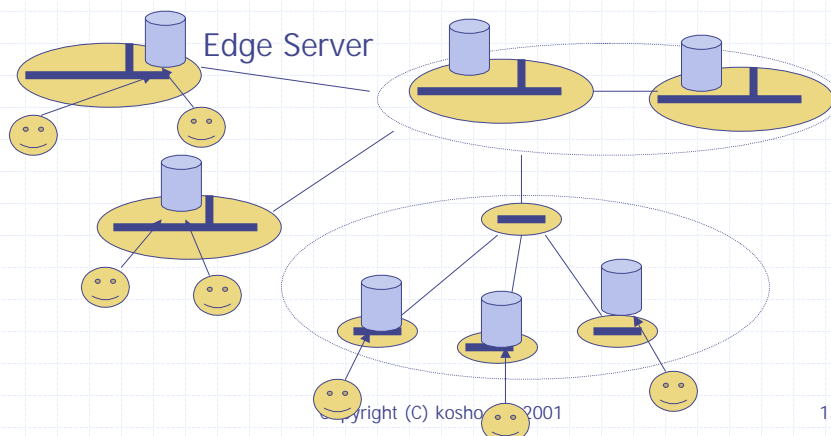


CDN入門(2)

◆ セグメント内に、コンテンツの複製、スプリッタを配置



Original Server



13

CDN入門(3)

◆ エッジサーバ

- 複製のプレイスホルダ
 - ◆ リバースプロキシ
 - ◆ 部分的ミラーサーバ
- スプリッタ

◆ コンテンツ・レゾルバ

- 最適な複製(エッジサーバ)を選択

◆ コンテンツ&Log管理

- 複製の配置、アクセスLogの回収

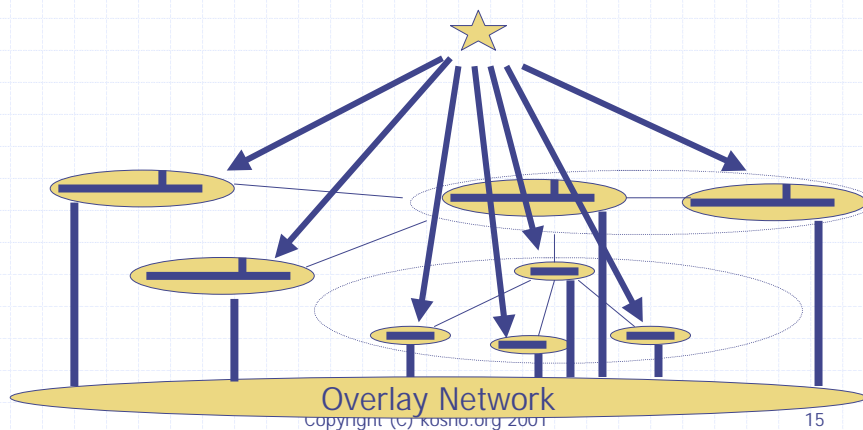
Copyright (C) kosho.org 2001

14

CDN入門(4)

◆ オーバレイネットワーク

- 全ネットワークとダイレクトに接続したネットワーク
- 衛星による一斉配信



15

CDN入門(5)

◆ ストリーム用CDNの特徴

- ストリームはQoSに敏感
 - ◆ 広帯域、長時間
 - ◆ アグレッシブかつ強力なCDNが必要
- ライブ配信とオンデマンド配信
 - ◆ 経路、QoS制御
- ストリームの複製、動的生成は困難

Copyright (C) kosho.org 2001

16

CDN入門(6)

◆CDN技術と実装レベル

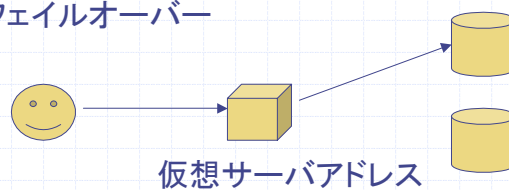
- グローバル・ローカル負荷分散
 - ◆ ラウンドロビン
 - ◆ 負荷分散
 - ◆ 地理的分散
 - ◆ 地理的分散+負荷分散
- コンテンツ複製技術
 - ◆ キャッシュ型
 - ◆ 複製操作(プレロード、明示的削除)型
- アクセスLOG管理
 - ◆ 回収型
 - ◆ 即時型

Part 3 負荷分散

ローカル負荷分散(1)

◆ Layer 4スイッチ

- 一つの仮想サーバアドレスを複数のサーバで共有
 - ◆ 負荷分散
 - ◆ フェイルオーバー



■ 手法

- ◆ MAT (Mac Address Translation)
- ◆ NAT (Network Address Translation)

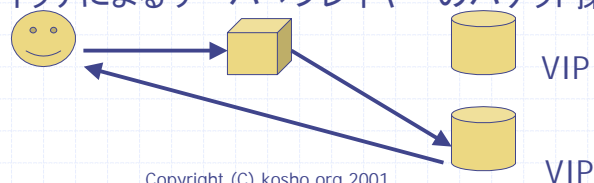
Copyright (C) kosho.org 2001

19

ローカル負荷分散(2)

◆ MAT (Mac Address Translation)

- 仮想サーバアドレスを各サーバに設定
 - ◆ Loopbackインターフェイス
- 通常は、サーバ間で仮想サーバアドレスあてパケットの争奪戦が発生
- スイッチが争奪戦を回避
 - ◆ ARP不使用、MACアドレス指定によるサーバ指定
 - ◆ 出力スイッチPort指定
- L4スイッチによるサーバ⇒プレイヤーのパケット操作なし



Copyright (C) kosho.org 2001

20

ローカル負荷分散(3)

◆ Loopbackインターフェイスへの仮想アドレスの割当

- Unix系
 - ◆ エイリアス
 - ◆ `ifconfig lo:1 129.168.0.200 netmask 255.255.255.255`
- Windows 2000
 - ◆ デフォルトではLoopbackインターフェイスを持たない
 - ◆ Loopbackインターフェイスを組み込み、アドレスを割り振る

◆ サーバの設定

- 仮想IPにバインディング

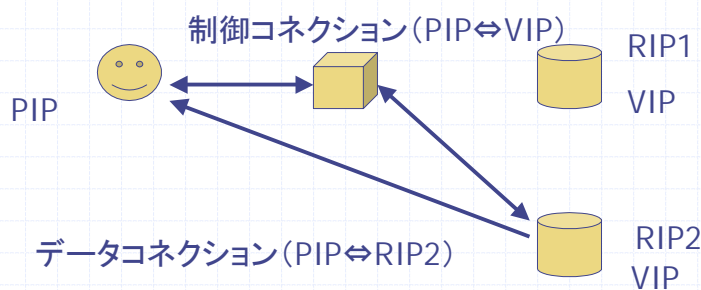
Copyright (C) kosho.org 2001

21

ローカル負荷分散(4)

◆ サーバヒンティング

- 制御コネクションによる、データコネクション用サーバの指定
- プレイヤー側の対応も必要



Copyright (C) kosho.org 2001

22

ローカル負荷分散(5)

◆ DSR (Direct Server Return)

- プレイヤーとサーバ間で直接にデータコネクションを持つ
 - ◆ MAT
 - ◆ サーバヒンティング
- ストリーミングの場合、入るパケットと出るパケット数が大きく異なる
 - ◆ 出るパケットを直接Internetに流す事により、負荷分散装置の負荷を低減
- NAT不使用
 - ◆ データコネクションを上手く扱えない負荷分散装置でも動く可能性が高い
- コネクション数の分散のみ
 - ◆ トラフィックの分散はできない
 - ◆ 詳細な負荷分散には、ストリーミングサーバ上の情報収集エージェントが必要

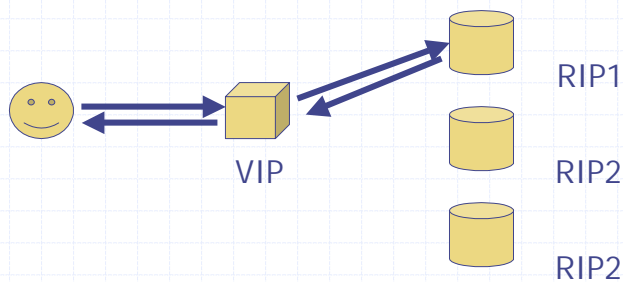
Copyright (C) kosho.org 2001

23

ローカル負荷分散(6)

◆ NAT (Network Address Translation)

- SWが仮想サーバとしてパケットを受取る
- IPヘッダの書換
 - ◆ VIP→RIPx
- サーバ⇒プレイヤーへのパケットのIPアドレスも書換
 - ◆ サーバのデフォルトGWをL4SWに設定



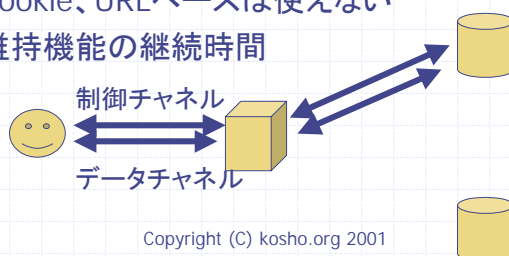
Copyright (C) kosho.org 2001

24

ローカル負荷分散(7)

◆ 一般的な注意(1)

- 制御チャンネル、データチャンネルの2つのコネクションを同時使用
- 維持(パーシステンス)機能が必要
 - ◆ プレイヤーからのコネクションを一つのサーバに割振
 - ◆ IPアドレスベースが一般的
 - 同じIPアドレスのリクエストは同一サーバに振分ける
 - ◆ Cookie、URLベースは使えない
 - ◆ 維持機能の継続時間



Copyright (C) kosho.org 2001

25

ローカル負荷分散(8)

◆ 一般的な注意(2)

- 負荷分散するポートは慎重に決める
 - ◆ 必要十分なポートをSLBに設定する
 - ◆ マルチポート
 - 互換性、HTTPクローキング等への対応のため、複数のポートを用意
 - HTTP, RTSP, ...
 - プレイヤーは使用可能なポート、プロトコルをプローブ
 - ◆ SLBのディレイド・コネクション
 - 不必要なポートをSLBに設定すると、プレイヤーのプローブ時にタイムアウト待ちが発生
 - ◆ スプリティング、キャッシュ用ポート
 - 特別なポート・プロトコルが使われている場合がある

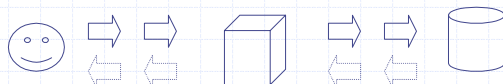
Copyright (C) kosho.org 2001

26

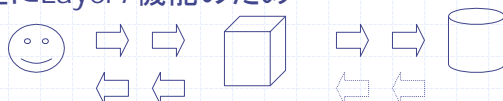
ローカル負荷分散(9)

◆ 一般的な注意(3)

- シンプルNAT
 - ◆ SLBはパケット書換えのみ



- デイレイドコネクション
 - ◆ 実サーバに代わりSLBがTCPコネクションを受ける
 - ◆ 主にLayer7機能のため



Copyright (C) kosho.org 2001

27

ローカル負荷分散(10)

◆ 一般的な注意(4)

- データチャネルの負荷分散は、通常のTCPの負荷分散 (WWWと同様) と異なる設定が必要
 - ◆ コネクションの方向とプロトコル
 - 制御コネクション
 - プレイヤー⇒サーバ
 - ポート番号は固定
 - TCP
 - データコネクション
 - サーバ⇒プレイヤー
 - ポート番号はフローティング
 - UDP

Copyright (C) kosho.org 2001

28

ローカル負荷分散(11)

◆ 一般的な注意(5)

- 制御コネクション(RTSP)
 - ◆ HTTPと類似のプロトコル
 - ◆ HTTPと同様の負荷分散が可能
- データコネクション
 - ◆ サーバからのコネクション開始
 - ◆ プレイヤーから定期的に転送状況をサーバに通知
 - データパケット欠落の再送要求
 - 無くても動く
 - パケット落ちが始まると映像のクオリティが急激に低下
 - ◆ フローティングポート
 - レンジ指定が必要
 - 全ポートを空ける
 - セキュリティ上の注意が必要

Copyright (C) kosho.org 2001

29

ローカル負荷分散(12)

◆ 一般的な注意(6)

- 負荷分散のメトリック
 - ◆ (重み付き)ラウンドロビン
 - ◆ コネクション数
 - ◆ トラフィック量
 - ストリーム1本の帯域により負荷が異なる
 - マルチレートエンコーディング
 - ◆ サーバ応答時間
 - あくまでもTCPの接続応答時間
 - ◆ サーバ負荷
 - ストリームサーバ上にエージェントが必要

Copyright (C) kosho.org 2001

30

グローバル負荷分散(1)

◆ サイト間の負荷分散

- 負荷+ネットワークポロジ-を考慮

◆ 手法

- DNS
- リダイレクト、動的生成
- Anycast
- (L4SW間トンネリング)

グローバル負荷分散(2)

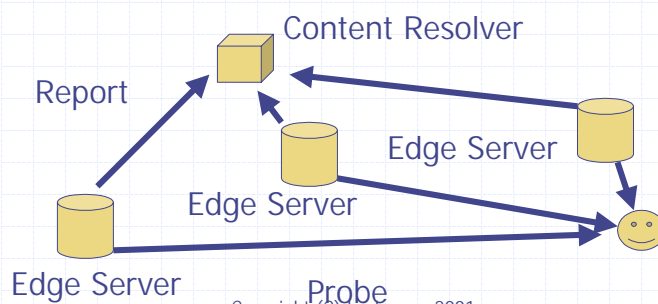
◆ ボトルネックの傾向

- 300 kbpsのストリームの普及で、帯域は100倍？
 - ◆ 平均アクセス速度が10倍
 - WWWの平均帯域30Kbps程度
 - ◆ ストリーミングの視聴により、帯域占有時間10倍
- 全世界
 - ◆ 国際間リンク
- 日本国内
 - ◆ バックボーンはある程度余裕
 - ◆ ISP内のボトルネックが顕在化

グローバル負荷分散(3)

◆ 近隣エッジサーバ選択

- それぞれのエッジサーバから計測
- コンテンツレゾルバで集計
- BGP AS Path、Ping RTT、Router Hop、...
 - ◆ 最も精度が高いのはPing RTT



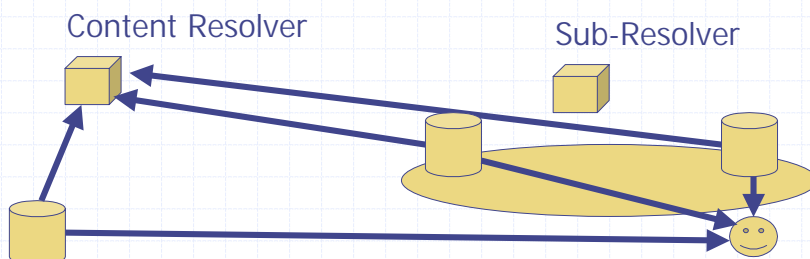
Copyright (C) kosho.org 2001

33

グローバル負荷分散(4)

◆ Content Resolvingのレスポンス問題

- 随時型の限界
 - ◆ 例: Content Resolverからプレイヤーが遠い場合
- 定期的なプロービング、データベースへの格納
- グループ化



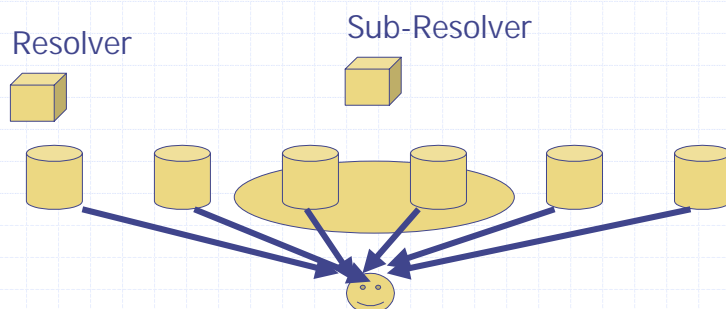
Copyright (C) kosho.org 2001

34

グローバル負荷分散(5)

◆ Content Resolvingのスケラビリティ問題

- エッジサーバが1000箇所あれば、1000台からのプローブが飛ぶ
- グループ化によりプローブを行うエッジサーバを制限



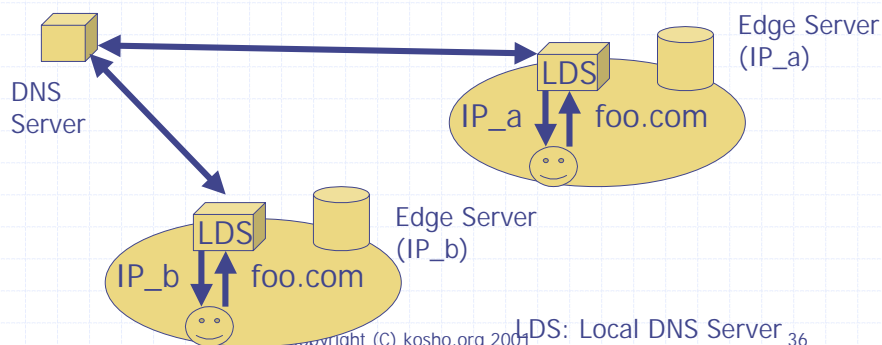
Copyright (C) kosho.org 2001

35

グローバル負荷分散(6)

◆ DNS

- ホスト名をResolveする時に、異なるIPアドレスを返す
 - ◆ ホスト単位のレゾリューション
 - ◆ ローカルDNS単位のユーザ認識
- WWW用としては実績多数



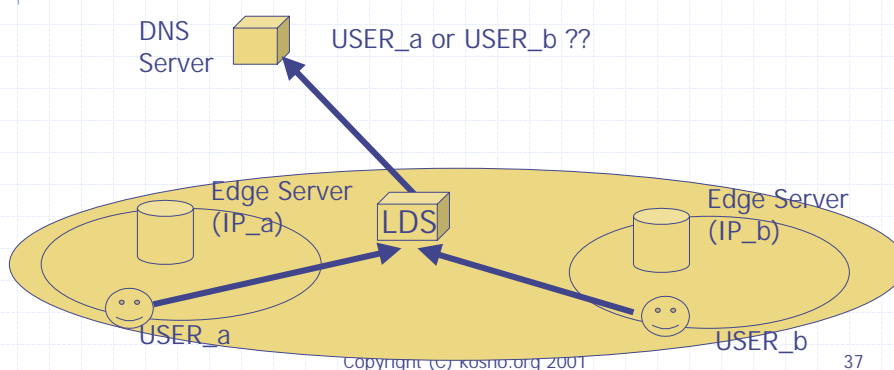
Copyright (C) kosho.org 2001

LDS: Local DNS Server 36

グローバル負荷分散(7)

◆DNSのローカルDNS問題

- ローカルDNS単位のユーザ認識
- 大手ISPでもローカルDNSサーバは数箇所程度
 - ◆ アクセスポイントは数十箇所

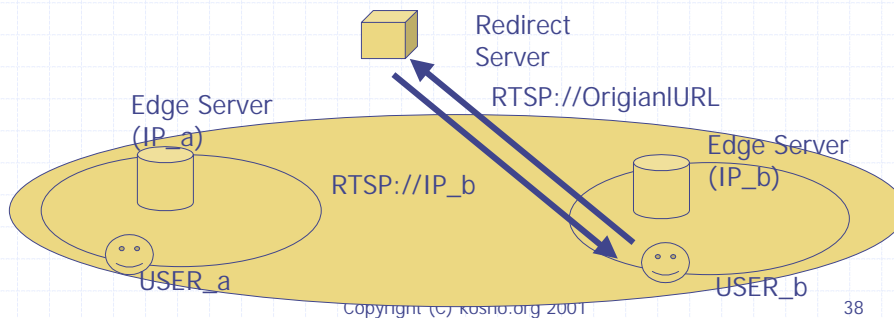


37

グローバル負荷分散(8)

◆リダイレクト、動的生成

- URLの動的生成、リダイレクト
 - ◆ オブジェクト単位のレゾリューション
 - ◆ プレイヤー単位のユーザ認識
- RTSP(制御プロトコル)のリダイレクトをサポートしたプロダクトは少ない



38

グローバル負荷分散(9)

◆リダイレクト、動的生成 (1)

- オブジェクト、ユーザ単位のレゾリューション
 - ◆ 細かな制御が可能
- Redirector負荷
 - ◆ オブジェクト単位にRedirectが発生
 - ◆ ストリーミングの場合、単位時間あたりの視聴コンテンツ数は少ない
- 同一オブジェクトが複数のURLを持つ
 - ◆ Forwardingキャッシュとの相性が悪い

グローバル負荷分散(10)

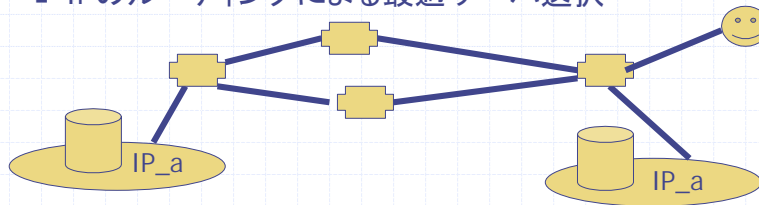
◆リダイレクト、動的生成(2)

- プロービング
 - ◆ プレイヤー毎にプロービングが必要
 - プレイヤーが常時接続されているとは限らない
 - スケーラビリティ問題
 - ◆ グルーピング等が必須
- テーブル作成
 - ◆ アクセスポイントごとのルーティング単位を予めエッジサーバにバインディング
 - ◆ フェイルオーバー規則
 - ◆ ISP内であれば、そんなに困難ではない

グローバル負荷分散(11)

◆ Anycast

- 同一IPアドレスを持つサーバを複数のルーティング単位に設置
 - ◆ 運用技術
- 特別な設備は不要
 - ◆ 運用でカバー
- サーバの負荷分散、フェイルオーバーは困難
 - ◆ 運用でカバー
- IPのルーティングによる最適サーバ選択

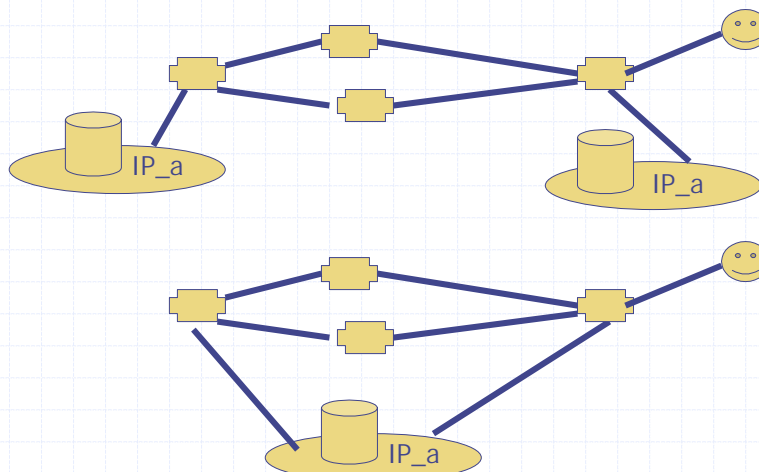


Copyright (C) kosho.org 2001

41

グローバル負荷分散(12)

◆ Anycast



Copyright (C) kosho.org 2001

42

グローバル負荷分散(13)

◆ 比較

	DNS	Redirect	Any Cast
レゾリューション粒度	サーバ	オブジェクト	サーバ
ユーザ認識粒度	ローカルDNS	プレイヤー	ルーティング単位
システム負荷	△	×	○
負荷分散、フェイルオーバー	△	○	×
スケーラビリティ	○	△	○
用途	広域 (ISP間)	閉域 (ISP内)	イベント

Copyright (C) kosho.org 2001

43

グローバル負荷分散(14)

◆ CDNピアリング

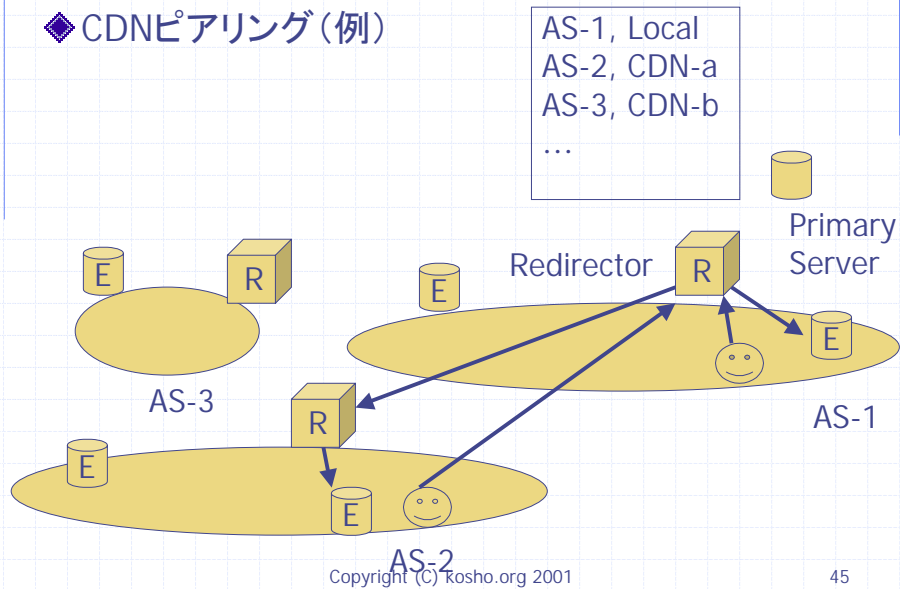
- 複数のCDNの結合
 - ◆ カバー率の拡大
- 階層的グローバル負荷分散
 - ◆ テーブルベース
 - DNS
 - リダイレクション
 - DNS+リダイレクション
 - ◆ メトリック
 - 地理的
 - 負荷的

Copyright (C) kosho.org 2001

44

グローバル負荷分散(15)

◆CDNピアリング(例)



Part 4 オンデマンドCDN

オンデマンドCDN(1)

◆ ストリーミングの特徴(WWWとの比較)

- 基本的に複製を許さない
 - ◆ メディアが高価値
 - ◆ Digital Rights Management
- 不完全性
 - ◆ 部分的再生
 - ◆ パケット落ちを許す配信
- 長時間再生
 - ◆ コンテンツを再生している時に、そのコンテンツが更新される可能性
 - ◆ リアルタイムの行動履歴回収
- マルチレート転送
 - ◆ プレイヤーの状況によりエンコーディングレートが変化

Copyright (C) kosho.org 2001

47

オンデマンドCDN(2)

◆ RFC2326

- ◆ Real Time Streaming Protocol (RTSP)
- HTTPと同様のキャッシュ管理
 - ◆ Cache-Control:
 - ◆ SETUP (RSTP命令)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のキャッシュが重要
 - WWWでもメタ情報のキャッシュは行っている
 - ◆ 連続メディア
 - 保持しているオブジェクトを配信している時は、そのオブジェクトのDescriptionを返す
 - 通過したチャンクだけをキャッシュ

Copyright (C) kosho.org 2001

48

オンデマンドCDN(3)

◆RFCドラフト(Expired)

- 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)
- オープン問題
 - ◆ コピープロテクション
 - ◆ アクセス・アカウントティング
 - ◆ ユーザ認証

オンデマンドCDN(4)

◆Real Proxy

- アカウントティング
 - ◆ 制御チャンネルを常に中継
 - ◆ キャッシュされているコンテンツもアカウントティングを行う
 - 配布権はストリームサーバで管理
 - 全てのLOGはオリジナルサーバにリアルタイムで中継
 - ◆ オリジナルサーバへの制御チャンネルが切れると、コンテンツの配送を中止
- オブジェクトの転送
 - ◆ 転送用プロトコル(TCP)を使用
 - 一定の領域単位の完全な転送
 - コンテンツの途中からの再生も可能

オンデマンドCDN(5)

◆ キャッシュか部分ミラーか(1)

- Redirectionによるレゾリューション
 - ◆ コンテンツ単位のレゾリューション
 - ◆ 部分ミラーが可能に
 - 同一オブジェクトが複数のURLを保持
- DNSによるレゾリューション
 - ◆ サーバ単位のレゾリューション
 - ◆ 部分ミラーは不可能
 - 全オブジェクトを持つ
 - キャッシュ型
 - 存在しないオブジェクトはオリジナルサーバから取得

オンデマンドCDN(6)

◆ キャッシュか部分ミラーか(2)

- キャッシュ
 - ◆ 技術的に発展途上
 - ◆ オブジェクトは自動的に複製される
 - 制御が不要
 - ◆ 制御コネクションが中継される
 - リアルタイムにアクセスLOGがプライマリサーバに集まる
- 部分ミラー
 - ◆ 技術的には枯れている
 - ◆ オブジェクトの複製は明示的に行なう
 - ◆ エッジサーバ上のアクセスLOGの回収が必要

◆ マルチキャストによるオンデマンド

- データを上手く分散させて、何時からでも再生可能

Part 5 ライブCDN

ライブCDN(1)

- ◆ アプリケーション層ルーティング、QoS
 - ネットワーク層のルーティング、QoS
 - ◆ 自立・分散アルゴリズム
 - ◆ 複雑なポリシーの実現、俊敏な変更は困難
 - メトリック、アナウンス経路の変更
 - 複数のAS管理者の合意が必要
 - スプリッタ単位のラフな経路制御、QoS
 - ◆ ストリーミングのみの制御
 - ◆ 中央制御
 - 変更の俊敏性

ライブCDN(2)

◆ 冗長配送

- QoS機能のないInternet
 - ◆ 届けるためには、冗長性(帯域の無駄)を避けない
 - 配送コスト(ハウジング費用)も増加する
- スプリッタ上でのバッファリング
 - ◆ サーバへの欠落パケットの再送要求
 - ◆ プレイヤーからの再送要求へのパケット再送
- 冗長化
 - ◆ 経路、サーバ
 - ◆ パケット

Copyright (C) kosho.org 2001

55

ライブCDN(3)

- 経路の冗長化
 - ◆ スプリッタで結合
 - パケットを一定時間バッファ、シーケンス番号で判別
 - ◆ シームレスな切替
- エンコーダの冗長化
 - ◆ 異なるソース、コンテンツ
 - シーケンス番号が同じでも中身が異なる
 - ◆ タイムアウト、再バッファリングが切替時に発生



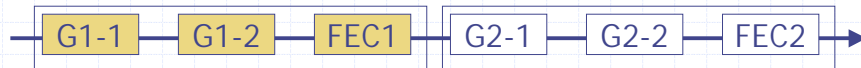
56

ライブCDN(4)

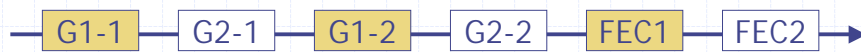
◆ パケットの冗長化

- Forward Error Correction (FEC)

- ◆ 冗長パケットの付加



- ◆ グループ内のパケットが欠落しても冗長パケットにより復元
- ◆ 連続欠落対策
 - パケット送付の順番を混ぜる
 - バッファで回復



Copyright (C) kosho.org 2001

57

Part 6 動的ストリーム

Copyright (C) kosho.org 2001

58

動的ストリーム(1)

◆ Internetの強み

パーソナライズ、インタラクティブ

◆ エッジサーバ上での動的生成が必要

- CDNの必要性はWWWと同じ
 - ◆ オリジナルサーバからのストリーム配信では、プレイヤーに届かない
- 基本は、モバイルコード
 - ◆ セキュリティ管理
 - ◆ リソース管理
 - ◆ これらを考慮した言語、ライブラリ仕様

動的ストリーム(2)

◆ WWWとの違い

- WWWページの動的生成は容易
 - ◆ CPU負荷は軽い
 - ◆ 不連続メディア
- ストリームにおける動的生成は困難
 - ◆ 分類が必要
 - 時間軸上パーソナライズ
 - 画面上パーソナライズ
 - インタラクティブ

動的ストリーム(3)

◆ 容易な時間軸のカスタマイズ

- 番組中へのCMの挿入、ストーリーの選択

- プレイリストによる切替
 - ◆ バッファリングが発生
- スプリッタ上でのストリームの切替
 - ◆ ストリーミングプロトコル上でのマーキング
 - ストリーム間のI-Frameの同期

動的ストリーム(4)

◆ 困難な画面上のカスタマイズ

- 画面上へのCMの挿入、カスタム画面

- 画面の編集には、圧縮を戻す必要がある
 - ◆ 高負荷
 - ◆ 圧縮の非可逆性
- オブジェクト化
 - ◆ MPEG4
 - 複数のストリームの合成(同期)は不可能
- プレイヤーでの合成
 - ◆ Synchronized Multimedia Integration Language (SMIL)等での重ね合せ
 - 再バッファリング
 - ◆ プレイヤー負荷
 - エンコーディングレート調整の必要性
 - ◆ CM抜きプレイヤーの可能性

動的ストリーム(5)

◆ 困難なインタラクティブ

- ストリーミングの問題点
 - ◆ エンコーディング
 - フレーム間圧縮
 - ◆ バッファリング
 - Internetの不安定性のカバー
- アプリケーション層によるQoSの実現
 - ◆ 広域QoS Provider
 - TCP over UDP over IP
 - 帯域に合わせた、エンコーディング、レゾリューションレートの変更

質問、議論

◆ CDS-ML

- CDN、ブロードバンドによる実体化する、次世代のメディア像を議論
- CDS-Tech(技術的话题用ML)も作りました
- <http://www.kosho.org/CDS-ML/>

◆ サーバ負荷分散技術

- Tony Bourke著、鍋島公章 監訳、オライリー・ジャパン