

リクエストを考慮した サーバプッシュ型の音楽サービスの構築

出雲 正尚†, 知念 賢一†, 山口 英†, 尾家 祐二‡

†奈良先端科学技術大学院大学
‡九州工業大学/奈良先端科学技術大学院大学

概要

インターネットの拡大や技術の進歩により大量の情報が流通している。多くの利用者に情報を提供するのにはサーバプッシュの配送形態が最近注目されてきている。しかしながら、既存のサーバプッシュは放送型のサービス形態であり、個々の利用者からの要求が考慮されていない。

そこで、本研究ではサーバプッシュに利用者のリクエストを反映させる機構を追加し、利用者の要求を考慮したサーバプッシュ型の音楽サービスを構築した。本論文では、利用者のリクエストが反映されるサーバプッシュ型の音楽サービスの構築について述べる。

An Implementation of Audio Push-media System with Handling Capability Request

Masanao Izumo†, Ken-ichi Chinen†, Suguru Yamaguchi†, Yuji Oie‡

†Nara Institute of Science and Technology
‡Kyushu Institute of Technology/Nara Institute of Science and Technology

Abstract

Today a huge amount of information is provided on the Internet. The server-push service has become very popular on the Internet because of its ease of providing information for a great number of users. However, the server-push service can not usually satisfy users' requirement, because of its broadcast service.

Therefore, we have developed yet another type of sever-push system for adding users' request handling capability; it selects the information to be provided according to users' request. In this paper, we describe the design and the implementation of the music server-push system as its example, and discuss its performance.

1 はじめに

近年、インターネットの拡大や技術の進歩により、様々なサービスがインターネット上で可能になってきた。インターネット上の動画や音声などのマルチメディア情報の流通も増えてきた。

インターネット上のマルチメディア情報の配送形態には、クライアントプル方式とサーバプッシュ方式の2種類の配送形態がある。クライアントプル方式は、クライアントがリクエストを発行し、サーバから情報を引き出す配送形態である。この方式では利用者毎に細かい制御を行えるという利点があるが、サーバは全てのリクエストに対応しなければならないため、ネットワークの帯域やサーバのコンピュータに与える負荷の面から、マルチメディア情報を多数の利用者に配送することは困難である。一方、サーバプッシュ方式はサーバが主導となり情報提供者側からクライアントに情報を配送する形態である。この方式では、サーバ側で配送のコントロールができるため、多くの利用者へ大量の情報を提供するのに適している。インターネットの利用者が増えているため、サーバプッシュ方式の配送形態が注目を浴びており、研究・開発が盛んに行われている [1]。

従来のサーバプッシュ方式で提供されるサービスの多くはテレビや新聞のような情報提供者から利用者への一方方向の放送型のサービスであった。放送型のサーバプッシュでは利用者の嗜好への配慮が乏しく、利用者側ではチャンネル毎に用意された情報を取捨選択し、自分の嗜好にあわせた情報を選択する形態にとどまっていた。

一方、コンピュータネットワークでは双方向のインタラクティブな情報の配送が可能であり、この特徴を生かすことにより、単なる一方方向の放送だけでなく、サーバから提供される情報について利用者からのフィードバックのある制御が可能である。言い換えれば、従来のサーバプッシュ方式のサービスに対して、ユーザとのインタラクティブ性を付加することによって利用者の意図に合わせたサーバプッシュ方式のサービスが実現できる。

そこで、本研究では、次世代のサーバプッシュ型サービスに求められるべき利用者とのインタラクティブな通信に基づいたサービス制御機構を提案し、その機構の有効性を実証するための実証システムとして、利用者のリクエストを考慮した音

楽の配送サービス JukeCast を構築した。

2 サーバプッシュ

一般に、インターネット上の情報サービスはクライアント・サーバモデルに基づいて構築されている。クライアントがリクエストをサーバに発行し、情報を取得する配送形態をクライアントプル (client pull) と呼ぶ。クライアントプルで情報を提供した場合は、利用者の嗜好に合わせて情報を提供することが可能である。クライアントプル型の情報の配送形態の主な例には WWW がある。

ところがインターネット上で流通される情報の量やその利用者数の増大によりさまざまな問題が生じてきた。利用者の増大により情報を提供しているサーバへのアクセスが集中し、ネットワークの帯域やサーバの処理能力の限界に達してきていることは、現在のインターネットにおいて最大の問題である。

このような現状をうけ、現在ではサーバプッシュと呼ばれる配送形態が研究・開発されている。サーバプッシュではサーバが主導となり情報を選択し配送する。クライアントプルと異なり、サーバ側の都合で情報を配送できる。利用者は情報を探さなくても、サーバ側で選択された情報を受け取ることができる。既存のサーバプッシュのサービスには、PointCast[2] や Marimba/CastaNet[3] などがある。

3 システム設計と構成

本研究では、サーバプッシュの配送形態において、クライアントとサーバとの間での協調動作を実現するための機構を開発している。その第一歩として、クライアントの要求をインタラクティブにサーバ側に反映させるサーバプッシュ型のシステムを考察し、そのプロトタイプとして音楽の配送システム JukeCast を実装した。

JukeCast システムの目標は、常にネットワーク上に音楽を配送し続け、利用者が特にリクエストを出さなくても、JukeCast のクライアントを起動するだけで、音楽を聴き続けることができることにある。また、利用者からのリクエストを受け付けることによって利用者の嗜好や流行を選曲に反映させる。

3.1 要求仕様

JukeCast システムを構築する際に、以下の項目を考慮した。

- 提供者側による自動選曲
- 利用者からのリクエストの反映
- 多数の利用者に対してサービス可能であること
- 高品質に音楽を配送できること

以下、これらの項目を達成するための機構や手法について述べる。

3.1.1 自動選曲と利用者からのリクエスト

利用者が特に行動を起こさなくても自動的に選曲するために、提供者側で選曲を行う。しかしながら、提供者による選曲では利用者の好みや流行に対応することが困難である。一方では、利用者からのリクエストを選曲に反映させなければならないものの、限られた資源では全てのリクエストを処理することはできない。

そこで利用者からの音楽のリクエストを提供者側に集め、リクエストの多い音楽を優先的に配送するようにする。

JukeCast は音楽を流し続けるシステムのため、リクエストがない場合も利用者へ音楽を配送する。

3.1.2 多数利用者のサポートと高品質配送

本システム JukeCast は、個人としてシステムを利用するのではなく多数の利用者で利用できることを目標としている。利用者毎にユニキャストを用いて音楽を配送したのでは同時にサービスできる利用者数は帯域に制限されてしまう。そこで、UDP ベースのマルチキャストを用いて利用者全員に同じ音楽データを配送する。

UDP は信頼性のないトランスポートプロトコルであるため、送信したパケットが通信路上で紛失し、音質が劣化する可能性がある。そこで本システムでは、パケットの紛失を考慮した音楽データ配送方式(3.3 章参照)を用いて、音楽データの品質劣化を改善している。

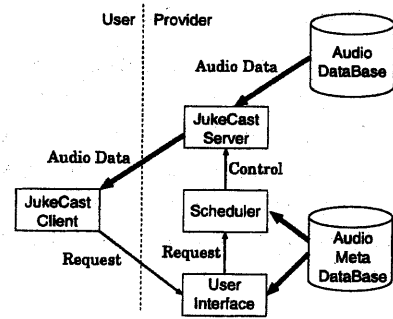


図 1: システム構成

3.2 構成

図 1 に JukeCast の構成を示す。図に示すように JukeCast システムは主に以下のモジュールから成り立っている。

- データベース群
- JukeCast サーバ
- JukeCast クライアント
- ユーザインターフェース
- スケジューラ

3.2.1 データベース群

音楽をサービスするためには、流される曲についての幾つかのデータが必要である。このデータを利用するために、JukeCast では、音楽の曲のデータそのものが入っている Audio Database と、曲に関する情報を蓄えた Audio Meta Database の 2 種類のデータベースを用いている。

Audio Meta Database では、曲のタイトルや作詞・作曲家といった曲の情報と、実際の曲のデータとの対応をとっている。

3.2.2 JukeCast サーバ

JukeCast サーバは Audio Database から曲のデータを読み出しながら、リアルタイムにそのデータをネットワーク上に配送している。配送する際にはデータをマルチキャストパケットに分割しながらネットワーク上に配送する。どの曲を配送するかの指示はスケジューラから受け取る。

3.2.3 JukeCast クライアント

JukeCast クライアントは JukeCast サーバから受け取ったマルチキャストパケットを組み立て、音楽を再生する。JukeCast クライアントは受信用のバッファキューを持っており、ある一定時間を過ぎても到着しないパケットは紛失したとみなす。3.3 章で後述するようにパケットの紛失した部分を復元する。

3.2.4 ユーザインターフェース

利用者は、聴きたい曲をユーザインターフェースを介して提供者側にリクエストをすることができる。利用者は、ユーザインターフェースを通して自分の欲している曲を探し、その曲をリクエストすることができる。リクエストした曲はスケジューラによって管理される。

3.2.5 スケジューラ

音楽の選曲処理と実際のデータの配送処理を分離して、処理の分担を行うために JukeCast サーバとスケジューラを分離した。

スケジューラは利用者からのリクエストを管理している。ユーザインターフェースを介して得られる利用者からのリクエストを保持し、可能な限り多くの利用者が欲している曲を選択するのがスケジューラの役割である。

一般的に、音楽の曲の人気には片寄りがあり、利用者間で流行が発生することが知られている。リクエストの多い曲は、リクエストをしていない利用者にとっても人気のある曲であることが予想される。そのため、スケジューラはリクエストの多い曲を優先的に選択している。

JukeCast サーバが音楽を配送している間、スケジューラは利用者からのリクエストをリクエストバッファにためる。JukeCast サーバが音楽を配送し終わった時点で、リクエストバッファの中からリクエスト数のもっとも多い曲を JukeCast サーバに通知し、新たな音楽を配送する。ここで採択されたリクエストはリクエストバッファから削除され、他のリクエストは次回に持ち越している。以上のようにして、リクエストの多い曲を優先的に配送する機能を実現している。

なお、リクエストがない時は Audio Meta DataBase の中からランダムに曲を選び、その曲の送出を Juke-

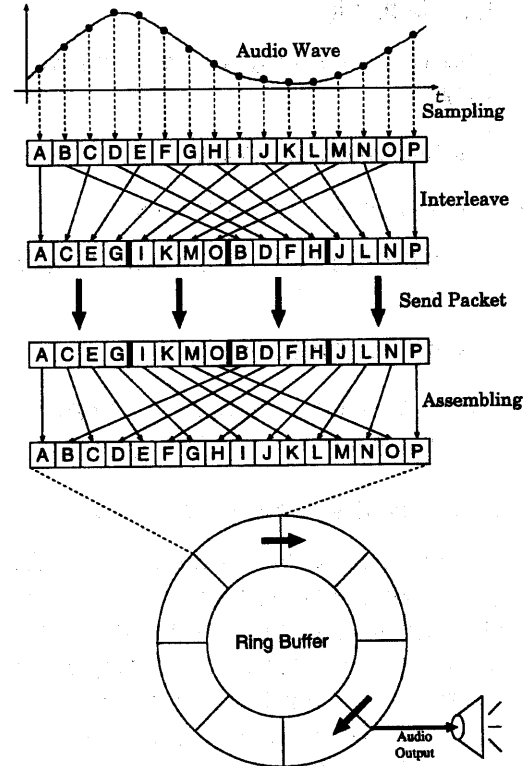


図 2: インターリーブパケット配送方式

Cast サーバに指示している。

3.3 JukeCast における音楽データの配送方式

音楽の配送では品質を損なわずに配送することが重要である。ところが JukeCast は UDP を用いて音楽データを配送しているため、送信したパケットがネットワークの経路の途中で紛失する場合がある。

インターネットの特性として、パケットの紛失はバースト的に発生することが知られている [4]。そこで本システムでは 図 2 に示すように、音楽データのサンプル値をインターリーブして送信している。図中の A, B, ... はデータをサンプリングした値を表している。図 2 において、最大 2 パケットまで連続して紛失したとしても、クライアント側で抜け落ちるサンプルの場所は 1 つ置きになる。この抜け落ちたサンプル値をクライアント側で補完することにより、データ波形があ

JukeCast サーバ	IRIX 5.3
JukeCast クライアント	IRIX 5.3/6.3 DEC OSF1 Solaris 2.5.1
ネットワーク	Ethernet Switch (10 Mbps)
音楽リソース	CD-ROM チェンジャー CD 66 枚, 876 曲
ネットワーク規模	LAN (ホスト数 77)

図 3: 実験環境

る程度復元できる。

例えば、44.1 KHz, 16 bit ステレオ (CD Audio) で音楽を転送する場合、パケットサイズ 1024 byte, インターリーブ長 64 パケットとすると、32 パケット (約 0.2 秒) 連続紛失しても、クライアント側でデータ波形を復元することができる。

またクライアント側では、パケット転送時間のゆらぎを吸収するためにリングバッファを用いている。再生直前に、まだ到着していないパケットは紛失したとみなし、抜け落ちたサンプル値を補完している。

本システムのインターリーブ配送方式では、サーバ/クライアントでの処理が容易で負荷が軽いため、音楽データのリアルタイム転送に向いている。

3.4 実装

JukeCast システムは C 言語と perl で実装されており、合わせて約 22,000 行からなる。サーバ (図 1 の provider 側) は IRIX 5.x/6.x 上で動作し、クライアント (図 1 の user 側) は IRIX 5.x/6.x, DEC/OSF1 (MMS サーバを使用), Solaris 2.5.1 上で動作する。ユーザインターフェースには WWW を用いており、音楽の検索やリクエストは perl で実装された CGI を用いて行うことができる。今回 Audio DataBase には CD-ROM チェンジャーを用い、SCSI を介して CD のデータを読み出している。個々の CD の識別は CD のトラック情報から判断している。

4 評価

表 3 に示す環境で本システムを 6 日間運用した。

20 人程度の人に本システムの実験に参加してもらった。6 日間の運用の結果、約 1800 曲、1 日平均 300 曲配送でき、サービスは安定運用できた。このうち、リクエストによる曲が約 900 曲であった。

本システムにより利用者のリクエストを考慮したサーバプッシュ方式の音楽配送サービスが実現できた。

曲と曲の間には 10 秒くらいの停止時間がある。これは、配送形態上の問題ではなく、曲と曲の CD を入れ替える時間と、クライアント側でのバッファへ貯め始める時間である。データを先読みして配送することにより、この時間を短縮することが可能であるが、今回はまだ実現できていない。

再生中の音楽が一瞬途切れる場合があった。これは主に以下の 2 種類の原因によるものである。

- クライアントが稼働するホストが高負荷の場合、音楽のデータパケットを組み立てる時間が再生に間に合わない。
- 音楽のデータが入っている CD から SCSI を通してデータを読み出しているが、CD によっては読み取りエラーが発生するものがあり、エラー発生時の再読み取りを行っている間、配送が遅れる。

なお、ネットワーク途中のパケット紛失はほとんどなかった。パケット紛失による品質の劣化は被験者には感じられなく、品質の劣化のない配送は達成できた。

5 今後の課題

現在のスケジューリングは過去のリクエスト情報を利用していない。今後は過去のリクエストを考慮した自動選曲も必要であると考えている。

今回は LAN 内の十分な帯域のネットワーク上で運用したが、今後はインターネットへの拡張を計画している。インターネット広域にサービスしようとする、帯域が問題となってくる。そのため、狭い帯域でも利用できるように音楽のデータを圧縮して配送しなければならない。

1 日に配送できる音楽の数は限られているので、リクエストの種類が多い場合は全てのリクエストに応じることができなくなる。リクエストの種類が多い場合には、リクエスト数の少ない曲を破棄するなどして、この問題を解決しなければならない。

6 まとめ

サーバプッシュ型で情報を配送するサービス形態が注目を浴びている。サーバプッシュの配送方式は放送型であり、個々の利用者の要求を満たすことが困難である。そこで、本研究ではサーバプッシュ方式に利用者のリクエストを反映させる機構のプロトタイプとして、利用者のリクエストを考慮したサーバプッシュ型の音楽サービス JukeCast を構築した。

JukeCast では、利用者からのリクエストを反映させるために、本来のサーバプッシュ型の配送に加え、利用者からのリクエストも考慮にいれている。利用者から受けたリクエストを蓄え、リクエストの多い曲を優先的に配送している。本システムを実際に運用し、利用者からのリクエストを反映させたサーバプッシュ型の音楽サービスを行うことができた。

また、マルチキャストを用いることにより、多数利用者にサービスを提供できた。さらに、インターリーブ配送方式により、高品質の音楽を提供できた。

数日運用実験をした結果、このシステムの有用性が示せた。

参考文献

- [1] R. J. Kummerfeld. Server initiated delivery of multimedia information. *International Conference on Worldwide Computing & Its Applications '97*, 1997.
- [2] PointCast Inc. The pointcast network - the leading news and information service on the internet for business and colleges. <http://www.pointcast.com/>.
- [3] Marimba Inc. Castanet. <http://www.marimba.com/products/>.
- [4] Jean-Chrysostome Bolot. End-to-end packet delay and loss behavior in the internet. In *ACM SIGCOMM93*, volume 23, Number 4, pp.289-298, Sep. 1993.