2Z－4

MKngプロジェクトにおけるマルチメディア技術: 動的QOS制御のための資源交渉手法の提案†

河内谷 清久仁　徳田 英幸
日本アイ・ビー・エム（株） 東京基礎研究所　慶應義塾大学 環境情報学部

1 はじめに

慶応大学を中心に行われている次世代マイクロカーネル(MKng)研究プロジェクトでは，Real-Time Machマイクロカーネルをベースに各種処理のための拡張を行っている[1]．このプロジェクトのサブテーマの一つに，マルチメディア処理の動的QOS制御のためのサポートがある[2]．

マルチメディアデータを計算機上で扱う場合，各メディアのもつ時間的制約を守りながら処理を進める必要がある．そのためには，CPUやネットワーク等の計算機資源を確保し処理の保証を行わなければならない．しかし，複数のマルチメディア処理を同時に行う場合等，必要十分な資源が確保できない状況も生じ得る．このような場合，マルチメディア処理のようなソフトリアルタイム処理では，使用可能な資源量に応じて処理のサービスの質（QOS: Quality of Service）を動的に変更し，時間制約を守りつつ処理を継続することが可能である．

しかし，このような動的QOS制御を実現するためには，アプリケーションレイヤが必要な資源量を見積もり，システムと交渉していける新しい資源管理フレームワークが必要となる．本稿では，そのようなフレームワークの実現にあたって考慮すべき問題点について述べ，それらを解決するための資源管理モデルについて提案する．

2 動的QOS制御のための問題点

複数のマルチメディア処理のQOSを動的に調整するためのフレームワークとして，我々は「QOSチケットモデル」というものを提案し，プロトタイプの作成や各種の実験を行ってきている[3, 4]．これにより，このモデルは扱う資源が一つでそれに対するQOS調整が連続的に行える場合には有効にはたらくことがわかっている．しかし，プロトタイプを拡張していくにあたり，以下のような問題点が新たに判明してきた．

1. 複数の関連する資源の扱い

   通常，複数の資源の使用量には何らかの関連がある．例えば，高いQOSの処理を行いたい場合には，CPUだけでなくその他の資源も比例して多く必要になる場合が多い．一方，ネットワークの使用帯域を減らすためにデータを圧縮すると，CPU資源は多く必要になるといった逆の相関がある場合もある．このような資源間の関連を考慮せずに独立して資源を割り当てても，実際には資源をうまく使い切れない．

2. 離散的QOS制御可能点

   一般に，QOSの変更は必ずしも連続的に可能だとは限らない．例えば動画の処理において解像度を変える場合等，フルサイズ，1/2サイズ，1/4サイズという風に，離散的にしかQOSが変更できないことが多い．このような状況では，中途半端な資源がもらえてもあまり意味がなく，制御可能点に適合しない分の資源は無駄になってしまう．

3. 資源の要求指定と配分の手法

   QOSチケットモデルでは資源の要求指定に「範囲」を使っていたが，これでは上で述べたような資源間の関係やQOS制御可能点を指定するには不十分で，新しい指定方法が必要になる．また，指定された制約を満たすように複数資源を複数のアプリケーションに配分する手法も開発する必要がある．

4. 必要な資源量の見積もりと較正

   あるQOSで処理を行うのに各資源がどれくらい必要かは，ハードウェア性能や処理の性質等によって異なるため，あらかじめ完全に見積もることは難しい．そのため，いかにして初期値を見積もり，資源要求を随時補正していくかが問題となる．更に，この処理は複数の離散的QOS制御可能点のそれぞれについて行われなければならない．

3 新しい資源管理手法による問題解決

上記の四つの問題点を解決するため，我々は資源管理モデルの改良を行っている[5]．改良の基本方針は以下のとおりである．

- 資源要求の指定方法の改良（問題1，2）
- 資源配分アルゴリズムの改良（問題3）
- 資源見積もり/補正アルゴリズムの改良（問題4）

以下，これらについて順に述べる．

3.1 資源要求表の導入

まず，問題1（複数関連資源）に関しては，資源間の関係を何らかの式で表し指定するという方式も考えられる．しかし，問題2（離散的QOS制御可能点）を考慮した場合，資源量が連続的に変えられることには実はあまり意味がない．そこで，資源の要求方式として必要な資源の範囲を個別に指定するのではなく，複数のQOS制御可能点（「QOSタイプ」と呼ぶ）ごとの必要資源量を指定するように変更を行った．

表1は，そのような指定を行う「資源要求表」の例で，三つのQOSタイプ1，2，3とそれぞれの資源要求が指示されている．表中の「Q値」は，対応するQOSタイプで処理が行えた場合のアプリケーションの「満足度」を表している．これは，各QOSタイプにおける処理の品質そのもの（大きいほうが高い）と考えることもできる．3.3節で述べる資源の配分アルゴリズムでは，なるべくQ値が大きくなるようなQOSタイプを選ぼうとする．

3.2 「資源チケット」アブストラクション

必要な資源の記述を従来の各資源独立の範囲指定から，このような表形式に変更することで，前章で述べた問題1と2には対応することができる．図1が，その改良を組み込んだ新しい資源管理モデルである．このモデルでは，アプリケーションがシステム資源の要求を行ったり，利用可能な資源量等の情報を得るための統一的アブストラクションとして「資源チケット」というものを設けている．このアブストラクションが，「資源Aware

表1: 資源要求表の例

QOSタイプ	CPU (%)	メモリ (MB)	ディスク (Mbps)	ネット (Mbps)	Q値
1	50	10	3	3	100
2	70	20	3	1	80
3	30	10	2	2	50

Multimedia Technology in the MKng Project:
A Resource-Negotiation Method for Dynamic QOS Control
Kiyokuni KAWACHIYA[1] and Hideyuki TOKUDA[2]
[1]IBM Research, Tokyo Research Laboratory
1623-14, Shimotsuruma, Yamato, Kanagawa 242, Japan
E-Mail: <kawatiya@trl.ibm.co.jp>
[2]Faculty of Environmental Information, Keio University

†この研究は，情報処理振興事業協会（IPA）が実施している創造的ソフトウェア育成事業「次世代マイクロカーネル研究プロジェクト」のもとに行われた．

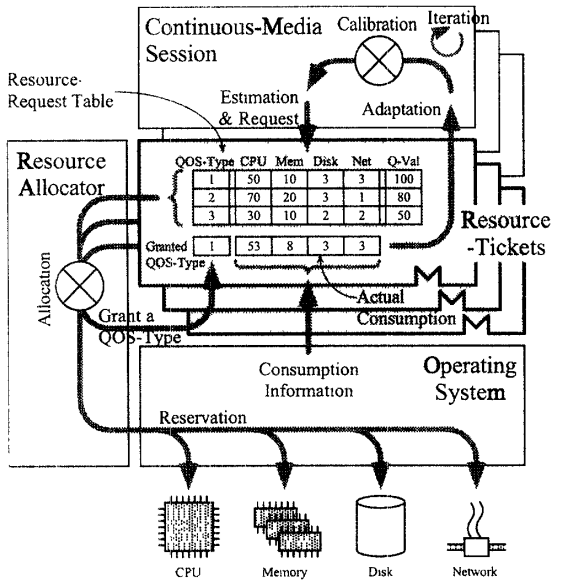

図 1: 資源チケットアブストラクション

な」プログラムを作るための最下位レベルのインタフェースとなる．

資源チケットは，実行中のアプリケーションごとに用意される．アプリケーションは実行時に，前節で述べた資源要求表を自分の資源チケットに登録する．複数の資源チケットの資源要求表をもとに,「資源アロケータ」が，なるべく多くのアプリケーションが満足できるような資源の配分を決定し，実際の資源予約を行う．どの QOS タイプ用の資源割り当てが行われたかは，資源チケットに書き戻される．なお，この処理は資源要求表に変更が生じるたびに動的に行われる．

各アプリケーションは，指定された QOS タイプに基づき，割り当てられた範囲内で資源を消費して，処理を行う．予約した資源内での処理の保証は，OS によって行われる．OS は同時に，各アプリケーションがどれくらい実際に資源を使用しているかを集計しており，その値は資源チケットに書き戻される．このフィードバック情報をもとに，各アプリケーションは資源要求の見直し（較正）を行い，資源要求表を更新していく．

このモデルを実現するためには，問題 3（資源の配分手法）と 4（資源の見積もりと較正）の解決が必要になる．前者は図中の資源アロケータ，後者は各アプリケーション自身（もしくはライブラリー）によって行われる．

3.3 複数関連資源の配分手法

資源アロケータは，複数のアプリケーションからの資源要求表をもとに，なるべく多くのアプリケーションが満足出来るような資源の配分を決定する．具体的には，選ばれた QOS タイプの Q 値の和が最大になるような資源の配分を求めている．

この配分問題は，複数の選択候補を持つ複数次元ナップサック問題として定式化でき，NP 完全である．我々の資源アロケータでは整数計画問題のためのヒューリスティックを導入することで解決を行う．導入するヒューリスティックとしては，ラグランジュ弛緩法やランダマイズドラウンディングに基づく局所探索法があげられる [6, 7]．

この資源配分のアルゴリズムは，資源チケット，各アプリケーションや OS 等の他のコンポーネントとは独立して変更可能であり，様々な資源管理ポリシーを導入することも可能である．

3.4 資源の見積もりと較正手法

各アプリケーションは，資源チケットから得られるフィードバック情報等をもとに，必要資源量の見積もりと補正を行う．これには，マルチメディア処理の「繰り返し性」を利用する．マルチメディア処理は一般に，同種の処理（例えば動画の 1 フレームの表示）が繰り返し行われるという特性をもつ．このため，初期の見積もり値としては適当な値を指定しておき，1 回あるいは数回の繰り返しが終わるたびに，実際に使用した資源量を見てその値を資源要求表に書き戻すという処理を繰り返すことで，資源要求を「較正」していくことが可能である．

以下に，この較正のための基本的アルゴリズムを示す．

1. 最初は，資源要求表は空である．まず，最も Q 値の大きい（つまり，最も望ましい）QOS タイプから交渉を始める．
2. その QOS タイプを資源要求表に追加登録する．資源要求の初期見積もり値としては，すべて 0（もしくは小さな値）を指定する．その結果，資源アロケータからはこの QOS タイプが指定される．
3. 指定された QOS タイプで，1 回あるいは数回の繰り返し処理を行う．資源チケットを見ると実際に消費した資源量がわかるので，その値をもとに資源要求表のエントリを修正（較正）する．
4. 資源アロケータから，その QOS タイプが依然として指定されている（つまり，資源が予約できている）場合は，処理を繰り返しつつ，較正を続ける．
5. もし，これまでに登録したどの QOS タイプの資源要求も満たされない場合，次に高い Q 値をもつ QOS タイプをインクリメンタルに追加して，ステップ 2 以降の較正処理を行う．

このアルゴリズムでは，最も高い Q 値をもつ QOS タイプから順に，補正しつつ資源要求の交渉を行っていくことになる．資源要求が満たされない場合は，順次低い Q 値の資源要求を追加していく．もし，処理中にシステム状況が変化し，高い Q 値用の資源要求が再び満たされるようになった場合，アプリケーションはその QOS タイプの処理に移行し，ステップ 4 の較正処理を続けることになる．

4 おわりに

複数のマルチメディア処理に対する資源配分と QOS 調整をシステムの環境に応じて動的に行っていくためには，資源の予約と適応を組み合わせた新しい資源管理モデルが必要である．そのようなモデルの実現にあたっては複数の関連する資源の扱いや，離散的な QOS 制御可能点の問題等を解決する必要がある．本稿では，これらを解決したモデルとして,「資源チケット」という統一的アブストラクションを使う方式を提案した．

各アプリケーションは，資源チケットに自分の対応可能な QOS タイプごとの資源要求を「資源要求表」として登録する．資源アロケータが，この表をもとに各アプリケーションに対する資源配分を決定し予約を行う．各アプリケーションは，指定された QOS タイプで処理を行いつつ，実際に消費した資源量の情報をもとに資源要求表を較正していく．このような資源管理手法により，環境に依存せず動的な QOS 制御を行える資源 Aware なアプリケーションを開発することが可能になる．

参考文献

[1] 徳田 他: “MKng: 次世代マイクロカーネル研究プロジェクトの概要,” 第 55 回情処全大論文集, 1Z-2 (1997).

[2] 河内谷, 徳田: “MKng プロジェクトにおける動的 QOS 制御サポート,” 第 53 回情処全大論文集, 5B-8, pp. 1-47–1-48 (1996).

[3] K. Kawachiya and H. Tokuda: “Dynamic QOS Control Based on the QOS-Ticket Model,” *Proc. IEEE ICMCS '96*, pp. 78–85 (1996).

[4] 河内谷: “マルチメディア処理の動的 QoS 制御のためのフレームワーク,” 信学会和文論文誌 B-I, Vol. J80-B-I, No. 6, pp. 465–471 (1997).

[5] K. Kawachiya and H. Tokuda: “A Negotiation-Based Resource Management Framework for Dynamic QOS Control,” *IBM Research Report, RT0192*, IBM (1997).

[6] 徳山: “ランダムアルゴリズムの話題から,” 信学会誌, Vol. 77, No. 9, pp. 957–967 (1994).

[7] R. Motwani and P. Raghavan: *Randomized Algorithms*, Cambridge University Press (1995).