マルチメディア通信と分散処理　92－21
（１９９９．２．２６）

情報ネットワークシステムのポリシー制御 POLICYCOMPUTING™に関する一検討

田中　俊介　菅野　政孝　小熊　慶一郎　松田　栄之

株式会社 ＮＴＴデータ　技術開発本部

〒135 東京都江東区豊洲 3-3-3 豊洲センタビル
Tel: 03(5546)9571　Fax: 03(5546)9572
Email： shun@open.rd.nttdata.co.jp

あらまし:
クライアントサーバシステムでは，クライアント(端末)での処理内容が多く，サーバが分散されているため，システムの運用管理コストが大きいという問題がある．本稿では，マスターポリシー(管理者にとって設定が行いやすい表現形式のポリシー)によって情報システム内の全てのリソース(機器，サーバプロセス，ユーザなど)を一元的に管理するための仕組み POLICYCOMPUTING™ を提案する．本方式では，管理者がマスターポリシーの設定を行うと，マスターポリシーから個別ポリシー(各リソースの個別の設定)を自動的に生成し，各リソースに設定を反映させる．

キーワード:　ディレクトリ，ネットワーク管理，ポリシー

POLICYCOMPUTING™, a Proposal about Managing Information Network System with Policy

Shunsuke Tanaka,　Masataka Sugano,　Keiichiro Oguma,　Shigeyuki Matsuda

NTT DATA CORPORATION
Research and Development Headquarters

Toyosu Center Bldg., 3-3-3, Toyosu, Koto-ku, Tokyo, 135, Japan
Tel: 03(5546)9571　Fax: 03(5546)9572
Email: shun@open.rd.nttdata.co.jp

Abstract:
It is a problem in the Client/Server System that the management cost is heavy, because of complex applications of clients and distributed servers. We propose POLICYCOMPUTING™, that manages all resource (hardware, service-program, user-account, etc.) composing the information system with Master-Policies (policies written in user-friendly language) at a single point. If the manager registers Master-Policies, our system produces Particular-Policies and controls all resource under Particular-Policies.

Key words:　Directory, Network Management, Policy

1. はじめに

クライアントサーバ系システムはホスト系システムと比較して運用管理のコストが大きくなるという問題点が指摘されている[1,2]. コストが大きくなる理由としては以下のような点があげられる.

・クライアント端末の設定項目数が多い.

・端末毎に別々の設定をしなければならない設定項目が多い(例: ホスト名).

・機能毎にサーバを分けるため, 同一の設定内容を複数のサーバに配布しなければならない(例:ユーザアカウント).

・サーバを階層的に配置するため, 目的のサーバを探し出せる仕組みが必要である(例: Web サーバ).

また, 管理できる人材が不足しており, 一部の人に管理業務を集中的に行わせざるを得ないこともコストを増大させる一因である[3].

運用管理コストを削減するためには多様なアプローチがあるが, 我々は「各リソース(機器, サーバプロセス, ユーザなど)の設定に必要な運用管理コスト」に注目する.

「各リソースの設定」を簡単に行えるようにする方法として DEN がある. DEN の仕組みについて第2章で紹介する. また, 第2章では DEN を利用しても削減できない「各リソースの設定に必要な管理コスト」を指摘する.

第3章以降で, POLICYCOMPUTING™ を提案し, DEN を利用しても削減できない「各リソースの設定に必要な管理コスト」を削減する仕組みについて述べる.

2. DEN

2. 1. DEN とは

DEN とは Directory Enabled Network の略である. 各リソースが個別に持っていた設定内容(ポリシー)を全てディレクトリサーバ内に格納することで, きめ細かいシステム管理を簡単に実現しようという概念である.

DMTF と IETF が共同して, ディレクトリのスキーマ(どの情報をディレクトリサーバ内のどこに格納するか)に関する標準仕様を作成している[4,5]. また, ディレクトリサーバと直接通信ができないリソースにディレクトリの情報を反映させるための仕組みについて Croll らが提案を行っている[6].

2. 2. DEN によるネットワーク管理の仕組み

DEN によるネットワーク管理の仕組みを図1に示す.

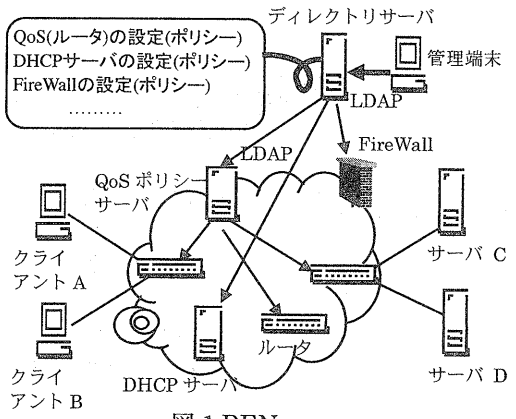

図 1 DEN

各リソースには必要最小限の設定(ディレクトリサーバのホスト名など)だけがなされており, その他の設定情報(ポリシー)はディレクトリサーバ内に格納されている. 各リソースはディレクトリサーバ内に格納された自分のポリシーを参照して動作するようになっている. 各リソースは LDAP[7]を利用してディレクトリサーバと通信する.

ディレクトリサーバと直接通信できない(LDAP に対応していない)リソースにはポリシーサーバーを利用してディレクトリの情報を反映させる. ポリシーサーバは, ディレクトリサーバに格納されたポリシーのうち, 自分が管理しているリソースのポリシーだけを取得し, 取得したポリシーをそれぞれのリソースに配布する[6].

2. 3. DEN による通信帯域制御(例)

DEN についてより詳しく紹介するために, DEN による通信帯域制御[8]について具体的に紹介する (図2参照).

システム構築時には以下の処理を行う.

1) 管理者が QoS ポリシーをディレクトリサーバに格納する.
2) ポリシーサーバは QoS ポリシーをディレクトリサーバから読み出す.

実際にデータフローが流れると, 以下の処理が行われる.

3) 新しいフローのパケットを発見したルータがポリシーサーバに問い合わせを出す(COPS プロトコル[9]を利用する).
4) ポリシーサーバは, 2)で取得したポリシーを参照して, パケットの処理方法を決定する.
5) ポリシーサーバがルータに応答を返す(COPS プロトコル[9]を利用する).
6) ルータが応答に従ってパケットを処理する.

※図 2 ではパケットを転送し，RSVP[10]による帯域確保を行っている．

以上の処理によって，QoS ポリシーで定めた通りの通信帯域が確保される．

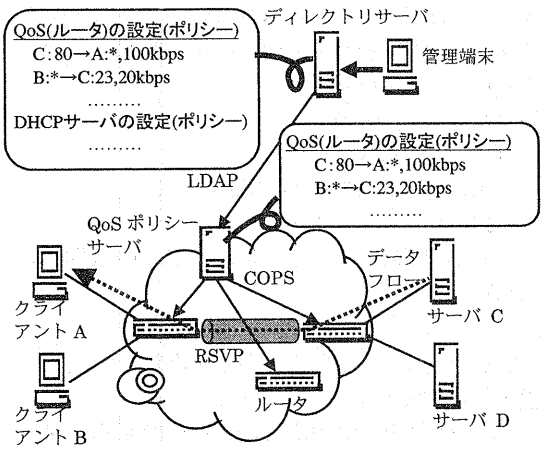

図 2　DEN による通信帯域制御

2. 4. DEN の効果と問題点 (不足点)

DEN を利用すると全てのポリシー（リソースの設定情報）をディレクトリ上で一元管理できるようになる．以下のような点から「各リソースの設定に必要な管理コスト」が軽減すると期待できる．

・同一の情報を複数回入力する必要が無くなる．

・全ての情報に同一の手段でアクセスできる．

しかし，DEN を利用した場合にも，設定する項目の内容自体は変わらないため，以下の 2 つの問題点が残っている．

・設定する項目の数が多い．

・システムの知識が豊富な人でなければ管理できない．

DEN を利用した場合にも「各リソースの設定に必要な管理コスト」は多く残っていると言える．

3. PolicyComputing™ の提案

3. 1. 目的

PolicyComputing™ の目的は，運用管理コストを削減し，少ない運用管理コストでもきめ細かいシステム管理(資源の効果的な活用，セキュリティの向上 など)ができるようにすることである．

3. 2. 方針

PolicyComputing™ では，運用管理コストのうち「各リソースの設定に必要な管理コスト」を削減することを目指す．「マスターポリシー(管理者にとって設定が行いやすい表現形式のポリシー)によって情報システム内の全てのリソース(機器，サーバプロセス，ユーザなど)を一元的に管理」し，以下のような点で「各リソースの設定に必要な管理コスト」を削減していくことを考えている．

・マスターポリシーから個別ポリシーを自動的に作成することから，管理者は少数のマスターポリシーを入力すれば良くなる．

・マスターポリシーは管理者にとって理解しやすい表現形式で記述できることから，システムに精通していない管理者でも十分に管理が行えるようになる．

3. 3. 実施内容

PolicyComputing™ でマスターポリシーに基づく一元管理を実現する仕組みは図 3 のようである．

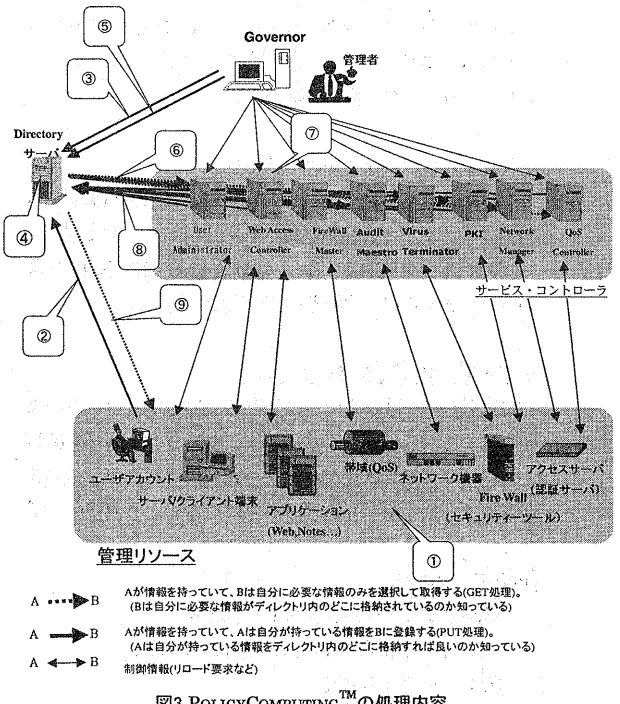

図3 PolicyComputing™ の処理内容

図 3 の①～⑨までのそれぞれの処理内容は以下のようである.

①管理者がリソースに「POLICYCOMPUTING™ に加えるための最小限の設定」を行う.

リソースを POLICYCOMPUTING™ に組み込む場合には, ディレクトリサーバと通信できなければならない. 自分のアドレス, ディレクトリサーバ(もしくはポリシーサーバ)のアドレスなどを設定する.

②各リソースがプロファイルを自動的に格納 する.

各々のリソースは起動時に自分が所持している動作状況をプロファイルとしてディレクトリサーバに格納する. また, 起動中に動作状況に変化が生じた場合には変化分の情報をディレクトリサーバに格納する. プロファイルとは以下のようなデータである.

[プロファイルの例]

・属性: ファイルの情報種別
  内容: ファイルXは人事情報
・属性: 端末にログインしているユーザ
  内容: HostA にユーザYがログインしている

リソースの中には, ディレクトリサーバと直接通信できないものがある. ディレクトリサーバと直接通信できないリソースは, ポリシーサーバなどを介してプロファイルをディレクトリサーバに情報を格納する.

③管理者がプロファイルを格納する.

プロファイルの中には, 管理リソースが自動的(機械的)に格納することができないプロファイルがある. 例えば, ユーザ・アカウントなどである. 自動的(機械的)に格納できないプロファイルは, 管理者が手作業で格納する.

④知識データ が登録されている.

知識データとはシステムに依存しない運用ノウハウのような情報である. あらかじめディレクトリサーバに格納しておき, システム管理者がシステム運用中に変更する必要がないようにする. 知識データとは以下のようなデータである.

[知識データの例]

・属性:アプリケーションのポート番号
  内容: ポート番号N番は IP-Phone である.
・属性:サービスの品質
  内容: IP-Phone は 100kbps の帯域があれば高品質である.

⑤管理者がマスターポリシーを格納する.

マスターポリシーも「属性」と「内容」によって構成されている. 内容は「A = B」という形式に容易に変更できるような文章である. 属性の値によってAおよびBに入る単語の種類を定義しておく(例えば, 「属性」が「文書セキュリティ」である場合はAには情報別とユーザグループの組が入り, Bには閲覧の可否が入る). マスターポリシーとは以下のようなデータである.

[マスターポリシーの例]

・属性: 文書セキュリティ
  内容: 人事情報は管理職のみ閲覧可とする
・属性: サービス QoS ポリシー
  内容: 管理職のコミュニケーション・ツールの通信は高品質にする

⑥マスターポリシー, プロファイル, 知識データのうち必要な情報を取得する.

ディレクトリサーバ内のどの属性の情報を参照するかは, サービス・コントローラ毎に異なっている. サービス・コントローラは自分にとって, どの属性の情報(どの属性のマスターポリシー, どの属性のプロファイル, どの属性の知識データ)が必要であるかを知っている(あらかじめ定義されていて, あらかじめサービス・コントローラに設定しておくものである). サービス・コントローラは自分にとって, 必要な属性のデータのみを取得する.

⑦取得した情報を利用して個別ポリシーを自動生成する.

個別ポリシーも「属性」と「内容」によって構成される. 内容はリソースの詳細な設定情報である. 個別ポリシーとは以下のようなデータである.

[個別ポリシーの例]

・属性: ファイル・アクセス
  内容: ユーザYは xxx にアクセス不可
・属性: ネットワーク QoS
  内容: 端末A-端末B間のポートN番の通信を 100kbps で帯域を確保する

個別ポリシーの作成は, 数式Aに定理Xを適用すると数式Aと等価で表現が異なる数式Bが作成されるという「数式の変換」に似ている. マスターポリシーにプロファイル・知識データを適用して, マスターポリシーと等価で表現が異なる個別ポリシーを作成する. 数式の変換では複数の数式を変換させて 1 つの数式を作成する場合が多いが, ポリ

シーの変換(個別ポリシーの作成)では，少数のマスターポリシーを変換させて多数の個別ポリシーを作成する場合が多くなる(図4参照).

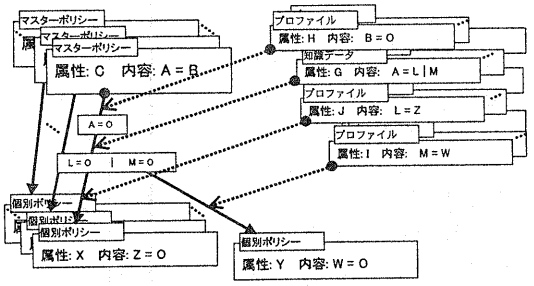

図4 個別ポリシーの生成方法

プロファイルの中には，「端末を利用しているユーザ名」など，現在の動作状況を表すものがある．こうしたプロファイルの内容が変化した場合には，ポリシー変換をやり直して，関係がある個別ポリシーを作成し直す.

どのマスターポリシーからどのプロファイル・知識データをどのように利用して個別ポリシーを生成するかは，サービス・コントローラ毎に異なっている．本稿では，QoS Controller における個別ポリシーの生成方法についてのみ掲載する(図6参照).

⑧個別ポリシー(設定情報)をディレクトリサーバに格納する.

⑦で作成した個別ポリシーをディレクトリサーバの所定の場所に格納する.

⑨各リソースが個別ポリシー(設定情報)を取得する.

リソースは自分にとって，どの属性の個別ポリシーが必要であるかを知っている(あらかじめ定義されていて，あらかじめリソースに設定しておく)．ディレクトリサーバ内のどの属性の情報を参照するかは，リソース毎に異なる(ある属性の情報はルータとFireWallという2つのサービス・コントローラから参照されるが，ある属性の情報は端末からしか参照されない場合もあり，同種類のリソースでも異なる属性の情報を参照する場合もある).

このでは処理は DEN で行われている処理と同じ処理を実行する.

4. POLICYCOMPUTING™ での通信帯域制御(実施例)

POLICYCOMPUTING™ を実際に利用した場合の例として，POLICYCOMPUTING™ での通信帯域制御(通信品質の管理)について説明する(図5参照).

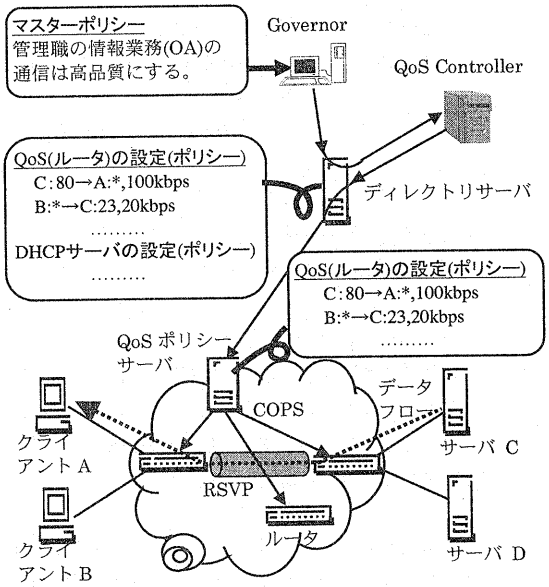

図5 POLICYCOMPUTING™ での通信帯域制御(実施例)

図5と図2の異なる点は，図5に QoS Controller と Governor が追加されている点である.

システム構築時には以下の処理を行う.

1) 管理者はマスターポリシーを入力する.
2) QoS Controller がマスターポリシーから個別ポリシーを生成する(図6参照).
3) ポリシーサーバは QoS ポリシーをディレクトリサーバから読み出す.

実際にデータフローが発生した場合の処理は DEN の処理(図2の処理)と同じである.

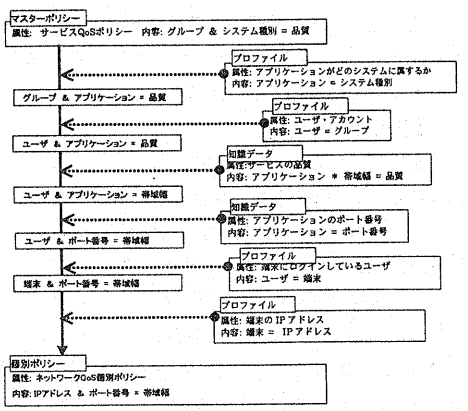

図6 QoS個別ポリシーの生成方法

5. 今後の課題

1) 個別ポリシー間に発生した矛盾の自動解決

マスターポリシー同士に矛盾が無くても，各種サービスマネージャが作成した個別ポリシー同士に矛盾が生じる場合もあると考えられる．矛盾の例としては以下のような例があげられる．

- 64kbps しか帯域が無いのに，5 人の人に 32kbps の帯域確保を許可する必要がある．
- A さんが Web サーバ X の Z というページにアクセスすることを許可したが，FireWall によって A さんは Web サーバ X にアクセスできない．

上記のような矛盾を「全て管理者に通知し，管理者に処理を選択させる」という方法で解決すると管理コストが大きくなると予想される．個別ポリシー同士の矛盾を自動的に解決する仕組みが必要だと考えていきたい．

2) 評価(実際に管理コストをどのくらい削減できるか)

現状のシステムで管理コストがどのくらいかかっているかを測定しておき，POLICYCOMPUTING™ を導入した場合に管理コストをどのくらい削減できるか測定したい．

システムの規模，アカウント更新の頻繁さ，管理者のスキルなどの違いによって削減の度合い変わってくると予想している．そこで，どのような条件ならば効果があり，どのような条件で効果が少ないかについて調査していきたい．

6. まとめ

我々は，大規模なクライアントサーバシステムは運用管理コストが大きいという問題点を指摘し，「各リソースの設定に必要な管理コスト」を削減するという点に着目した．

現在，各リソースの設定に必要な管理コストを削減する仕組みとしては DEN が提案されている．しかし DEN を利用しただけでは「各リソースの設定に必要な管理コスト」の一部しか削減できない．

本稿では「各リソースの設定に必要な管理コスト」を削減することを目標として，マスターポリシーによる一元管理システム POLICYCOMPUTING™ を提案した．POLICYCOMPUTING™ の特徴は DEN を利用した上で，更に，マスターポリシーから個別ポリシー(各リソースの設定項目)の作成ができることである．

今後は，「個別ポリシー間に矛盾が生じた場合の自動解決の検討」と「POLICYCOMPUTING™ の評価(管理コストをどのくらい削減できるかの測定)」を行っていく予定である．

[参考文献]

[1] 大鐘,TCP/IP と OSI ネットワーク管理,SRC,Apr.1993

[2] K. Dec, etc, Management Solutions for the Distributed Computing Revolution, Garter Group Strategic Analysis Report, Jul.1996

[3] 山口,インターネットの今後,情報処理学会連続セミナー'98 第 3 回,Oct.1998

[4] S. Judd ,J. Strassner , Directory-enabled Networks,DMTF Draft, Sep.1998

[5] J. Strassner, E. Ellesson, Terminology for describing network policy and services, Internet-Draft Aug.1998

[6] A. Croll, A. Shivnan, Policy-based networking and the role of directories, 3Com WhitePaper, Mar.1998

[7] W. Yeong, etc,Lightweight Directory Access Protocol, RFC1777, March 1995

[8] R. Rajan,etc., Schema for Differentiated Services and Integrated Services in Networks, Internet-Dfaft, Oct.1998

[9] J.Boyle,etc,The COPS (Common Open Policy Service) Protocol, Internet-Dfaft, Jan.1999

[10] R. Braden,etc, Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification, RFC2205,Sep.1997