

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4283899号  
(P4283899)

(45) 発行日 平成21年6月24日(2009.6.24)

(24) 登録日 平成21年3月27日(2009.3.27)

(51) Int.Cl. F I  
**G06F 13/00 (2006.01)** G O 6 F 13/00 3 5 1 C  
**H04L 12/56 (2006.01)** H O 4 L 12/56 2 0 0 C

請求項の数 8 (全 26 頁)

(21) 出願番号	特願平7-296877	(73) 特許権者	506355419
(22) 出願日	平成7年11月15日(1995.11.15)		エリクソン アーバー
(65) 公開番号	特開平8-272710		Ericsson AB
(43) 公開日	平成8年10月18日(1996.10.18)		スウェーデン, ストックホルム, トーシャ
審査請求日	平成14年8月28日(2002.8.28)		マンズガーデン 23
審判番号	不服2006-8595(P2006-8595/J1)	(74) 代理人	100066728
審判請求日	平成18年5月1日(2006.5.1)		弁理士 丸山 敏之
(31) 優先権主張番号	08/362217	(74) 代理人	100100099
(32) 優先日	平成6年12月22日(1994.12.22)		弁理士 宮野 孝雄
(33) 優先権主張国	米国(US)	(74) 代理人	100111017
(31) 優先権主張番号	08/375624		弁理士 北住 公一
(32) 優先日	平成7年1月20日(1995.1.20)	(74) 代理人	100119596
(33) 優先権主張国	米国(US)		弁理士 長塚 俊也
		(74) 代理人	100141841
			弁理士 久徳 高寛

最終頁に続く

(54) 【発明の名称】 サーバがサービスを比率制御によりエンティティへ提供する時期を制御する方法及びスケジューラ

(57) 【特許請求の範囲】

【請求項1】

N個のエンティティの到着時刻である開始時刻  $s_i$  を記憶する開始時刻メモリ(16) (Nは1以上の整数であり、エンティティはユーザからのセル、 $i$  は1以上N以下の整数であり、 $s_i$  は、 $i$  番目のエンティティがサーバからのサービスを最初に要求する時刻に対応する。);

N個のエンティティの終了時刻  $f_i$  を記憶する終了時刻メモリ(18) ( $f_i$  は、 $i$  番目のエンティティがサーバによりその時刻までにオペレートされるべき時刻に対応する。);

開始時刻  $s_i$  と終了時刻  $f_i$  とを特定できるように時刻の経過を刻む仮想時計(20); 及び

各仮想時刻にて、最も早い終了時刻  $f_i$  と、ある開始時刻  $s_i$  とを有するエンティティが最初にオペレートされ、次に、もし終了時刻  $f_i$  が同じならば、最も早い開始時刻  $s_i$  が、若しくは最も高いID又はレート<sup>10</sup>を有するエンティティが、次にオペレートされる、という順序で、サーバによりオペレートされるべきエンティティを選択して、エンティティを終了時刻  $f_i$  の前にサーバによりオペレートするコントローラ(22)を具え、

該エンティティがサーバによりオペレートされたときは、仮想時刻を1単位分増加し、該コントローラ(22)は他にもし、仮想時刻と同一またはそれより前の開始時刻  $s_i$  のエンティティが存在しなければ、どんなエンティティにもサービスを提供しないで、所定の時間後に現在の仮想時刻を1単位分、増加して、

N個のエンティティが、サーバによってオペレートされる時期を制御するスケジューラ装置。

## 【請求項 2】

コントローラ(22)は、サーバからエンティティへのサービスを連続的に提供する、請求項 1 に記載のスケジューラ装置。

## 【請求項 3】

開始時刻メモリ(16)は、開始時刻優先キューを含み、終了時刻メモリ(18)は、終了時刻優先キューを含む、請求項 2 に記載のスケジューラ装置。

## 【請求項 4】

ユーザと接続され、且つサーバによってオペレートされるためのエンティティを受取る、要素プールを具え、該要素プールにおいて、エンティティの到着時刻  $s_i$  は、 $i$  番目のエンティティが要素プールに到着する時刻に対応し、終了時刻  $f_i$  は、 $i$  番目のエンティティが、要素プールを離れて、サーバによりその時刻までにオペレートされるべき時刻に対応し、且つコントローラ(22)は、要素プールからサーバへエンティティを転送する、請求項 3 に記載のスケジューラ装置。

10

## 【請求項 5】

仮想時計の時刻を制御し、且つ、要素プールにおけるエンティティと、開始時刻優先キュー及び終了時刻優先キューにおける夫々の開始時刻  $s_i$  及び終了時刻  $f_i$  とを記憶するために、コントローラ(22)は、仮想時計、要素プール、開始時刻メモリ(16)及び終了時刻メモリ(18)へ接続される、請求項 4 に記載のスケジューラ装置。

## 【請求項 6】

コントローラ(22)は、最も早い終了時刻  $f_i$  を有するエンティティが唯 1 つしかなければ、最も早い終了時刻  $f_i$  と、ある開始時刻  $s_i$  とを有するエンティティを要素プールの外へ転送し、そうでない場合は、最も早い開始時刻  $s_i$  を有するエンティティを要素プールの外へ転送し、且つ、終了時刻  $f_i$  より前に転送する請求項 5 に記載のスケジューラ装置。

20

## 【請求項 7】

$N$  個のエンティティ ( $N$  は 1 以上の整数であり、エンティティはユーザからのセル) がサーバ(14)からのサービスを要求する工程；

$N$  個の各エンティティがサーバ(14)からのサービスを要求する開始時刻  $s_i$  と、 $N$  個の各エンティティがサーバによるサービスを完了すべき終了時刻  $f_i$  とをサーバ(14)に連繋したメモリ内に記憶する工程；

及び或る仮想時刻にて、最も早い終了時刻  $f_i$  を有するエンティティが唯 1 つしかなければ、サーバ(14)は  $N$  個のエンティティのうち、最も早い終了時刻  $f_i$  を有するエンティティにサービスを提供し、そうでない場合、サーバ(14)は最も早い開始時刻  $s_i$  を有するエンティティにサービスを提供し、且つ、仮想時刻と同一またはそれより前の開始時刻  $s_i$  のエンティティが存在すれば、その終了時刻  $f_i$  より前にエンティティへサービスし、仮想時刻を 1 単位分増加し、サーバがエンティティへのサービスを提供しないときは、所定の時間後に現在の仮想時間を 1 単位分、増加する工程を具え、

30

サーバがエンティティへのサービスを提供する時期をスケジュールする方法。

## 【請求項 8】

サービスを提供する工程の後に、サーバからのサービスを要求している他のエンティティに関して、該提供する工程を仮想時刻を進めて繰返す工程が存在する、請求項 7 に記載の方法。

40

## 【発明の詳細な説明】

## 【0001】

## 【発明の属する技術分野】

本発明は、例えばコンピュータネットワークにおける、スケジューラ (scheduler) に関する。特に、本発明は、エンティティ (entity) が、その終了時刻前に、サーバ(server) からサービスを受取ることができるスケジューラに関する。

## 【0002】

## 【従来の技術】

利用可能なリソースを多種多様なユーザの間で共有しなければならない状況下では、過密

50

問題がリソースの利用に関して起こり得る。非常に簡単に言うと、異なるユーザが所定のリソースを同時に使用したいが、該リソースの能力により一度に1ユーザにしかサービスの提供をできない時が存在し得る。この過密問題を制御するため、スケジューラが一般に利用されて、ユーザがサーバへアクセスする時期を規制する。

【0003】

【発明が解決しようとする課題】

スケジューラは、ユーザにサーバへのアクセスを公平に提供し、その結果どのユーザも、そのニーズに見合った過度の遅れを経験しない様にすることが理想である。従来から、ユーザのニーズが非常に分散しているとき、スケジューラは、必ずしもユーザに対して公平ではなかった。多くの場合、比較的大きなニーズを有するユーザが待機したままであり、一方比較的小さいニーズを有するユーザが、サーバにより提供されている。また、その逆のこともある。

本発明は、より大きなニーズを有するユーザが、所定の時刻にサービスを要求しない時に、より小さいニーズを有するユーザが提供され得るように、ユーザのニーズに応じる方向を示す。

さらに、本発明は、サーバが常に1ユーザに提供していることがないように、サーバの比率制御(rate control)を提供する。

【0004】

【課題を解決する為の手段】

本発明は、N個のエンティティがサーバによりサービスされる時期を制御するためのスケジューラに関する。ここで、Nは1以上の整数である。スケジューラは、開始時刻メモリを具え、該メモリは、N個のエンティティの開始時刻 $s_i$ を記憶する。ここで、 $i$ は1以上N以下の整数であり、 $s_i$ は、 $i$ 番目のエンティティがサーバからのサービスを要求する時刻に対応する。また、スケジューラは、終了時刻メモリを具え、該メモリは、N個のエンティティの終了時刻 $f_i$ を記憶する。ここで、 $f_i$ は、 $i$ 番目のエンティティがサーバによりサービスされるべき時刻に対応する。また、スケジューラは、仮想時計(virtual clock)を具え、該時計は、開始時刻 $s_i$ 及び終了時刻 $f_i$ を特定できるように、仮想時刻の経過を刻む。さらに、スケジューラは、コントローラを具え、該コントローラは、以下のような順序でエンティティを選択して、サーバからサービスを受取る。すなわち、最も早い終了時刻 $f_i$ と、ある開始時刻 $s_i$ とを有するエンティティが最初にサービスされ、もし終了時刻が同じならば、次に最も早いまたは最も遅い開始時刻 $s_i$ 、若しくは、最も高いまたは最も低いIDまたはレート(rate)を有するエンティティが、各所定時刻でサービスされる、という順序である。もし比率制御が用いられると、以下のような別の基準が存在する。すなわち、もし仮想時刻と同一またはそれより前の開始時刻を有するエンティティが存在しなければ、サーバは、サービスをどのエンティティへも提供せず、仮想時計は、仮想時刻をその1単位分増加するという基準である。比率制御において、仮想時刻は、常に、サーバの行動に関係なく時間関数として、仮想時刻の数単位分、望ましくは仮想時刻の1単位分増加される。その時間関数は、単に、所定の時間長さ(period)であるのが望ましい。前述した順序の後半から明らかなように、もしエンティティの終了時刻が同じならば、どのエンティティがサーバへ提供されて、オペレートされるべきかを示すために、どのようなプロトコルが希望されようとも、それは、オペレータが自由に選択すべきことである。

【0005】

本発明は、サーバがサービスをエンティティへ提供する時期をスケジュールする方法に関する。該方法は、N個のエンティティによって、サーバからのサービスを要求する工程を具え、ここで、Nは1以上の整数である。次に、N個のエンティティがそれぞれサービスを要求し始める時期と、N個のエンティティがそれぞれサービスを受取完了すべき時期を記憶する工程がある。次に、もし、最も早い受取時刻を有するエンティティが唯1つしか無ければ、N個のエンティティのうち最も早い受取時刻と、ある開始時刻を有するエンティティへサービスを提供し、そうでなければ、最も早い開始時刻を有するエンティティが

10

20

30

40

50

、多分その受取時刻の前に、サービスを受取る工程がある。もし、比率制御が使用されれば、次のような別の基準が存在する。すなわち、もし、仮想時刻と同一またはそれより前の開始時刻を有するエンティティが存在しなければ、サーバは使用されず、仮想時刻は、時間の関数として、仮想時刻の数単位分増加されるという基準である。

【0006】

本発明は、それぞれが重み  $w$  を有する  $N$  個のエンティティが、サーバにより提供を受ける時期をスケジュールするスケジューラに関する。ここで、 $N$  は 1 以上の整数であり、 $w$  は実数である。スケジューラは、開始時刻メモリを具え、該メモリは、 $N$  個のエンティティがサービスを要求する開始時刻  $s_i$  を記憶する。ここで、 $i$  は 1 以上  $N$  以下の整数であり、 $s_i$  は、 $i$  番目のエンティティがサーバからのサービスを次に得ることのできる時刻である。また、スケジューラは、終了時刻メモリを具え、該メモリは、 $N$  個のエンティティがサービスされるべき終了時刻  $f_i$  を記憶する。ここで、 $f_i$  は、その時刻までに  $i$  番目のエンティティがサーバによりサービスされるべき時刻である。また、スケジューラは、時間の経過  $V_k$  を刻む仮想時計を具え、ここで、 $k$  は 0 以上の整数で、且つ

$$V_k = \text{MAX} \left( (V_{k-1} + P) + (1 * X) + \left( (1 / AW) * T \right), \right. \\ \left. (s_{\text{Smin}} * Y) + (f_{\text{Fmin}} - s_{\text{Smin}}) * Z \right)$$

ここで、 $(|P| + |T| + |X|) * (|Y| + |Z|) > 0$

であり、ここで、 $s_{\text{Smin}}$  は、エンティティがサーバからのサービスを要求する時のなかで最も早い開始時刻であり； $f_{\text{Fmin}}$  は、サーバからのサービスを待機しているエンティティのなかで最も早い終了時刻であり； $V_{k-1}$  は、サーバがエンティティに前回サービスを提供した仮想時刻であり； $AW$  = 現在サービスを求めている全てのエンティティにわたっての重み  $w_i$  の総和、ここで、 $w_i$  は  $i$  番目のエンティティの重みであり；且つ  $P$ 、 $X$ 、 $T$ 、 $Y$  及び  $Z$  は実数である。 $P = 1$ 、 $T = 0$ 、 $X = 1$ 、 $Y = 1$  及び  $Z = 0$  が望ましいが、その応用及びその目的に基づいて、 $T$  及び  $X$  の両方が 0 でない限り、 $T$  または  $X$  の一方は、本質的に任意の実数となることができ；且つ  $Y$  及び  $Z$  の両方が 0 でない限り、 $Y$  または  $Z$  の一方は、本質的に任意の実数となることができる。もし、比率制御が利用されれば、 $Y = Z = 0$  であり、式の最後の行は適用されない。

【0007】

また、スケジューラは、コントローラを具え、該コントローラは、どのエンティティがサーバによりオペレートされるべきかを以下のような順序で選択する。すなわち、最も早い終了時刻  $f_i$  と、ある開始時刻  $s_i$  とを有するエンティティが最初にサービスを受取り、次に、所定の基準に基づくエンティティという順序である。所定の基準とは、例えば最も早い開始時刻または ID ナンバーのような、本来、ユーザが適用を望む任意の形式のタイブレーキング手法 (tie-breaking procedure) であってもよい。もし比率制御が利用されれば、以下のような別の基準が存在する。すなわち、もし、仮想時刻と同一またはそれより前の開始時刻を有するエンティティが存在しなければ、サーバは、どんなエンティティへもサービスを提供しないで、仮想時計は、仮想時刻を、時間の関数として仮想時刻の 1 単位分増加するのが望ましく、且つそれは、所定の時間後であるのが望ましいという基準である。

【0008】

本発明は、サーバがサービスをエンティティへ提供する時期をスケジュールするための方法に関する。該方法は、最初のエンティティがサーバからのサービスを要求する時期を特定する工程を具える。次に、エンティティがサーバからのサービスを要求する時期の関数として、例えば最初のエンティティまたは 2 番目のエンティティのようなエンティティへ、サービスを提供する工程が存在する。もし、比率制御を利用すれば、該提供工程は仮想時刻の関数であり、所定の時間後に、仮想時刻をその 1 単位分増加する追加の工程が存在する。

【0009】

10

20

30

40

50

本発明は、サーバがサービスをエンティティへ提供する時期を制御するためのスケジューラに関する。該スケジューラは、メモリを具え、該メモリは、エンティティがサーバからのサービスを要求する時期の関数である時刻を記憶する。また、スケジューラは、仮想時計を具え、該時計は、エンティティがサーバからのサービスを要求する時期の関数として、仮想時刻の経過を刻む。また、スケジューラはコントローラを具え、該コントローラにより、エンティティは、エンティティがサーバからのサービスを要求する時期の関数として、サーバからのサービスを受け取る。もし、比率制御が利用されれば、仮想時計は、時間関数、望ましくは所定の時間長さの関数として、時刻の経過を刻む。

【 0 0 1 0 】

【発明の実施の形態】

図中の同じ参照数字は、幾つかの図を通じて、同じようなまたは一致した部分を示す。図面中、特に図 1 及び図 2 4 において、スケジューラ(10)は、N 個のエンティティ(12)がサーバ(14)によりオペレートされる時期を制御する。ここで、N は 1 以上の整数である。スケジューラ(10)は、例えば優先キュー (queue) (16) のような、開始時刻メモリを具え、該メモリは、N 個のエンティティの開始時刻  $s_i$  を記憶する。ここで、 $i$  は 1 以上 N 以下の整数であり、 $s_i$  は、 $i$  番目のエンティティがサーバ(14)からのサービスを要求し始める時刻に一致する。また、スケジューラ(10)は、例えば優先キュー(18)のような、終了時刻メモリを具え、該メモリは、N 個のエンティティ(12)の終了時刻  $f_i$  を記憶する。ここで、 $f_i$  は、 $i$  番目のエンティティがサーバ(14)によりオペレートされるべき時刻に一致する。また、スケジューラ(10)は、仮想時計(20)を具え、該時計(20)は、開始時刻  $s_i$  及び終了時刻  $f_i$  を特定できるような時刻の経過を刻む。さらに、スケジューラ(10)は、コントローラ(22)を具え、該コントローラ(22)は、以下のような順序で、エンティティにサーバ(14)からのサービスを提供する。すなわち、最も早い終了時刻  $f_i$  と、ある開始時刻  $s_i$  とを有するエンティティが最初にオペレートされ、次に、もし終了時刻が同じならば、最も早いまたは最も遅い開始時刻  $s_i$  を有するか、若しくは、最も高いまたは最も低い ID またはレート<sup>レ</sup>を有するエンティティが、各所定時刻でオペレートされる、という順序である。後者から明らかのように、もしエンティティの終了時刻が同じならば、どのエンティティが、サーバ(14)へ提供されてオペレートされるべきかを示すために求められる任意のプロトコルを、オペレータが選択する。もし比率制御が使用されれば、次のような別の基準が存在する。すなわち、もし、仮想時刻と同一またはそれより前の開始時刻を有するエンティティが存在しなければ、サーバは、どのエンティティに対してもサービスを提供せず、仮想時計は、望ましくは所定の時間後で、仮想時刻を時間の関数として仮想時刻の 1 単位分増加する、という基準である。サーバがサービスを提供しないときでさえ、仮想時計は仮想時刻をその 1 単位分増加することに注目すべきである。その結果、仮想時刻は常に仮想時間の 1 増加分ずつ進行する。このときの処理の流れをフローチャートにて示したものが、図 2 5 である。

【 0 0 1 1 】

スケジューラ(10)は、要素プール(24)を含み得るのが望ましく、該プール(24)は、N 個のエンティティ(12)と接続され、該エンティティ(12)から、サーバ(14)によりオペレートされるべきエンティティを受け取る。そのとき、エンティティの到着時刻  $s_i$  は、 $i$  番目のエンティティ(12)が要素プール(24)に到着する時刻に対応し、終了時刻  $f_i$  は、 $i$  番目のエンティティ(12)が、要素プール(24)から離れて、サーバ(14)によりオペレートされるべき時刻に対応する。

【 0 0 1 2 】

コントローラ(22)は、サーバ(14)によりオペレートされるべきエンティティを、要素プール(24)から転送または選択する。コントローラ(22)は、仮想時計(20)、要素プール(24)、開始時刻優先キュー(16)及び終了時刻優先キュー(18)へ接続されるのが望ましい。これは、仮想時計(20)の時刻を制御し、要素プール(24)にエンティティを記憶し、且つ、開始時刻優先キュー(16)及び終了時刻優先キューに到着時刻  $s_i$  及び終了時刻  $f_i$  をそれぞれ記憶するためである。コントローラ(22)は、エンティティすなわち要素プール(24)の要素を、

10

20

30

40

50

順次オペレートするのが望ましい。エンティティがサービスを受取るとき、すなわちエンティティが要素プール(24)を離れるときに、コントローラ(22)により、仮想時計(20)が現在の仮想時刻を進めるのが望ましい。もし、エンティティの最も早い開始時刻が仮想時刻よりも遅ければ、コントローラ(22)により、仮想時計(20)は、エンティティすなわち要素の最も早い開始時刻へ、仮想時刻を増加することができ、他の場合では、コントローラ(22)により、仮想時計(20)は、仮想時刻をその1単位分増加する。もし比率制御が利用されれば、コントローラにより、後者が、望ましくは、もしサービスが提供されなければ、所定の時間後に生じせしめるのみである。

【0013】

この要素プール(24)を有する実施形態において、コントローラ(22)は、もし、最も早い終了時刻  $f_i$  を有するエンティティが唯一つしか無ければ、最も早い終了時刻  $f_i$  と、ある開始時刻  $s_i$  とを有するエンティティを、要素プール(24)の外へ転送する。そうでない場合、最も早い開始時刻  $s_i$  を有するエンティティが、コントローラ(22)により要素プール(24)の外へ、且つ該終了時刻  $f_i$  の前に転送されることが望ましい。もし仮想時刻と同一またはそれより前の開始時刻  $s_i$  を有するエンティティが、要素プールに必ず存在すれば、転送される。そうでない場合、もし比率制御が利用されれば、どのエンティティもコントローラにより要素プール(24)の外へ転送されない。

【0014】

本発明は、それぞれが重み  $w$  を有する  $N$  個のエンティティが、サーバ(14)によってサービスを提供される時期をスケジュールするためのスケジューラ(10)に関する。ここで  $N$  は1以上の整数であり、 $w$  は実数である。スケジューラ(10)は、開始時刻メモリ(16)を具え、該メモリ(16)は、 $N$  個のエンティティ(12)がサービスを要求するときに、開始時刻  $s_i$  を記憶する。ここで、 $i$  は、1以上  $N$  以下の整数であり、 $s_i$  は、 $i$  番目のエンティティがサーバ(14)からのサービスを次に得てもよい時刻である。また、スケジューラ(10)は、終了時刻メモリ(18)を具え、該メモリ(18)は、 $N$  個のエンティティがサービスされるべきときに、終了時刻  $f_i$  を記憶する。ここで、 $f_i$  は、その時刻までには  $i$  番目のエンティティがサーバ(14)によりサービスされるべき時刻である。また、スケジューラ(10)は、時刻  $V_k$  の経過を刻む仮想時計(20)を具え、ここで、 $k$  は0以上の整数であり、

$$V_k = \text{MAX} \left( (V_{k-1} + P) + (1 * X) + \left( (1 / AW) * T \right), \right. \\ \left. (s_{Smin} * Y) + \left( (f_{Fmin} - s_{Smin}) * Z \right) \right) \quad (1)$$

$$\text{ここで、} (|P| + |T| + |X|) * (|Y| + |Z|) > 0$$

であり、ここで、 $s_{Smin}$  は、エンティティがサーバ(14)からサービスを要求する時期の最も早い開始時刻であり； $f_{Fmin}$  は、サーバ(14)からのサービスを待機しているエンティティの最も早い終了時刻であり； $V_{k-1}$  は、サーバ(14)がエンティティに前回提供した仮想時刻であり； $AW$  = 現在、サービスを求めている全てのエンティティにわたっての  $w_i$  の総和、ここで、 $w_i$  は  $i$  番目のエンティティの重みであり；且つ  $P$ 、 $X$ 、 $T$ 、 $Y$  及び  $Z$  は実数である。 $P = 1$ 、 $T = 0$ 、 $X = 1$ 、 $Y = 1$  及び  $Z = 0$  が望ましいが、その応用とその目的に基づいて、 $T$  及び  $X$  の両方が0でない限り、 $T$  または  $X$  の一方は、本質的に任意の実数であり得；且つ  $Y$  及び  $Z$  の両方が0でない限り、 $Y$  または  $Z$  の一方は、本質的に任意の実数であり得る。もし比率制御が利用されれば、 $Y = Z = 0$  であり、(1)式の最後の行は適用できない。すなわち、 $V_k = V_{k-1} + 1$  である。

【0015】

また、スケジューラ(10)は、コントローラを具え、該コントローラは、どのエンティティがサーバ(14)によりオペレートされるべきかを、以下のような順序で選択する。すなわち、最も早い終了時刻  $f_i$  と、ある開始時刻  $s_i$  とを有するエンティティと、次に所定の基準に基づくエンティティという順序である。所定の基準とは、例えば最も早い開始時刻またはIDナンバー等のような、本質的に、ユーザが適用を望む任意の方式のタイ・ブレーキング手法となり得る。もし比率制御が利用されれば、以下のような別の基準が存在する。

すなわち、もし、仮想時刻と同一またはそれより前の開始時刻を有するエンティティが無ければ、サーバは、どんなエンティティへのサービスも提供しないで、望ましくは、仮想時刻は、仮想時刻をその1単位分増加するという基準である。

【0016】

より具体的には、 $w_i$ は、 $i$ 番目のエンティティの"重み"であり、0より大きく1以下であるとする。 $w_i$ は、 $i$ 番目のエンティティが持っているサーバ(14)の分数割当て (fractional share) を表わす (もし  $w_i = 1$  ならば、サーバ(14)は、全て  $i$  番目のエンティティに捧げられ、もし  $w_i = 0.5$  ならば、サーバの作業の1/2が、 $i$  番目のエンティティに捧げられる)。

注意：この中で「1」は、サーバ(14)の作業の最大比率を表すために使用され、そのため、唯1つのエンティティに、サーバ(14)の能力の100%を超える重みを与えることは無意味である。

【0017】

もし、エンティティが、どれ位の作業の実行をサーバ(14)に求めることができるかに関して、制限がなければ、全エンティティの重みの総和は、1以下であることが望ましい。これは、サーバ(14)の能力の100%以下を割当てることに等しい。もし、エンティティが、サーバの能力のある割合 ( $RS_i$  とし、 $RS_i < 1$  である) 以下の量を求めるようにある程度制限されれば、より一般的な記述は、以下ようになる。すなわち、

$\text{MIN}(w_i, RS_i)$  の (全エンティティにわたっての) 和は1よりも小さくあるべきである。

$W =$  全エンティティにわたっての  $w_i$  の総和、すなわち

【数1】

$$W = \sum_{i=1}^N w_i$$

とする。 $AW =$  現在、サービスを求めている全エンティティにわたっての  $w_i$  の総和 (すなわち、もし「 $W$ 」がウェイト (Weight) の総和ならば、「 $AW$ 」はアクティブ・ウェイト (Active Weight) の総和である)、すなわち

【数2】

$$AW = \sum_{i=1}^N w_i \times a_i$$

とする。ここで、もしサービスを求めているなければ、 $a_i = 0$  であり、サービスを求めておれば、 $a_i = 1$  である。

【0018】

さらに、 $s_i$  は、 $i$  番目のエンティティが次にサービスを得る時刻であり、 $f_i$  は、その時刻までには  $i$  番目のエンティティがサービスされるべき時刻である。

【0019】

$u_i$  は、サーバが実際に  $i$  番目のエンティティのために働くときに、サーバが  $i$  番目のエンティティに与える作業の単位数とする (単純化するため、全てのエンティティに関して、 $u_i = 1$  である場合を仮定する (パケットネットワークの場合において、これは、全てのパケットが同じサイズを有する場合、例えば ATM セルの場合に等しい))。  $u_i$  は、応用する時の状況に応じて異なってもよい。例えば、 $u_i$  は、パケットに基づくネットワーク中にて、「1」と等しくなくても構わない。

【0020】

$F_{\text{min}} (1 < F_{\text{min}} < N) =$  最も小さい終了時刻を有するエンティティの ID 番号、とする (「最も小さい終了時刻を有するエンティティ」として与えられる全ての例は、「任意の必要とされるタイ・ブレーキング (tie breaking) に基づいて最初に終了すべきエンティティ」を意味すると解すること)。

10

20

30

40

50

その結果、 $f_{Fmin}$  = 最も小さい終了時刻を有するエンティティの終了時刻の値、である。

【0021】

$Smin(1 \leq Smin < N)$  = 最も小さい開始時刻を有するエンティティのID番号、とする  
(ここで意味するものは、上記と同じである。すなわち、もしタイがあれば、それら  
はある程度ブレイクされると仮定せよ)。

その結果、 $s_{Smin}$  = 最も小さい開始時刻を有するエンティティの開始時刻の値である。

【0022】

$V_k$  は、サーバが実際に作業をする  $k$  番目の時刻を示す。

$$V_k = \text{MAX} (V_{k-1} + (1 * X), (s_{Smin} * Y) + ((f_{Fmin} - s_{Smin}) * Z)) \quad (2) \quad 10$$

【0023】

サーバ(14)が作業を実行する度に、サーバ(14)は、そのID番号がFminであるエンティティをサービスする。サーバは、オペレートするためにSminの値を知る必要はなく、 $s_{Smin}$ の値を知るだけでよい。

【0024】

もう1つの式は、以下の式である：

$$V_k = \text{MAX} (V_{k-1} + ((1/AW) * T), (s_{Smin} * Y) + ((f_{Fmin} - s_{Smin}) * Z)) \quad (3) \quad 20$$

ここで、 $|T| * (|Y| + |Z|) > 0$

【0025】

(1)式は、(2)式及び(3)式を一緒にしたものである。

【0026】

( $P, T, X$ ) = (1, 0, 1) 及び ( $Y, Z$ ) = (0, 0) を有する(1)式は、「仮想時計 (Virtual Clock)」(VC)である。Zhang, L., 「パケット交換ネットワークプロトコルのための新しいアーキテクチャ」(1989年)、レポートLCS TR 455、コンピュータ科学研究所、マサチューセッツ工科大学、ケンブリッジ、マサチューセッツ州、を参照することにより、この明細書中に参考という形で組み入れられる。 30

【0027】

( $P, T, X$ ) = (0, 0, 0) 及び ( $Y, Z$ ) = (1, 1) を有する(1)式は、「自己計時フェア・キューイング (Self-Clocked Fair Queueing)」(SFQ)である。S. J. Golestani, 「A Self-Clocked Fair Queueing Scheme for Broadband Applications」、IN Proc. IEEE Infocom, May 1994、を参照することにより、この明細書中に参考という形で組み入れられる。

【0028】

この場合、式の後半部分が次のように見える点は注目すべきである。

$(s_{Smin} * 1) + ((f_{Fmin} - s_{Smin}) * 1)$ 、すなわち

$s_{Smin} + f_{Fmin} - s_{Smin}$ 、すなわち

$f_{Fmin}$

(そして、この場合、「 $s_{Smin}$ 」の値は、実際に必要とされない)

実際には、このように、概念的に文章にて記述される。

【0029】

( $P, T, X$ ) = (1, 1, 0) 及び ( $Y, Z$ ) = (0, 0) を有する(1)式は、「略」「重み付けフェア・キューイング (Weighted Fair Queueing)」(WFQ)である。A. Demers, S. Keshav, 及び S. Shenker, 「フェア・キューイング・アルゴリズムの解析及びシミュレーション」；インターネットワーキング：研究及び実験、第1巻、3-26 (1990)と、A. K. J. Parekh, 「A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks」、Ph.D.学位論文、マサチューセッツ工科大学、ケ 50

ンブリッジ、マサチューセッツ州、1992年2月とを参照することにより、この明細書中に参考という形で組入れられる。

【0030】

実施の観点からすると、この見解での実際問題は、以下の2点である。

a) この方式における「AW」の計算は、通常、サーバの離散モデルよりもむしろ流体モデルをエミュレートすることを必要とする

b)  $1/AW$ の計算は、非常に費用がかかり、実行が困難である(もし、 $AW = (1 - )$ ならば、 $1/AW = 1/(1 - )$ )であり、正確に計算することが難しい)

【0031】

(P, T, X) = (1, 0, 1) 及び (Y, Z) = (1, 0) を有する(1)式は、本発明の「仮想サービス・キューイング (Virtual Service Queueing) (VSQ)」である。

10

【0032】

これらの観察を行なう理由は、以下の2点である。

a) P, T, Xの組とY, Zの組とが、MAX関数の半分づつを表すことに注目し、VSQのみが、関数の両側で0の係数を有しないことが明らかである。

b) もし、SFQの方式における関数を、式を文中に通常書くように書換えれば、VSQのみが、値  $s_{Smin}$  を利用することもまた、明らかである。

【0033】

VSQの変形は、「比率制御サービス・キューイング (Rate Control Service Queueing) (RC VSQ)」である。比率制御が実行される時、 $Y = Z = 0$  である。しかしながら、エンティティが、サーバ(10)からのサービスを要求する時期の開始時刻と関連のないVCと異なり、RC VSQは、特定されるべきエンティティの開始時刻、具体的には、エンティティの終了時刻と同様に、最も小さい開始時刻を要求する。その結果、望ましい実施形態において、RC VSQは、例えば優先キュー(16)のような、開始時刻メモリと、例えば優先キュー(18)のような、終了時刻メモリとを、上述のように具える。次に、RC VSQのオペレーションにおいて、サーバ(10)は次の様にしてサービスを提供する：

20

【0034】

もし、 $Fmin$ の開始時刻 ( $s_{Fmin}$ ) が仮想時刻と同一またはそれより前であれば、 $Fmin$ にサービスし；その他もし、 $Smin$ の開始時刻 ( $s_{Smin}$ ) が、仮想時刻と同一またはそれより前であれば、 $Smin$ にサービスし；その他の場合は、何もしないで、仮想時刻を1単位分増加する。

30

【0035】

より簡潔には、次のように書かれる：

もし、 $s_{Fmin} < V$ 、ならば、 $Fmin$ にサービスし、その他もし、 $s_{Smin} < V$ 、ならば、 $Smin$ にサービスし、その他の場合は、何もしないで、 $V_k = V_{k-1} + 1$ 、とする。

【0036】

比率制御が実行される時、サーバ(10)は、それが何であろうとも、規定の比率でサービスを提供し、それより早くはない。比率制御は、必ずしも作業保存 (work conserving) である必要はなく、一方VSQは、作業保存である。

40

【0037】

本発明は、図22に示されるように、サーバ(10)がエンティティにオペレートする時期をスケジュールする方法に関するものである。該方法は、N個のエンティティからサーバのサービスを要求する工程を具え、ここで、Nは1以上の整数である。次に、N個の各エンティティが、最初にサービスを要求する時期と、サーバからのサービスを受取り完了すべき時期とを記憶する工程が存在する。次に、もし、最も早い受取時刻を有するエンティティが唯1つしかなければ、N個のエンティティのうち、最も早い受取時刻と、ある要求時刻とを有するエンティティへサービスを提供し、そうでなければ、最も早い要求時刻を有するエンティティがサーバからのサービスを受取る、という工程が存在する。該工程の後に、他のエンティティに関して、該工程を繰返す工程が存在することが望ましい。コン

50

トローラ(22)により、上記工程を実行することができる。

もし、比率制御が使用されれば、以下のような別の基準が存在する。すなわち、もし、仮想時刻と同一またはそれより前の要求時刻を有するエンティティが存在しなければ、サーバは何もしない、という基準である。

【0038】

また、本発明は、図23に示されるように、サーバ(14)がサービスをエンティティへ提供する時期をスケジュールするための方法に関する。該方法は、エンティティがサーバ(14)からのサービスを要求する時期を特定する工程を具える。最初のエンティティがサーバ(14)からのサービスを要求する時期を特定するためのプロセスは、最初のエンティティがサーバ(14)からのサービスを要求する時期を計算するための任意の技術及び方法を含み得る。これは、エンティティがサーバ(14)からのサービスを要求する時刻を、実際に決めることにより、該要求が起こる時期を特定することを含み得る。すなわち、それは、前のエンティティがサーバ(14)からのサービスを要求した時期から計算されることができ、エンティティがサーバ(14)からのサービスを要求する時期を、ある関数または関係式により、決定されることができる。すなわち、それは、現在、サーバ(14)からのサービスを受けている、別のエンティティの関数として決定され得る。その他に、最初のエンティティがサービスを要求する時刻は、該エンティティがサーバ(14)からのサービスを受取るべき時刻から特定され得る。例えば、開始時刻プラス1/W<sub>i</sub>が終了時刻に等しい。

10

【0039】

さらに、前記特定工程は、サーバがエンティティにサービスを提供すべき時刻を特定するプロセス全体におけるある時点で、エンティティがサーバからのサービスを要求する時刻が特定される、計算の一部であっても構わない。基本的には、特定工程は、プロセスに対する結論、または、プロセスに対する中間工程であることができる。該エンティティは、サーバ(14)からのサービスを要求する最初のエンティティであるのが望ましい。

20

【0040】

サーバ(14)が、次にエンティティへサービスを提供する時期をスケジュールする方法は、エンティティがサーバ(14)からのサービスを要求する時期の関数として、エンティティへのサービスを提供する工程を具える。該エンティティは、最初のエンティティであることができ、サーバ(14)からのサービスを、最初のエンティティがサーバ(14)からのサービスを要求する時期の関数として、受け取る。1番目のエンティティが、サーバ(14)からのサービスを初めて要求する時期が、1番目のエンティティへのサービスを提供するための基礎であるのが望ましい。他に、エンティティへのサービスを提供する工程は、最初のエンティティがサーバ(14)からのサービスを要求する時期の関数として、2番目のエンティティへのサービスを提供する工程であり得る。そのとき、2番目のエンティティがサービスを受取る時刻は、最初のエンティティがサーバ(14)からのサービスを最初に要求した時期の、ある関数関係式である。例えば、もし、 $V = 1$ で、最初のエンティティへのサービスを提供すれば、2番目のエンティティへのサービスは、 $V = 2$ すなわち $V = 1 + 1$ の時刻で提供される。ここで、 $V = 1$ は、最初のエンティティがサーバ(14)からのサービスを要求する時刻である。もし比率制御が利用されれば、前記提供工程は仮想時刻の関数であり、所定の時間後に、仮想時刻をその1単位分増加する追加工程が存在する。望ましい実施形態においては、仮想時刻は、他のどんな考慮にかかわらず、所定の時間後に、仮想時刻の1単位分増加される。

30

40

【0041】

本発明は、サーバがサービスをエンティティへ提供する時期を制御するためのスケジューラに関する。該スケジューラは、メモリを具え、該メモリは、エンティティがサーバ(14)からのサービスを要求する時期の関数である時刻を記憶する。また、スケジューラ(10)は、仮想時計(20)を具え、該時計(20)は、エンティティがサーバ(14)からのサービスを要求する時期の関数として、時間の経過を刻む。さらに、スケジューラ(10)はコントローラ(22)を具え、該コントローラ(22)により、エンティティは、エンティティがサーバ(14)からのサービスを要求する時期の関数として、サーバ(14)からのサービスを受け取る。該メモ

50

りは、例えば優先キュー(16)のような、開始時刻メモリであり得る。もし、比率制御が利用されれば、仮想時計は、エンティティへ提供されるサービスが存在しようがしまいが、所定の時間長さの関数として、時間の経過を刻む。

【0042】

【発明の実施形態の作用】

本発明の動作において、エンティティの主体である数人のユーザが、サーバ(14)からのサービスを受け取る。該ユーザは、個別の時刻または同じ時刻のどちらかで、サーバ(14)のオペレーションを要求するセルを有する。セルを有するどのユーザも、同時にサーバ(14)を呼出さないか、またはサーバ(14)を待っていないければ、サーバ(14)からのサービスを要求したユーザは、サーバからサービスを受け取る。しかしながら、もしそのような場合で

10

ないならば、スケジューラ(10)は、セルを有するユーザが、サーバ(14)によりサービスを受け取る時期をスケジュールするだろう。

【0043】

オペレーションの間、スケジューラ(10)は、ユーザがサーバ(14)からのサービスを最初に要求する時刻をログイン(log in)する。また、スケジューラ(10)は、ユーザがサーバ(14)からのサービスを待機し終わるべき最も遅い時刻をログインする。スケジューラ(10)は、サーバ(14)からのサービスを要求する各ユーザに関する開始時刻及び終了時刻を記録する。この情報は、ユーザと、ユーザがサーバ(14)からのサービスを最初に待機し始めた時期を特定する仮想時計(20)とから得られる。これは、ユーザが、サービスを要求するときに、サーバ(14)またはスケジューラ(10)へ信号を送る際に起こり得る；スケジューラ(10)またはサーバ(14)が、センサーまたはトリガを用いて、ユーザがサービスを要求している時期を特定する際に起こり得る。本発明のこの様相は、従来技術にてよく知られるように、遠隔通信のスイッチが、如何に動作して、幾つかの接続から該スイッチに到着する信号をスイッチするのか、と同じであり得る。

20

【0044】

ユーザにサービスするためにサーバ(14)が利用可能なとき、スケジューラ(10)は、サーバ(14)からのサービスを待機しているユーザのうち、どのユーザが最も早い終了時刻を有するかを決定する。次に、最も早い終了時刻を有するユーザは、サーバ(14)によりサービスが提供され、ユーザのセルは、サーバ(14)によりオペレートされる。もしユーザが同じ終了時刻を有するならば、サーバは、どのユーザが最も早い開始時刻を有するかを見て、最も早い開始時刻を有するユーザへサービスを提供する。スケジューラ(10)により、ユーザは、たとえサービスを待機し終わる時刻がまだ起こっていないくとも、サーバ(14)からのサービスを受取る。

30

【0045】

各ユーザがサーバ(14)によりサービスを提供されている間に、スケジューラ(10)は、サーバ(14)からのサービスを受取るべき次のユーザを探す。高いレートを有するユーザは、より低いレートを有するユーザよりも頻繁に、サーバ(14)からのサービスを受け取るだろう。しかしながら、比較的高いレートを有するユーザがサーバ(14)からのサービスを受取る間に、より高いレートを有するユーザよりも、早い終了時刻且つ/又は早い開始時刻を有している、より低いレートを有するユーザが、サーバ(14)からのサービスを受取る場合が存在するだろう。サーバ(14)からのサービスを受取っているユーザの終了時刻が起こったとき、スケジューラ(10)は該ユーザにより制御されないで、より低いレートを有するユーザは、サーバ(14)がより高いレートへのサービスを提供していないときに、サーバ(14)からのサービスを断続的に受取ることができる。その結果、サーバ(14)は、所定の単位サイクルのオペレーション中、ある一定のサービス能力を有するという事実と、様々なユーザの有するレート全てが加わると、サイクルに関する該単位レートに等しくなるという事実とにより制御されるから、1サイクルの進行中に、各ユーザは、サーバ(14)によりサービスを提供してもらおうであろう。

40

【0046】

簡単な例として、5つのユーザA-Eのレートが、それぞれ0.5、0.2、0.1、0.1、0.1で

50

あるとする。時刻  $V = 0$  で、各ユーザ A - E のセルは、サーバ(14)からサービスを受けるために到着する。ユーザ A からのセルは、終了時刻が  $V = 2$  であり、ユーザ B からのセルは、終了時刻が  $V = 5$  であり、且つユーザ C、D 及び E からのセルは終了時刻が  $V = 10$  である。これが図 2 に示される。

【 0 0 4 7 】

時刻  $V = 0$  で、ユーザ A からのセルが存在し、且つ、該セルは、サービスを待機している全てのセルの中で最も早い終了時刻  $V = 2$  を有するから、サーバ(14)はユーザ A にサービスする。サーバ(14)がユーザ A にサービスするから、時刻を  $V = 1$  へ 1 単位進める。

【 0 0 4 8 】

時刻  $V = 1$  で、サーバ(14)は、図 3 に示されるように、ユーザ B からのセルが  $V = 5$  という最も早い終了時刻を有するから、該セルに提供する。サーバ(14)がユーザ B のセルをオペレートするから、時刻を 1 単位分増加して、 $V = 2$  となる。

【 0 0 4 9 】

$V = 2$  で、図 4 に示されるように、ユーザ A からのセルが到着して、該セルの開始時刻が  $V = 2$  であり、且つ終了時刻が  $V = 4$  である。 $V = 2$  で、ユーザ A のセルに関する終了時刻  $V = 4$  は、最も早い終了時刻であるので、サーバ(14)は、ユーザ A からのセルにサービスする。サーバ(14)がユーザ A のセルをオペレートするとき、時刻が  $V = 3$  へ進められる。

【 0 0 5 0 】

時刻  $V = 3$  で、サーバ(14)は、ユーザ C からのセルをオペレートする。なぜなら、該セルは、図 5 に示されるように、サービスを待機しているどのセルにも匹敵する、早い終了時刻  $V = 10$  を有するからである。サーバ(14)がユーザ C からのセルにサービスを提供することにより、時刻が  $V = 4$  へ進められる。

【 0 0 5 1 】

時刻  $V = 4$  で、ユーザ A からのセルが、サーバ(14)のサービスを受けるために到着する。該セルは、時刻  $V = 4$  で到着し、図 6 に示されるように、終了時刻が  $V = 6$  である。ユーザ A からのセルが、最も早い終了時刻  $V = 6$  を有するから、サーバ(14)は該セルに提供する。それから、時刻が  $V = 5$  へ進められる。

【 0 0 5 2 】

時刻  $V = 5$  で、ユーザ B からのセルがサーバ(14)からのサービスを受取るために到着する。該セルの開始時刻は  $V = 5$  であり、終了時刻は  $V = 10$  である。時刻  $V = 5$  で、サーバ(14)は、ユーザ D からのセルへサービスを提供する。なぜなら、サービスを待機している全てのセルの終了時刻は同じであるが、ユーザ D からのセルの開始時刻は 0 であり、該セルは、ユーザ E からのセルとは同じレートであるが、ユーザ B からのセルよりも早いからである。図 7 を参照。サーバ(14)がユーザ D からのセルをサービスした後に、時刻が  $V = 6$  へ進められる。

【 0 0 5 3 】

時刻  $V = 6$  で、ユーザ A からのセルがサーバ(14)からサービスを受けるために到着する。該セルの開始時刻は  $V = 6$  であり、終了時刻は  $V = 8$  である。サーバ(14)は、ユーザ A からのセルが、サーバ(14)からのサービスを待機しているセルのうち最も早い終了時刻を有するから、該セルへサービスを提供する。図 8 を参照。サーバによりユーザ A からのセルがサービスされることにより、時刻は  $V = 7$  へ進められる。

【 0 0 5 4 】

時刻  $V = 7$  で、サーバ(14)は、ユーザ E からのセルへサービスを提供する。なぜなら、該セルは、最も早い開始時刻  $V = 0$  と、サーバ(14)からのサービスを待機しているユーザ B からのセルと同じ終了時刻とを有するからである。図 9 を参照。サーバ(14)がユーザ E からのセルへサービスを提供した後に、時刻は  $V = 8$  へ進められる。

【 0 0 5 5 】

時刻  $V = 8$  で、ユーザ A からのセルが、サーバ(14)からサービスを受けるために到着する。時刻  $V = 8$  で、サーバ(14)はユーザ B からのセルへサービスを提供する。なぜなら、該

10

20

30

40

50

セルは、最も早い開始時刻  $V = 5$  と、サーバ(14)からのサービスを待機しているユーザ A からのセルと同じ終了時刻を有するからである。図 10 を参照。サーバ(14)がユーザ B からのセルへサービスを提供した後に、時刻は  $V = 9$  へ進む。

【 0 0 5 6 】

時刻  $V = 9$  で、ユーザ A からのセルが、サーバ(14)からのサービスを待機している任意のセルのうち最も早い終了時刻を有するから、サーバ(14)は、該セルへサービスを提供する。図 11 を参照。この時点で、1 単位サイクルが完了し、全てのユーザがサービスを受け取った。サーバ(14)がユーザ A のセルへサービスを提供した後に、時刻は  $V = 10$  へ進む。

【 0 0 5 7 】

時刻  $V = 10$  で、ユーザ A からのセルは、サービスを受けるために到着して、終了時刻  $V = 12$  を有し、ユーザ B からのセルは、サービスを受けるために到着して、終了時刻  $V = 15$  を有し、且つ各ユーザ C、D 及び E からのセルは、到着して、終了時刻  $V = 20$  を有する（図示せず）。それから、上述のようなサイクルを繰り返す。

【 0 0 5 8 】

ユーザのレートを足しても 1 より小さいときのスケジューラの動作を説明する別の簡単な例が与えられる。この例では、3 つのユーザ A - C が、それぞれ 0.5、0.2 及び 0.1 のレートを有する。

【 0 0 5 9 】

時刻  $V = 0$  で、ユーザ A、B 及び C からのセルが、到着して、開始時刻  $V = 0$  を有する。ユーザ A からのセルは、終了時刻が  $V = 2$  であり、ユーザ B からのセルは、終了時刻が  $V = 5$  であり、且つユーザ C からのセルは、終了時刻が  $V = 10$  である。図 12 を参照。 $V = 0$  で、ユーザ A からのセルは、セルの中で最も早い終了時刻  $V = 2$  を有するから、サーバ(14)からのサービスを受け取る。サーバ(14)はユーザ A にサービスするから、時刻は  $V = 1$  へ 1 単位進められる。

【 0 0 6 0 】

$V = 1$  で、サーバ(14)は、ユーザ B からのセルが、セルの中で最も早い終了時刻  $V = 5$  を有するから、該セルにサービスする。図 13 を参照。それから、時刻が  $V = 2$  へ 1 増加分進められる。

【 0 0 6 1 】

時刻  $V = 2$  で、ユーザ A からのセルは、サーバ(14)からサービスを受けるために到着する。サーバ(14)は、ユーザ A からのセルが、セルの中で最も早い終了時刻  $V = 4$  を有するから、該セルにサービスする。図 14 を参照。それから、時刻が  $V = 3$  へ 1 増加分進められる。

【 0 0 6 2 】

時刻  $V = 3$  で、サーバは、ユーザ C からのセルへサービスを提供する。なぜなら、該セルは、現在の仮想時刻  $V = 3$  よりも早い開始時刻を有する唯一のセルであり、また、終了時刻  $V = 10$  を有するからである。図 15 を参照。

【 0 0 6 3 】

サーバ(14)がユーザ C からのセルにサービスする後に、仮想時刻は、サーバ(14)からのサービスを要求するセルのうち、最も早い開始時刻へ、1 増加分進められる。この場合では、サービスを待機しているセルのうち、最も早い開始時刻は、 $V = 4$  であり、それにより、 $V = 3$  から 1 時刻増加する。

【 0 0 6 4 】

時刻  $V = 4$  で、ユーザ A からのセルは、サーバ(14)からサービスを受けるために到着する。サーバ(14)は、ユーザ A からのセルへサービスを提供する。なぜならば、該セルは開始時刻  $V = 4$  を有し、これは現在の仮想時刻と等しく、且つ該セルは終了時刻  $V = 6$  を有し、これはサーバ(14)からのサービスを待機している任意のセルのうち、最も早い終了時刻であるからである。図 16 を参照。サーバ(14)がユーザ A からのセルへサービスを提供するとき、現在の仮想時刻  $V = 4$  と同じかまたはそれより早い開始時刻を有する他のセルが存在しないので、仮想時刻は、サーバ(14)からのサービスを要望するセルのうち、最も早

10

20

30

40

50

い開始時刻へ増加される。この場合では、ユーザBからのセルは、サーバ(14)からのサービスを要求するときに、開始時刻  $V = 5$  を有するであろうから、仮想時刻は、 $V = 5$  へ1単位分増加される。

【0065】

時刻  $V = 5$  で、ユーザBからのセルは、サーバ(14)からサービスを受けるために到着する。時刻  $V = 5$  で、ユーザBからのセルは、サーバ(14)からのサービスを待機している任意のセルのなかで、最も早い終了時刻を有するから、サーバ(14)は、該セルへサービスを提供する。図17を参照。サーバ(14)がユーザBからのセルへサービスを提供するとき、仮想時刻は、セルがサーバ(14)からのサービスを要求する最も早い開始時刻に対応する、次の仮想時刻へ増加される。この場合では、ユーザAからのセルの開始時刻が  $V = 6$  である。従って、時刻は  $V = 6$  へ1単位分増加される。

10

【0066】

時刻  $V = 6$  で、ユーザAからのセルは、サーバ(14)からサービスを受けるために到着する。時刻  $V = 6$  で、サーバ(14)は、ユーザAからのセルが最も早い終了時刻を有するから、該セルへサービスを提供する。図18を参照。それから、仮想時刻は、 $V = 7$  ではなく、 $V = 8$  へ増加される。なぜならば、サーバ(14)からのサービスを待機しているセルのうち、最も早い開始時刻は、 $V = 8$  で起こるであろうからである。その結果、仮想時刻は、この場合、2単位分増加される。もし、仮想時刻が時刻の1単位分ずつ増加される比率制御の実施形態が実施されれば、仮想時刻は、 $V = 8$  ではなく、 $V = 7$  へ増加されるであろうことに注意すべきである。 $V = 7$  で、サーバ(14)は、どんなユーザへもサービスを提供しないだろう。それから、所定の時間後に、仮想時刻は、 $V = 8$  へ1単位増加される。

20

【0067】

時刻  $V = 8$  で、ユーザAからのセルは、サーバ(14)からサービスを受けるために到着する。サーバ(14)は、ユーザAからのセルが最も早い終了時刻を有するから、該セルへサービスを提供する。図19を参照。サーバ(14)がユーザAからのセルへサービスを提供した後に、時刻は  $V = 10$  へ増加される。なぜならば、 $V = 10$  で、サーバ(14)からのサービスを待機しているセルに関して、最も早い開始時刻が存在するからである。そのとき、仮想時刻は、 $V = 8$  から  $V = 10$  へ2単位分増加される。

【0068】

$V = 10$  で、ユーザA、B及びCからのセルが、サーバ(14)からサービスを受けるために到着する。ユーザAからのセルに関する終了時刻は、 $V = 12$  であり、ユーザBからのセルに関する終了時刻は、 $V = 15$  であり、且つユーザCからのセルに関する終了時刻は、 $V = 20$  である(図示せず)。 $V = 10$  では、図が  $V = 10$  までしか描かれていないので、サービスを受けるために現れるセルは存在しない。しかしながら、 $V = 10$  に関連する終了時刻が、示されていないけれども、存在することから、 $V = 10$  は、図20において示される。

30

【0069】

【実施例】

本発明の幾つかの実施例が、図26乃至図44に示されている。図38乃至図44において、 $V S Q$  の前にある  $R C$  は、2つの実施例における比率制御を表すことに注意せよ。これらの実施例は、サーバ(14)によってオペレートされるために要素プール(24)に到着する要素を示し、該要素は、ATMネットワークに関するセルであるのが望ましい。該実施例をよりよく理解するために、スケジューラ(10)の動作の幾つかを次に説明する。

40

【0070】

実施例1(図26)において、3つのエンティティ(12)が、要素プール(24)へ接続される。重み及びレートは、これらのエンティティ(12)と関係がある。例えば、0番目のエンティティ(12)に関するユーザ0は、重みが0.5であり、レートが0.5である。1番目のエンティティ(12)に関するユーザ1は、重みが0.333であり、レートが0.333である。2番目のエンティティ(12)に関するユーザ2は、重みが0.166であり、レートが0.166である。時刻  $T = 1$  で、仮想時刻は0である。セル000、001及び002は、0番目、1番目及び2番目のユーザの0番目、1番目及び2番目のエンティティ(12)に対応し、要素プール(24)に到着す

50

る。該セルが、要素プール(24)にて最初に到着した開始時刻と、要素プール(24)を離れてしまうべき終了時刻とは、それぞれ、開始時刻優先キュー(16)と終了時刻優先キュー(18)にて記録される。これは、開始時刻列及び終了時刻列並びに開始キュー列及び終了キュー列にて示される。

**【0071】**

開始時刻列において、ユーザ0は、0番目のユーザに関連するセル000が要素プール(24)に到着した時に関する位置が、2番目すなわち最後である。該要素が要素プールに到着した時刻は、時刻 $T = 0$ である。終了時刻列において、ユーザ0の位置は、最初すなわち0番目である。というのは、ユーザ0の終了時刻 $T = 2000$ は、要素プール(24)における全ての要素の中で最も早い終了時刻だからである。ユーザ0に関して、該列において特定される量 $Q$ があり、該 $Q$ は、要素プール(24)において、ユーザ0のセルが唯一つであることを示している。開始キュー列において、ユーザ0は、最後すなわち2番目の位置にあり、開始時刻が $T = 0$ である。しかしながら、終了キュー列に基づく終了時刻優先キュー(18)において、ユーザ0は、0番目の位置にある。なぜならば、該時刻 $T = 2000$ は、要素プール(24)における全ての要素の中で最も早い終了時刻だからである。

10

**【0072】**

ユーザ1に関しては、開始時刻の位置が1であり、関連する開始時刻が $T = 0$ である。終了時刻の位置も1である。なぜなら、終了時刻 $T = 3000$ は、要素プール(24)における3つのセルの中で2番目に早い終了時刻だからである。ユーザ1に関して、要素プール(24)におけるセルの量 $Q$ が1であり、開始時刻優先キュー(16)において、ユーザ1は、位置が1番目で、該時刻が $T = 0$ である。ユーザ1は、終了キュー列における位置も1番目である。というのは、ユーザ1の終了すべき時刻は、要素プール(24)におけるセルのうち2番目に早いからである。

20

**【0073】**

ユーザ2は、開始時刻優先キュー(16)における位置が、最も早いすなわち0番目であり、時刻が $T = 0$ である。ユーザ2の終了時刻 $T = 6000$ は、要素プール(24)における全てのセルのなかで最も遅い終了時刻であるから、その終了時刻位置は、最後すなわち2番目である。ユーザ2は、要素プール(24)におけるセルの量 $Q$ が1であり、開始キューの位置が0番目であり、時刻が $T = 0$ である。終了時刻優先キュー(18)において、ユーザ2の終了キューの位置は2番目である。なぜならば、その終了時刻 $T = 6000$ は、要素プール(24)における3つのセルのうち最も遅い終了時刻だからである。

30

**【0074】**

時刻0000での動作において、ユーザ0のセルは、サーバ(14)によりオペレートされる。これを反映して、ユーザ0は、開始時刻の位置が-1で、時刻が $T = 0$ となり、終了時刻の位置が-1で、時刻が $T = 2000$ となる。ユーザ0の $Q$ 量は、その対応するセルが、サーバ(14)によりオペレートされて、もはや要素プール(24)に存在しないから、今や0である。開始時刻優先キュー(16)において、ユーザ0の位置が-1で、時刻が $T = -1$ であり、これは、ユーザ0が要素プールにもはや存在しないという事実を反映している。また、ユーザ0に関連するセルが要素プール(24)にもはや存在しないから、終了時刻優先キュー(18)により、ユーザ0の終了キューが要素プール(24)において-1であることが反映される。サーバは、同時に1つのセルしかオペレートできないから、最も早い終了時刻を有するユーザ0のセルのみが、要素プール(24)から離れる。残りのユーザは、終了時刻に関する位置が1段上昇する。というのは、今や、ユーザ1のセルに関する時刻 $T = 3000$ は、最も早い終了時刻であり、ユーザ2に関連するセルに関する時刻 $T = 6000$ は、今や、2番目に遅い終了時刻であるからである。また、本実施例から明らかのように、ユーザ0、1及び2のセル全てが、たとえそれらの重みが異なっても、同じ開始時刻で要素プール(24)に到着したから、該セルは、開始時刻優先キュー(16)において、最も大きいユーザナンバーを基準として最も小さいユーザナンバーまで順序付けられる。しかしながら、本質的には、任意のやり方の順序付け(scheme of ordering)で行なうことは自由である。

40

**【0075】**

50

時刻が  $T = 2$  で、仮想時刻が1000に等しい時(サーバ(14)が、たった今、セルをオペレートしたから、仮想時刻は、仮想時刻の1単位分 - - ここでは1000 - - 増加される)、ユーザ0、ユーザ1及びユーザ2からの新しいセルが要素プール(24)に到着する。ユーザ0の開始時刻は、開始時刻優先キュー(16)において到着を確認される時刻であるから、 $T = 200$ である。終了時刻優先キュー(18)におけるユーザ0の位置は1である。なぜなら、その終了時刻  $T = 4000$ は、終了時刻において、セルのなかで2番目に早いからである。さらに、ユーザ1とユーザ2の該量  $Q$ は、現在両方とも2である。というのは、これら各ユーザも、時刻  $T = 2$  で要素プール(24)内にセルが追加されるからである。ユーザ1は、要素プール(24)における全てのセルのうち最も早い終了時刻3000を有し、また、時刻  $V = 1000$ よりも早い開始時刻  $T = 0$ を有し、その結果、要素プール(24)に存在するから、要素プール(24)にあるこのセルが、サーバ(14)へ提供されて、サーバ(14)によりオペレートされる。その結果、サーバ(14)の次のオペレーションでは、サーバ(14)は、終了時刻が  $T = 3000$ であるユーザ1のセルに提供する。このセルは、要素プール(24)から移動され、その結果開始時刻優先キュー(16)と終了時刻優先キュー(18)が調整される。ユーザ1に関するセルの量  $Q$ は1に減らされ、一方ユーザ0に関するセルの量  $Q$ は1のままであり、且つユーザ2に関するセルの量  $Q$ は2のままである。ユーザ1に関して、要素プール(24)に、まだ残っているセルがあるから、その存在により、開始時刻列の時刻が  $T = 3000$ となり、終了時刻列の終了時刻が  $T = 6000$ となることが明らかになる。

**【0076】**

時刻  $T = 3$  では、 $V = 2000$ である。というのは、セルが該サーバ(14)によりオペレートされたことにより、仮想時刻  $V = 1000$ が1単位時刻分増加されるからである。時刻  $T = 3$  及び  $V = 2000$ では、唯1つの新しいセルが到着する。この新しいセルは、ユーザ0からである。これを反映して、要素プール(24)におけるセルの量  $Q$ がユーザ0に関して2になる。この時刻で、ユーザ0は、終了時刻優先キュー(18)に  $T = 4000$ のセルを有し、該時刻は、最も早い終了時刻である。また、このセルは、開始時刻が  $T = 2000$ であるから、時刻  $V = 2000$ での要素プール(24)に存在し、且つサーバ(14)によりオペレートされるべきセルである。 $T = 3$  且つ仮想時刻が2000に等しい時刻で、サーバがオペレートするとき、終了時刻  $T = 4000$ を有するユーザ0のセルは、要素プール(24)から移動されて、サーバ(14)によりオペレートされる。次に、要素プール(24)におけるユーザ0に関連するセルの量  $Q$ は、1つ分減少される。

**【0077】**

この時点で、全てのユーザのセルが有する全ての終了時刻は、 $T = 6000$ である。その結果、終了時刻に関して全て等しいから、どのセルがサーバによって次にオペレートされるかに関して、開始時刻が制御するようになる。このことは、時刻  $T = 4$  及び  $V = 3000$ で見られることができる。該時刻で、ユーザ2は、要素プール(24)に2つのセルを有し、該セルの1つが、最も早い開始時刻  $T = 0$ を有する。次に、ユーザ2のこのセルが、サーバ(14)によりオペレートされる。図26乃至図44における本実施例の残り部分は、上記の説明に基づいて理解されるべきである。

**【0078】**

本発明を応用した特有の場合において、ATMスイッチ要素は、図21に示されるように、1乃至N個の入力接続(102)と1乃至M個の出力接続(104)を有する。ATMセルは、スイッチ要素(100)により、入力接続(102)を介して受信される。スイッチ要素(100)は、該セルをオペレートして、出力接続(104)を介して送信する。セルは、特定の入力接続(102)上から入ってくる時はいつでも、従来よく知られるように、プール(106)に置かれる。そして、従来よく知られるように、スイッチ要素のスケジューラ(108)に、対応するプール(106)にセルが存在することを通知する。各入力接続(102)は、スイッチ要素(100)に進入するセルが記憶される専用のプール(106)を有することができること、または個々の入力接続(102)からのセルの全てが、スイッチ要素のサーバ(110)がオペレートする準備ができるまで、記憶される共通プール(106)が存在し得ることに注意すべきである。

**【0079】**

スイッチ要素(100)についてのスケジューラ(108)は、ATM状況を除いて上述のとおり動作し、スイッチ要素(100)のサーバ(110)からサービスを受けるために、セルから得られた競り値(a bid value)を決定したように、セルについてスケジューラが特定した終了時刻によって、その終了時刻が、決定される。この競り(bid)の使用は、従来よく知られている。例えば、クレイグ・パートリッジ(Craig Partridge)著「ギガビット・ネットワーク」アディソン・ウェスリー(Addison Wesley)1994年発行；ATM・フォーラム編集「ATM・ユーザ・ネットワーク・インターフェース仕様書、3.0版」プレントイス・ホール(Prentice Hall)1994年発行；レイフ・オー・オンヴュラル(Raif O. Onvural)著「非同期転送モード・ネットワーク：パフォーマンス上の問題点」アーテック・ハウス社(Artech House Inc.)、マサチューセッツ州ノーウッド(Norwood)、1994年発行；及びヒュイ・ツァン(Hui Zhang)とスリニヴァソフ・ケシャフ(Srinivasov Keshav)共著「レートに基づくサービス規律の比較(Comparison of Rate-Based Service Disciplines)」ACM・SIGCOMM '91の議事録、を参照。これら全てが参照として組込まれる。

10

【0080】

パケットに関して、全てのパケットは、従来知られるように、スタンプ(stamp)されることができる。さらに、従来及び上述から知られるように、キューの先頭でのパケットが上述に基づいてコンピュータ処理されることができ、該キューにおけるパケット全てがスタンプされる必要があるとは限らない。理想的には、最初のN個のパケットは、スタンプされることができ、その後のどんなパケットもスタンプされる必要がない。ここで、Nは、パケットがサービスを受取る際に遅れることができる最大のサービスインターバル数である。

20

【0081】

一般に、i番目のパケットに対して、

$$S_k^i = \text{MAX} (F_k^{i-1}, v(a_k^i)),$$

$$F_k^i = S_k^i + L_k^i / w_k^i$$

ここで、L = i番目のパケットの長さ、  
 $a_k^i$ は、i番目のパケットがサービスを受けるための到着時刻、  
 $v(a_k^i)$ は、到着時刻 $a_k^i$ での仮想時刻である。

30

【0082】

【発明の効果】

本発明の方法及び装置によって、より大きなニーズを有するユーザが、所定の時刻にサービスを要求しない時に、より小さいニーズを有するユーザにサービスを提供することができ、ユーザに対して、より公平なサービスの提供が実現できた。

さらに、本発明のサーバの比率制御によって、サービスを要求するユーザの中で、1ユーザに過度に集中してサービスを提供することがなく、サーバの利用効率が向上した。

【0083】

本発明は、例示の目的のために上記実施形態にて詳細に記載されたが、その細部は、単に前記例示目的のためであり、特許請求の範囲に記載されるようなもの以外にも、本発明の精神及び範囲から離れることなく、従来技術に熟練する人々により、変形を成し得ると、理解されるべきである。

40

【図面の簡単な説明】

【図1】本発明の概要図である。

【図2】仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。

【図3】仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。

【図4】仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。

【図5】仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。

【図6】仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。

【図7】仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。

50

【図 8】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 9】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 10】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 11】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 12】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 13】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 14】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 15】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 16】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 17】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 18】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 19】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 20】 仮想時刻の関数としてサーバからのサービスを待機するセルの概要図である。  
 【図 21】 ATM スイッチ要素の概要図である。

10

【図 22】 サーバがサービスをエンティティへ提供する時期をスケジュールする、本発明の方法のフローチャートである。

【図 23】 サーバがサービスをエンティティへ提供する時期をスケジュールする、他の本発明の方法のフローチャートである。

【図 24】 図 1 の各構成要素の繋がりを示すブロック図である。

【図 25】 比率制御を使用する本発明のスケジューラに関するフローチャートである。

20

【図 26】 本発明の実施例に関するセルの概要図である。

【図 27】 本発明の実施例に関するセルの概要図である。

【図 28】 本発明の実施例に関するセルの概要図である。

【図 29】 本発明の実施例に関するセルの概要図である。

【図 30】 本発明の実施例に関するセルの概要図である。

【図 31】 本発明の実施例に関するセルの概要図である。

【図 32】 本発明の実施例に関するセルの概要図である。

【図 33】 本発明の実施例に関するセルの概要図である。

【図 34】 本発明の実施例に関するセルの概要図である。

【図 35】 本発明の実施例に関するセルの概要図である。

30

【図 36】 本発明の実施例に関するセルの概要図である。

【図 37】 本発明の実施例に関するセルの概要図である。

【図 38】 本発明で、比率制御を使用する場合の実施例に関するセルの概要図である。

【図 39】 本発明で、比率制御を使用する場合の実施例に関するセルの概要図である。

【図 40】 本発明で、比率制御を使用する場合の実施例に関するセルの概要図である。

【図 41】 本発明で、比率制御を使用する場合の実施例に関するセルの概要図である。

【図 42】 本発明で、比率制御を使用する場合の実施例に関するセルの概要図である。

【図 43】 本発明で、比率制御を使用する場合の実施例に関するセルの概要図である。

【図 44】 本発明で、比率制御を使用する場合の実施例に関するセルの概要図である。

【符号の説明】

40

(10) スケジューラ

(12) エンティティ

(14) サーバ

(16) 開始時刻メモリ、開始時刻優先キュー

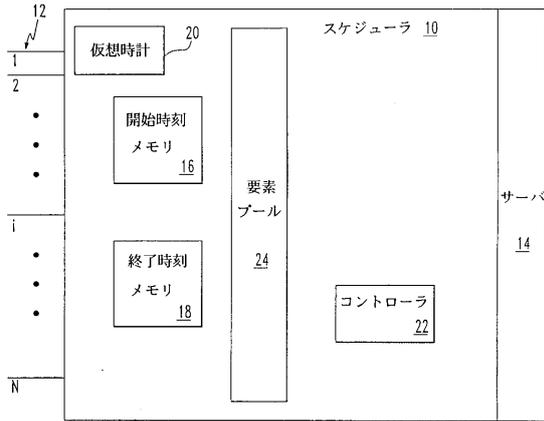
(18) 終了時刻メモリ、終了時刻優先キュー

(20) 仮想時計

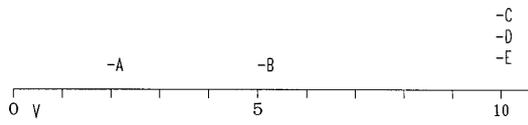
(22) コントローラ

(24) 要素プール

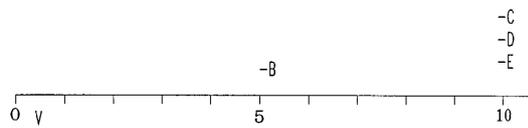
【 図 1 】



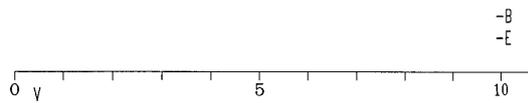
【 図 2 】



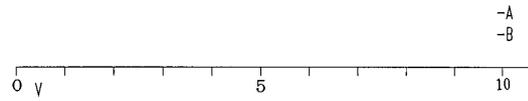
【 図 3 】



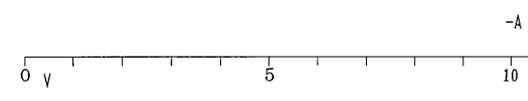
【 図 9 】



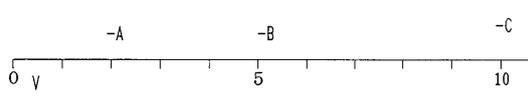
【 図 10 】



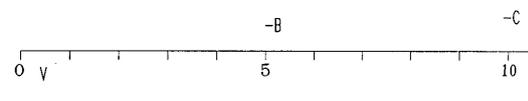
【 図 11 】



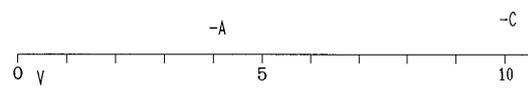
【 図 12 】



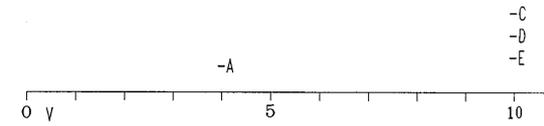
【 図 13 】



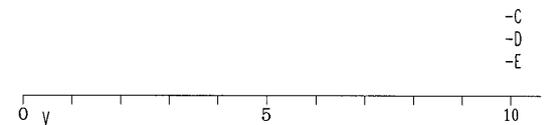
【 図 14 】



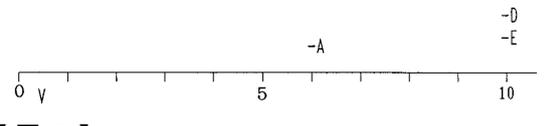
【 図 4 】



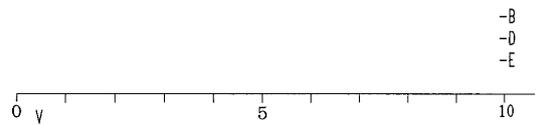
【 図 5 】



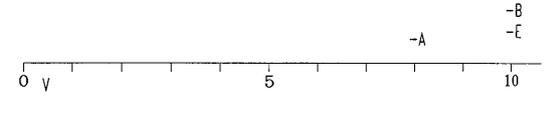
【 図 6 】



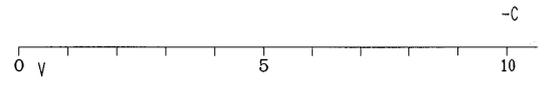
【 図 7 】



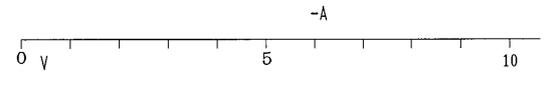
【 図 8 】



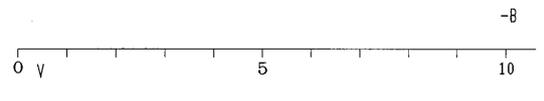
【 図 15 】



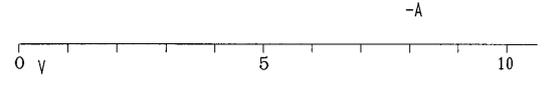
【 図 16 】



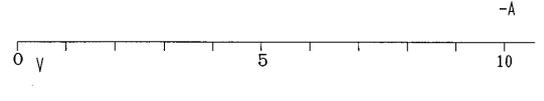
【 図 17 】



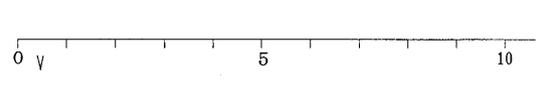
【 図 18 】



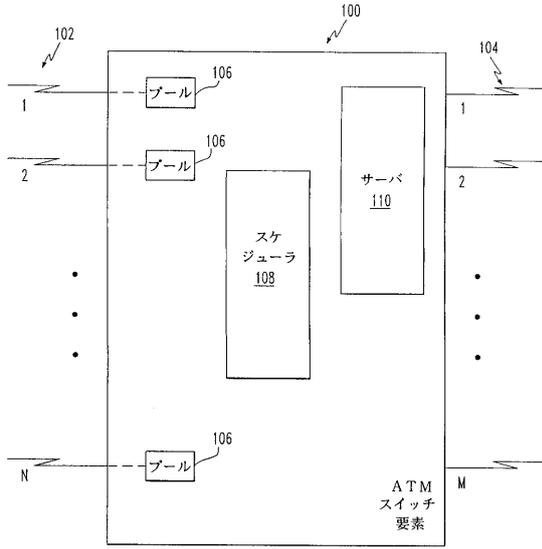
【 図 19 】



【 図 20 】



【図 2 1】



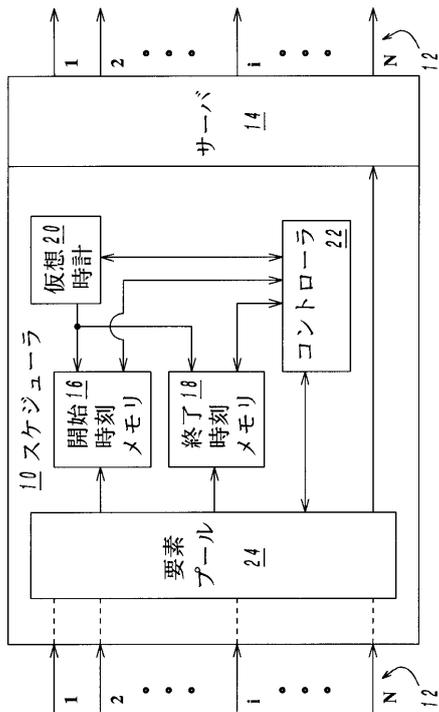
【図 2 2】



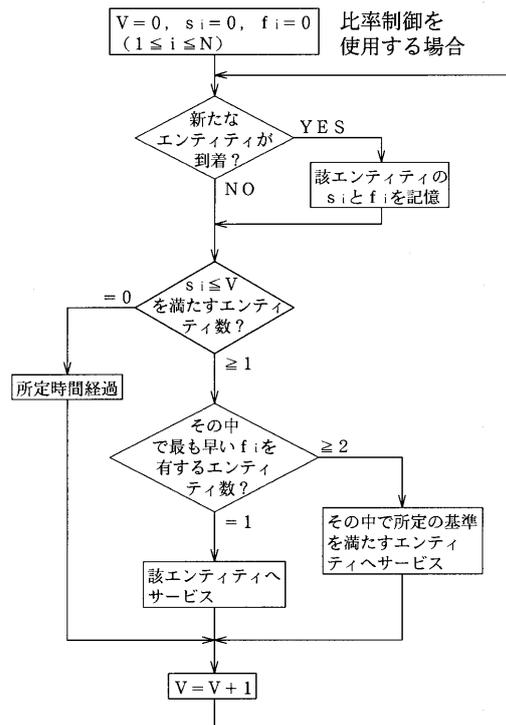
【図 2 3】



【図 2 4】



【図 2 5】



【 26】

VSQ\_27

1/6

.../vsq\_mlg = 0, time = 20, mark = 0, inc = 1
con # M R AM
0 0.500000 0.500000 0.500000
1 0.333333 0.333333 0.333333
2 0.166667 0.166667 0.166667

T= 1
V= 0
ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 2, 0) ( 0, 2000) 1 ( 2, 0) ( 0, 2000)
1 ( 1, 0) ( 1, 3000) 1 ( 1, 0) ( 1, 3000)
2 ( 0, 0) ( 2, 6000) 1 ( 0, 0) ( 2, 6000)

T= 2
V= 1000
ARRIVALS....
H000
H001
H002
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 1, 2000) ( 0, 4000) 1 ( 2, 0) ( 1, 3000)
1 ( 1, 0) ( 0, 3000) 2 ( 1, 0) ( 0, 4000)
2 ( 0, 0) ( 2, 6000) 2 ( 0, 2000) ( 2, 6000)

T= 3
V= 2000
ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 2, 4000) ( 2, 6000) 1 ( 2, 0) ( 2, 6000)
1 ( 1, 3000) ( 2, 6000) 1 ( 1, 3000) ( 1, 6000)
2 ( 0, 0) ( 0, 6000) 2 ( 0, 4000) ( 0, 6000)

【 27】

VSQ\_27

2/6

T= 4
V= 3000
ARRIVALS....
H001
SERVING H002
SERVING H000
SERVING H001

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 1, 4000) ( 1, 6000) 1 ( 1, 3000) ( 1, 6000)
1 ( 0, 3000) ( 0, 6000) 2 ( 0, 3000) ( 0, 6000)
2 ( 2, 0) ( 0, 6000) 2 ( 0, 6000) ( 2, 12000)

T= 5
V= 4000
ARRIVALS....
H000
H001
H002
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 0, 4000) ( 0, 6000) 2 ( 0, 4000) ( 0, 6000)
1 ( 2, 6000) ( 1, 9000) 1 ( 2, 6000) ( 1, 9000)
2 ( 1, 6000) ( 2, 12000) 1 ( 1, 6000) ( 2, 12000)

T= 6
V= 5000
ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 2, 6000) ( 0, 8000) 1 ( 2, 6000) ( 0, 8000)
1 ( 1, 6000) ( 0, 9000) 1 ( 1, 6000) ( 1, 9000)
2 ( 0, 6000) ( 2, 12000) 1 ( 0, 6000) ( 2, 12000)

T= 7
V= 6000
ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 2, 6000) ( 0, 8000) 2 ( 2, 6000) ( 0, 8000)
1 ( 1, 6000) ( 1, 9000) 1 ( 1, 6000) ( 1, 9000)
2 ( 0, 6000) ( 2, 12000) 2 ( 0, 6000) ( 2, 12000)

【 28】

VSQ\_27

3/6

ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 2, 8000) ( 1, 10000) 1 ( 2, 6000) ( 1, 9000)
1 ( 1, 6000) ( 0, 9000) 2 ( 1, 6000) ( 0, 10000)
2 ( 0, 6000) ( 2, 12000) 2 ( 0, 8000) ( 2, 12000)

T= 8
V= 7000
ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 1, 8000) ( 0, 10000) 1 ( 2, 6000) ( 0, 10000)
1 ( 2, 9000) ( 2, 12000) 1 ( 0, 8000) ( 2, 12000)
2 ( 0, 6000) ( 1, 12000) 2 ( 1, 9000) ( 1, 12000)

T= 9
V= 8000
ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 1, 8000) ( 0, 10000) 1 ( 2, 6000) ( 0, 10000)
1 ( 2, 9000) ( 2, 12000) 1 ( 0, 8000) ( 2, 12000)
2 ( 0, 6000) ( 1, 12000) 2 ( 1, 9000) ( 1, 12000)

T= 10
V= 9000
ARRIVALS....
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 2, 10000) ( 2, 12000) 1 ( 2, 6000) ( 2, 12000)
1 ( 1, 9000) ( 1, 12000) 2 ( 1, 9000) ( 1, 12000)
2 ( 0, 6000) ( 0, 12000) 2 ( 0, 10000) ( 0, 12000)

【 29】

VSQ\_27

4/6

ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 1, 10000) ( 1, 12000) 2 ( 1, 9000) ( 1, 12000)
1 ( 0, 9000) ( 0, 12000) 2 ( 0, 10000) ( 0, 12000)
2 ( 2, 12000) ( 2, 18000) 1 ( 2, 12000) ( 2, 18000)

T= 12
V= 11000
ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 0, 10000) ( 0, 12000) 2 ( 0, 10000) ( 0, 12000)
1 ( 2, 12000) ( 1, 15000) 1 ( 2, 12000) ( 1, 15000)
2 ( 1, 12000) ( 2, 18000) 1 ( 1, 12000) ( 2, 18000)

T= 13
V= 12000
ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 2, 12000) ( 0, 14000) 2 ( 2, 12000) ( 0, 14000)
1 ( 1, 12000) ( 1, 15000) 2 ( 1, 12000) ( 1, 15000)
2 ( 0, 12000) ( 2, 18000) 2 ( 0, 12000) ( 2, 18000)

T= 14
V= 13000
ARRIVALS....
H000
H001
H002
SERVING H000
SERVING H001
SERVING H002

Start Time Finish Time Start Queue Finish Queue
USER# (pos, T) (pos, T) (U, T) (U, T)
0 ( 2, 14000) ( 1, 16000) 1 ( 2, 12000) ( 1, 15000)
1 ( 1, 12000) ( 0, 15000) 2 ( 1, 12000) ( 0, 16000)
2 ( 0, 12000) ( 2, 18000) 2 ( 0, 14000) ( 2, 18000)

【 30】

VSQ\_27

5/6

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 1, 14000)	( 0, 18000)	1 ( 2, 12000)	( 0, 16000)
1 ( 2, 15000)	( 1, 18000)	1 ( 0, 14000)	( 2, 18000)
2 ( 0, 12000)	( 1, 18000)	2 ( 1, 15000)	( 1, 18000)

Ts = 15  
Vs = 14000  
ARRIVALS....  
U000

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 1, 14000)	( 0, 16000)	2 ( 2, 12000)	( 0, 16000)
1 ( 2, 15000)	( 1, 18000)	1 ( 0, 14000)	( 2, 18000)
2 ( 0, 12000)	( 0, 18000)	2 ( 1, 15000)	( 1, 18000)

SERVING U000

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 2, 16000)	( 1, 18000)	1 ( 2, 12000)	( 2, 18000)
1 ( 1, 15000)	( 1, 18000)	1 ( 2, 15000)	( 1, 18000)
2 ( 0, 12000)	( 0, 18000)	2 ( 0, 16000)	( 0, 18000)

Ts = 16  
Vs = 15000  
ARRIVALS....  
U001

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 2, 16000)	( 2, 18000)	1 ( 2, 12000)	( 2, 18000)
1 ( 1, 15000)	( 0, 18000)	2 ( 0, 16000)	( 0, 18000)
2 ( 0, 12000)	( 0, 18000)	2 ( 0, 16000)	( 0, 18000)

SERVING U002

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 1, 16000)	( 1, 18000)	2 ( 1, 15000)	( 1, 18000)
1 ( 0, 15000)	( 0, 18000)	2 ( 0, 16000)	( 0, 18000)
2 ( 2, 18000)	( 2, 24000)	1 ( 2, 18000)	( 2, 24000)

Ts = 17  
Vs = 16000  
ARRIVALS....  
U000

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 1, 16000)	( 1, 18000)	2 ( 1, 15000)	( 1, 18000)
1 ( 0, 15000)	( 0, 18000)	2 ( 0, 16000)	( 0, 18000)
2 ( 2, 18000)	( 2, 24000)	1 ( 2, 18000)	( 2, 24000)

SERVING U001

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 0, 16000)	( 0, 18000)	2 ( 0, 16000)	( 0, 18000)
1 ( 2, 18000)	( 1, 21000)	1 ( 2, 18000)	( 1, 21000)
2 ( 1, 18000)	( 2, 24000)	1 ( 1, 18000)	( 2, 24000)

Ts = 18  
Vs = 17000  
ARRIVALS....

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 0, 16000)	( 0, 18000)	2 ( 0, 16000)	( 0, 18000)
1 ( 2, 18000)	( 1, 21000)	1 ( 2, 18000)	( 1, 21000)
2 ( 1, 18000)	( 2, 24000)	1 ( 1, 18000)	( 2, 24000)

【 31】

VSQ\_27

6/6

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 0, 16000)	( 0, 18000)	2 ( 2, 18000)	( 0, 18000)
1 ( 2, 18000)	( 1, 21000)	1 ( 1, 18000)	( 1, 21000)
2 ( 1, 18000)	( 2, 24000)	1 ( 1, 18000)	( 2, 24000)

SERVING U000

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 2, 18000)	( 0, 20000)	1 ( 2, 18000)	( 0, 20000)
1 ( 1, 18000)	( 1, 21000)	1 ( 1, 18000)	( 1, 21000)
2 ( 0, 18000)	( 2, 24000)	1 ( 0, 18000)	( 2, 24000)

Ts = 19  
Vs = 18000  
ARRIVALS....  
U000

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 2, 18000)	( 0, 20000)	2 ( 2, 18000)	( 0, 20000)
1 ( 1, 18000)	( 1, 21000)	2 ( 1, 18000)	( 1, 21000)
2 ( 0, 18000)	( 2, 24000)	2 ( 0, 18000)	( 2, 24000)

SERVING U002

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 2, 18000)	( 2, 18000)	1 ( 2, 12000)	( 2, 18000)
1 ( 1, 18000)	( 1, 21000)	2 ( 0, 16000)	( 0, 22000)
2 ( 0, 18000)	( 2, 24000)	2 ( 0, 21000)	( 1, 21000)

SERVING U000

Start Time	Finish Time	Start Queue	Finish Queue
0 ( 2, 18000)	( 2, 18000)	1 ( 2, 12000)	( 2, 18000)
1 ( 1, 18000)	( 1, 21000)	2 ( 0, 16000)	( 0, 22000)
2 ( 0, 18000)	( 2, 24000)	2 ( 0, 21000)	( 1, 21000)

W = Weight, R = Rate  
Max E = Max Early, Max L = Max Late, (both in units of that users service time)  
con | W | R | should send | did send | Max E | Max L  
0 | 0.500000 | 0.500000 | 10 | 10 | 0.500000 | 0.000000  
1 | 0.333333 | 0.333333 | 6 | 6 | 0.333334 | 0.000000  
2 | 0.166667 | 0.166667 | 3 | 3 | 0.333333 | 0.000000  
AM = Achievable Rate, RM = Realized Rate, AB = Average Buffer Size  
con | W | R | RM | RM | AB  
0 | 0.500000 | 0.500000 | 0.500000 | 0.500000 | 1.000000  
1 | 0.333333 | 0.333333 | 0.333333 | 0.300000 | 1.150000  
2 | 0.166667 | 0.166667 | 0.166667 | 0.150000 | 1.300000

【 32】

VSQ\_26

1/6

.../vsq\_m\_elp = 0, time = 20, mark = 0, inc = 1  
con | W | R | AM  
0 | 0.500000 | 0.500000 | 0.500000  
1 | 0.333300 | 0.333300 | 0.333300  
2 | 0.166660 | 0.166660 | 0.166660

Ts = 1  
Vs = 0  
ARRIVALS....

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

SERVING U000

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

Ts = 2  
Vs = 0  
ARRIVALS....

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

SERVING U000

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

Ts = 3  
Vs = 0  
ARRIVALS....

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

SERVING U000

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

Ts = 4  
Vs = 3000  
ARRIVALS....

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

SERVING U001

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

【 33】

VSQ\_26

2/6

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

Ts = 5  
Vs = 3000  
ARRIVALS....

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

SERVING U000

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

Ts = 6  
Vs = 10000  
ARRIVALS....

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

SERVING U000

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

Ts = 7  
Vs = 10000  
ARRIVALS....

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

SERVING U001

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

Ts = 8  
Vs = 11000  
ARRIVALS....

Start Time	Finish Time	Start Queue	Finish Queue
0 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
1 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)
2 (-1, 0)	(-1, 0)	0 (-1, -1)	(-1, -1)

【 3 4 】

VSQ\_26 3/6

Start Time	Finish Time	Start Queue	Finish Queue
1 ( 1, 10000)	( 0, 13000)	1 ( 1, 10000)	( 2, 16000)
2 ( 0, 10000)	( 1, 16000)	1 ( -1, -1)	( -1, -1)

SERVING U001

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 10000)	( -1, 12000)	0	( 0, 10000)	( 0, 16000)
1	( -1, 10000)	( -1, 10000)	0	( -1, -1)	( -1, -1)
2	( 0, 10000)	( 0, 16000)	1	( -1, -1)	( -1, -1)

T= 9  
V= 12000  
ARRIVALS....  
U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 1, 12000)	( 0, 14000)	1	( 2, 10000)	( 0, 14000)
1	( -1, 10000)	( -1, 13000)	0	( 0, 12000)	( 2, 16000)
2	( 0, 10000)	( 1, 16000)	1	( -1, -1)	( -1, -1)

SERVING U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 12000)	( -1, 14000)	0	( 2, 10000)	( 2, 16000)
1	( -1, 10000)	( -1, 13000)	0	( -1, -1)	( -1, -1)
2	( 0, 10000)	( 0, 16000)	1	( -1, -1)	( -1, -1)

T= 10  
V= 13000  
ARRIVALS....  
U001

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 12000)	( -1, 14000)	0	( -2, 10000)	( 2, 16000)
1	( 1, 13000)	( 1, 16000)	1	( 1, 13000)	( 1, 16000)
2	( 0, 10000)	( 0, 16000)	1	( -1, -1)	( -1, -1)

SERVING U002

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 0, 13000)	( 0, 16000)	0	( -1, 13000)	( 1, 16000)
1	( 0, 13000)	( 0, 16000)	0	( -1, -1)	( -1, -1)
2	( -1, 10000)	( -1, 16000)	0	( -1, -1)	( -1, -1)

T= 11  
V= 14000  
ARRIVALS....  
U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 1, 14000)	( 1, 16000)	1	( 1, 13000)	( 1, 16000)
1	( 0, 13000)	( 0, 16000)	1	( 0, 14000)	( 0, 16000)
2	( -1, 10000)	( -1, 16000)	0	( -1, -1)	( -1, -1)

SERVING U001

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 0, 14000)	( 0, 16000)	0	( 0, 14000)	( 0, 16000)
1	( -1, 10000)	( -1, 16000)	0	( -1, -1)	( -1, -1)
2	( -1, 10000)	( -1, 16000)	0	( -1, -1)	( -1, -1)

【 3 5 】

VSQ\_26 4/6

T= 12  
V= 15000  
ARRIVALS....

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 0, 14000)	( -1, 16000)	0	( 0, 14000)	( 0, 16000)
1	( -1, 13000)	( -1, 16000)	0	( -1, -1)	( -1, -1)
2	( -1, 10000)	( -1, 16000)	0	( -1, -1)	( -1, -1)

SERVING U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 14000)	( -1, 16000)	0	( -1, -1)	( -1, -1)
1	( -1, 13000)	( -1, 16000)	0	( -1, -1)	( -1, -1)
2	( -1, 10000)	( -1, 16000)	0	( -1, -1)	( -1, -1)

T= 13  
V= 17000  
ARRIVALS....  
U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 2, 17000)	( 0, 19000)	1	( 2, 17000)	( 0, 19000)
1	( 1, 17000)	( 1, 20000)	1	( 1, 17000)	( 1, 20000)
2	( 0, 17000)	( 2, 23000)	1	( 0, 17000)	( 2, 23000)

SERVING U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 17000)	( -1, 19000)	0	( 2, 17000)	( 1, 20000)
1	( 1, 17000)	( 1, 20000)	1	( 1, 17000)	( 1, 20000)
2	( 0, 17000)	( 1, 23000)	1	( -1, -1)	( -1, -1)

T= 14  
V= 18000  
ARRIVALS....  
U001

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 2, 17000)	( 0, 19000)	1	( 2, 17000)	( 0, 19000)
1	( 1, 17000)	( 1, 20000)	1	( 1, 17000)	( 1, 20000)
2	( 0, 17000)	( 1, 23000)	1	( -1, -1)	( -1, -1)

SERVING U001

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 17000)	( -1, 19000)	0	( 2, 17000)	( 2, 23000)
1	( 1, 17000)	( 1, 20000)	1	( 1, 17000)	( 1, 20000)
2	( 0, 17000)	( 1, 23000)	1	( -1, -1)	( -1, -1)

T= 15  
V= 19000  
ARRIVALS....  
U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 1, 19000)	( 1, 21000)	1	( 2, 17000)	( 1, 21000)
1	( 1, 17000)	( 1, 20000)	0	( 0, 19000)	( 1, 23000)
2	( 0, 17000)	( 1, 23000)	1	( -1, -1)	( -1, -1)

【 3 6 】

VSQ\_26 5/6

SERVING U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 19000)	( -1, 21000)	0	( 2, 17000)	( 2, 23000)
1	( 0, 20000)	( 0, 23000)	1	( 1, 20000)	( 1, 23000)
2	( 0, 17000)	( 0, 23000)	1	( -1, -1)	( -1, -1)

T= 16  
V= 20000  
ARRIVALS....  
U001

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 19000)	( -1, 21000)	0	( 2, 17000)	( 2, 23000)
1	( 1, 20000)	( 1, 23000)	1	( 1, 20000)	( 1, 23000)
2	( 0, 17000)	( 0, 23000)	1	( -1, -1)	( -1, -1)

SERVING U002

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 19000)	( -1, 21000)	0	( 1, 20000)	( 1, 23000)
1	( 0, 20000)	( 0, 23000)	1	( -1, -1)	( -1, -1)
2	( -1, 17000)	( -1, 23000)	0	( -1, -1)	( -1, -1)

T= 17  
V= 21000  
ARRIVALS....  
U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 1, 21000)	( 1, 23000)	1	( 1, 20000)	( 1, 23000)
1	( 0, 20000)	( 0, 23000)	1	( 0, 21000)	( 0, 23000)
2	( -1, 17000)	( -1, 23000)	0	( -1, -1)	( -1, -1)

SERVING U001

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 0, 21000)	( 0, 23000)	1	( 0, 21000)	( 0, 23000)
1	( -1, 20000)	( -1, 23000)	0	( -1, -1)	( -1, -1)
2	( -1, 17000)	( -1, 23000)	0	( -1, -1)	( -1, -1)

T= 18  
V= 22000  
ARRIVALS....  
U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 0, 21000)	( 0, 23000)	0	( 0, 21000)	( 0, 23000)
1	( -1, 20000)	( -1, 23000)	0	( -1, -1)	( -1, -1)
2	( -1, 17000)	( -1, 23000)	0	( -1, -1)	( -1, -1)

SERVING U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 21000)	( -1, 23000)	0	( 0, 21000)	( 0, 23000)
1	( -1, 20000)	( -1, 23000)	0	( -1, -1)	( -1, -1)
2	( -1, 17000)	( -1, 23000)	0	( -1, -1)	( -1, -1)

T= 19  
V= 24000

【 3 7 】

VSQ\_26 6/6

ARRIVALS....  
U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( 2, 24000)	( 0, 26000)	1	( 2, 24000)	( 0, 26000)
1	( 1, 24000)	( 1, 27000)	1	( 1, 24000)	( 1, 27000)
2	( 0, 24000)	( 2, 30000)	1	( 0, 24000)	( 2, 30000)

SERVING U000

USER#	(pos, T)	(pos, T)	Q	(U, T)	(U, T)
0	( -1, 24000)	( -1, 26000)	0	( 2, 24000)	( 1, 27000)
1	( 1, 24000)	( 1, 27000)	1	( 1, 24000)	( 1, 27000)
2	( 0, 24000)	( 1, 30000)	1	( -1, -1)	( -1, -1)

W = Weight, R = Rate  
Max E = Max Early, Max L = Max Late, (both in units of that users service time)  
con # W R should send did send Max E Max L  
0 0.500000 0.500000 10 9 0.000000 1.000000  
1 0.333300 0.333300 6 5 0.000000 0.666600  
2 0.166600 0.166600 3 2 0.000000 0.666600  
AW = Achievative Rate, RR = Realized Rate, AB = Average Buffer Size  
con # W R RR AB  
0 0.500000 0.500000 0.500000 0.100000  
1 0.333300 0.333300 0.333300 0.250000 0.200000  
2 0.166600 0.166600 0.166600 0.100000 0.300000

【 38】

RCVSQ\_05

1/4

```

./././rcvsq) alg = 0. Lma = 10
con 1 W R AM
0 0.333300 0.400000 0.333300
1 0.250000 0.300000 0.250000
2 0.100000 0.100000 0.100000
3 0.100000 0.100000 0.100000
4 0.100000 0.100000 0.100000

```

T= 1

V= 0

ARRIVALS....

U000

U001

U002

U003

U004

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 4, 0)	( 0, 30003)	1	( 4, 0)	( 0, 30003)	
1	( 3, 0)	( 3, 40000)	1	( 3, 0)	( 3, 40000)	
2	( 2, 0)	( 4, 100000)	1	( 2, 0)	( 4, 100000)	
3	( 1, 0)	( 3, 100000)	1	( 1, 0)	( 3, 100000)	
4	( 0, 0)	( 2, 100000)	1	( 0, 0)	( 2, 100000)	

SERVING U000

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( -1, 0)	( -1, 30003)	0	( 4, 0)	( 1, 40000)	
1	( 3, 0)	( 0, 40000)	1	( 3, 0)	( 4, 100000)	
2	( 2, 0)	( 3, 100000)	1	( 2, 0)	( 3, 100000)	
3	( 1, 0)	( 2, 100000)	1	( 1, 0)	( 2, 100000)	
4	( 0, 0)	( 1, 100000)	1	( -1, -1)	( -1, -1)	

T= 2

V= 10000

ARRIVALS....

U000

U001

U002

U003

U004

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( -1, 0)	( 0, 30003)	0	( 4, 0)	( 1, 40000)	
1	( 3, 0)	( 0, 40000)	1	( 3, 0)	( 4, 100000)	
2	( 2, 0)	( 3, 100000)	1	( 2, 0)	( 3, 100000)	
3	( 1, 0)	( 2, 100000)	1	( 1, 0)	( 2, 100000)	
4	( 0, 0)	( 1, 100000)	1	( -1, -1)	( -1, -1)	

SERVING U001

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( -1, 0)	( -1, 30003)	0	( 4, 0)	( 4, 100000)	
1	( -1, 0)	( -1, 40000)	0	( 3, 0)	( 2, 100000)	
2	( 2, 0)	( 2, 100000)	1	( 2, 0)	( 2, 100000)	
3	( 1, 0)	( 1, 100000)	1	( -1, -1)	( -1, -1)	
4	( 0, 0)	( 0, 100000)	1	( -1, -1)	( -1, -1)	

T= 3

V= 20000

ARRIVALS....

U000

U001

U002

U003

U004

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 3, 30003)	( 0, 60006)	1	( 4, 0)	( 0, 60006)	
1	( -1, 0)	( -1, 40000)	0	( 3, 0)	( 4, 100000)	
2	( -1, 0)	( -1, 100000)	1	( 2, 0)	( 3, 100000)	

SERVING U001

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 1, 60006)	( 0, 90009)	1	( 2, 0)	( 0, 90009)	
1	( -1, 40000)	( -1, 80000)	0	( 0, 60006)	( 2, 100000)	
2	( -1, 0)	( 3, 100000)	1	( -1, -1)	( -1, -1)	
3	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

T= 7

V= 60000

ARRIVALS....

U001

U002

U003

U004

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 1, 60006)	( 0, 90009)	1	( 2, 0)	( 0, 90009)	
1	( 2, 80000)	( 2, 120000)	1	( 0, 60006)	( 2, 100000)	
2	( -1, 0)	( 3, 100000)	1	( -1, -1)	( -1, -1)	
3	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

SERVING U002

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 0, 60006)	( 0, 90009)	1	( 0, 60006)	( 0, 90009)	
1	( 1, 80000)	( 1, 120000)	1	( 1, 80000)	( 1, 120000)	
2	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
3	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

T= 8

V= 70000

ARRIVALS....

U001

U002

U003

U004

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 0, 60006)	( 0, 90009)	1	( 0, 60006)	( 0, 90009)	
1	( 1, 80000)	( 1, 120000)	1	( 1, 80000)	( 1, 120000)	
2	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
3	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

T= 9

V= 80000

ARRIVALS....

U001

U002

U003

U004

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 1, 90009)	( 1, 120012)	1	( 1, 80000)	( 1, 120000)	
1	( 0, 80000)	( 0, 120000)	1	( 0, 90009)	( 0, 120012)	

【 39】

RCVSQ\_05

2/4

```

3 ( 1, 0) ( 2, 100000) 1 ( 0, 30003) ( 2, 100000)
4 ( 0, 0) ( 1, 100000) 1 ( -1, -1) ( -1, -1)

```

SERVING U000

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 2, 30003)	( 0, 60006)	1	( 3, 0)	( 0, 60006)	
1	( -1, 0)	( -1, 40000)	0	( 2, 0)	( 1, 100000)	
2	( 1, 0)	( 2, 100000)	1	( 0, 30003)	( 2, 100000)	
3	( 0, 0)	( 1, 100000)	1	( -1, -1)	( -1, -1)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

T= 4

V= 30000

ARRIVALS....

U001

U002

U003

U004

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 2, 30003)	( 0, 60006)	1	( 3, 0)	( 0, 60006)	
1	( 3, 40000)	( 1, 80000)	1	( 2, 0)	( 1, 80000)	
2	( 1, 0)	( 3, 100000)	1	( 0, 30003)	( 3, 100000)	
3	( 0, 0)	( 2, 100000)	1	( 1, 40000)	( 2, 100000)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

SERVING U003

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 1, 30003)	( 0, 60006)	1	( 2, 0)	( 0, 60006)	
1	( 2, 40000)	( 1, 80000)	1	( 0, 30003)	( 1, 80000)	
2	( 0, 0)	( 2, 100000)	1	( 1, 40000)	( 2, 100000)	
3	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

T= 5

V= 40000

ARRIVALS....

U001

U002

U003

U004

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 1, 30003)	( 0, 60006)	1	( 2, 0)	( 0, 60006)	
1	( 2, 40000)	( 1, 80000)	1	( 0, 30003)	( 1, 80000)	
2	( 0, 0)	( 2, 100000)	1	( 1, 40000)	( 2, 100000)	
3	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

SERVING U000

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( -1, 30003)	( -1, 60006)	0	( 4, 0)	( 4, 100000)	
1	( 1, 40000)	( 0, 80000)	1	( 1, 40000)	( 2, 100000)	
2	( 0, 0)	( 1, 100000)	1	( -1, -1)	( -1, -1)	
3	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

T= 6

V= 50000

ARRIVALS....

U000

U001

U002

U003

U004

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 2, 100000)	( 0, 90009)	1	( 2, 0)	( 0, 90009)	
1	( 1, 40000)	( 0, 80000)	1	( 1, 40000)	( 0, 90009)	

【 40】

RCVSQ\_05

3/4

```

2 ( 0, 0) ( 2, 100000) 1 ( 0, 60006) ( 2, 100000)
3 ( -1, -1) ( -1, 100000) 0 ( -1, -1) ( -1, -1)
4 ( -1, 0) ( -1, 100000) 0 ( -1, -1) ( -1, -1)

```

SERVING U001

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 1, 60006)	( 0, 90009)	1	( 2, 0)	( 0, 90009)	
1	( -1, 40000)	( -1, 80000)	0	( 0, 60006)	( 2, 100000)	
2	( -1, 0)	( 3, 100000)	1	( -1, -1)	( -1, -1)	
3	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

T= 7

V= 60000

ARRIVALS....

U001

U002

U003

U004

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 1, 60006)	( 0, 90009)	1	( 2, 0)	( 0, 90009)	
1	( 2, 80000)	( 2, 120000)	1	( 0, 60006)	( 2, 100000)	
2	( -1, 0)	( 3, 100000)	1	( -1, -1)	( -1, -1)	
3	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	
4	( -1, 0)	( -1, 100000)	0	( -1, -1)	( -1, -1)	

SERVING U002

USER#	(pos. T)	(pos. T)	Q	(U. T)	(U. T)	Finish Queue
0	( 0, 60006)	( 0, 90009)	1	( 0, 60006)	( 0, 90009)	
1	( 1, 80000)	( 1, 120000)	1	( 1, 80000)	( 1, 120000)	
2	( -1, 0)	( -1				



---

フロントページの続き

(72)発明者 ジョン シー . アール . ベネット  
アメリカ合衆国 1 5 2 1 7 ペンシルベニア , ピッツバーグ , ドーゼル プレイス 6 5 2 5 ,  
# 2

合議体

審判長 江嶋 清仁

審判官 圓道 浩史

審判官 清水 稔

(56)参考文献 特開平 6 - 2 8 2 4 4 7 ( J P , A )

(58)調査した分野(Int.Cl. , D B 名)

G06F13/00

H04L12/56