

ジョブ配送システム JAM/JC の構成とプロトタイプ\*

4M-8

谷内 典行†  
静岡大学工学研究科‡

太田 剛 渡辺 尚 水野 忠則‡  
静岡大学情報学部‡

1 はじめに

近年、高速なネットワークの普及とともに、計算機をネットワークで接続して利用する形態が主流になってきている。そのため、ユーザはネットワーク上の計算機資源を自由に利用することができるようになった。しかし現実には、計算機システム全体の構成や状態、あるいは実行しようとしているジョブに関する知識が不足しているため、計算機システムが本来持っている能力を十分に利用することができていないことがほとんどである。そこで本研究では、ジョブの特性や計算機の特性を考慮して自動的にジョブに適した計算機を選び出し、そこへジョブを配送するシステム JAM/JC (Job Allocation Method based on Job Characteristics) を構築することを目的とする。

本システムは、分散システムの本来の性能をユーザがより簡単に利用できるようにすることを目的としている。ホストの特性とジョブの特性の両方を考慮して配送先を決定する点において、Utopia[1] や Task Broker[2] などの既存の分散システムと異なっている。また、リソース使用量などのような一般にジョブの実行前には分からない特性についても、過去の実行履歴より予測して利用する予定であるが、本稿ではジョブ特性の予測は扱わない。

2 JAM/JC の概観

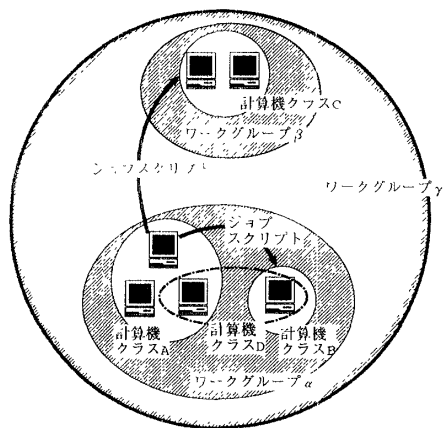


図 1: JAM/JC の概観

ジョブ配送システム JAM/JC の全体の概観を図 1 に示す。JAM/JC は、実行したいジョブの特性とシステム内の各ホストの特性を考慮し、ジョブの実行により適していると予測されるホストをユーザに代わって選択し、そこへジョブを自動的に配送することを目的としたシステムである。ここでいうジョブの特性/ホストの特性とは、例えば表 1 で示すようなものが含まれる。ジョ

ジョブ特性	<ul style="list-style-type: none"> <li>・ジョブが必要とする CPU 量, I/O 量, メモリ量など,</li> <li>・どの計算機で実行可能か,</li> <li>・端末制御を必要とする,</li> <li>・XWindow のクライアントである,</li> </ul>
ホスト特性	<ul style="list-style-type: none"> <li>・CPU 能力, I/O 能力, メモリ量</li> <li>・接続されている周辺機器,</li> <li>・現在実行中のジョブとそれらが使用しているリソース量,</li> </ul>

表 1: ジョブ特性とホスト特性

ブの配送先決定に必要なこれらの情報を、計算機システム全体で効率良く利用するために、計算機クラス、ジョブスクリプト、ワークグループの 3 つの概念を用いる。以下に、それぞれについて簡単に説明する。

計算機クラス

計算機クラスとは、ジョブを実行する能力が同じものごとに分類したホストの集合である。ジョブを実行する能力があるということは、ジョブが必要とする計算機資源を持っていることや、そのジョブを処理するために十分な能力を持っていることを意味する。分散システム上のホストを計算機クラスで分類することにより、実行したいジョブが分散システム上のどのホストで実行可能であるのかを容易に知ることができるようになる。各計算機クラスは、(a)CPU や OS の種類のような互換性、(b)CPU やファイル I/O の処理速度、メモリ量などの計算機資源、(c)サーバなどの計算機の役割、という 3 つの基準で分類される。ネットワーク上の計算機は、すべて 1 つ以上の計算機クラスに属する。

ジョブスクリプト

ジョブスクリプトは、ジョブ特性に関する情報を記述したもので、ジョブを配送する際に実際に送られるものである。表 1 に示したようなジョブの特性のほか、配送や実行に必要なさまざまな情報もこのスクリプトに記述される。例えば、ユーザ ID や実行ディレクトリ、ユーザが実際に利用しているホスト名などがそれに含まれる。必要とする CPU や I/O 能力、メモリ量など、実際にジョブを実行するまでは分からない情報については、過去の履歴情報から得られる予測値が用いられる。

ワークグループ

ワークグループとは、大学の研究室や会社の開発チームのような、共有される頻度が高いホストをグループ化したものであり、1 つ以上の計算機クラスから成る。一般に、分散システムの規模が大きくなると、各ホストの負荷管理やジョブの配送に伴う通信コストが増加する。しかし、常に分散システム上の最適なホストを選択するのではなく、同じワークグループ内のホストを優先して順最適なホストを選択することにより、このオーバーヘッドを減少させることができる。ワークグループは階層的に構成されており、ワークグループ内で実行できないジョブやより多くのホストから配送先を探したいジョブに関しては、より上位層のワークグループ内で配送先を探す。このワークグループを使用することにより、数十台〜

\*The Architecture of Job Allocation Method:JAM/JC and its Implementaion  
†Noriyuki Taniuchi, Graduate School of Enginnering, Shizuoka University  
‡Tsuyoshi Ohta, Takashi Watanabe and Tadanori Mizuno, Faculty of Information, Shizuoka University

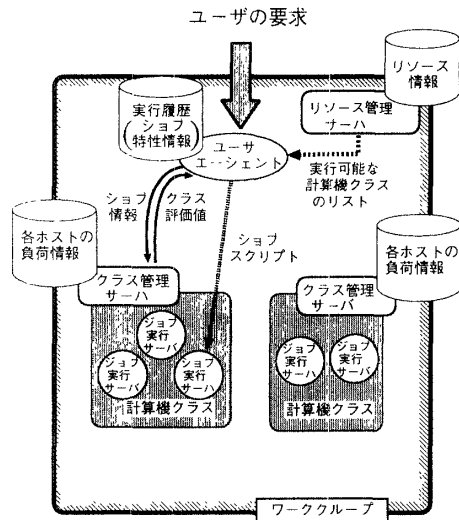


図 2: ワークグループの構成

数百台からなる大規模な分散システムでの配送先決定に必要な、システム全体の通信量を制限することができると考えている [3]。このワークグループと同様なクラス分けは、Utopia[1]でも計算機クラススタという名前でなされている。

Utopia の計算機クラススタは階層構造を持っていないが、通信量の抑制という目的は同じである。

### 3 JAM/JC の構成

図 2 にワークグループ内の構成を示し、以下にその構成要素と動作概要について簡単に説明する。

#### 3.1 構成要素

##### ユーザエージェント

ユーザエージェントは、ジョブの配送機構とユーザとの間のインターフェースであり、各ユーザごとに存在する。ユーザから要求を受け取ると、ユーザエージェントは自分が持つデータベースから得られる情報を付け加え、ジョブスクリプトを作成する。このデータベースには、事前に分かっているジョブ特性や、過去の実行時に計測したジョブのリソース使用量の統計、問い合わせるべきクラスのリストなどが記録されている。

##### クラス管理サーバ

クラス管理サーバは、各計算機クラスに 1 つずつ存在しており、それぞれの計算機クラスに属しているホストの現在の負荷情報を集中管理する。そして、各ユーザエージェントからの問い合わせに対して、計算機クラス内でそのジョブを実行するのにもっとも適していると考えられるホストを選択し、そのホスト名と評価値を返す。

##### リソース管理サーバ

リソース管理サーバは、各ワークグループに 1 つ存在する。ワークグループ内のどの計算機クラスでジョブを実行できるのかという情報を管理し、ユーザエージェントがジョブスクリプトを作成する際の問い合わせに答える役割をもつ。通常ユーザエージェントは、どの計算機クラスでジョブを実行できるのかという情報を既にデータベースに記録している。しかし、過去に実行したことがないジョブの配送先を決定する場合や、より上位の階層のワークグループで

配送先を探したい場合、あるいはユーザエージェントが所属するワークグループ内で実行できないジョブである場合に、このような問い合わせが生じる。

##### ジョブ実行サーバ

ジョブ実行サーバは、各ホストに 1 つ存在する。ユーザエージェントからジョブスクリプトを受け取り、実際にジョブを実行した後、ユーザエージェントに結果を返す。また、実行した際にジョブが使用したリソース量もユーザエージェントに報告する。

### 3.2 システム動作の概要

JAM/JC は、以下のような手順で動作する。

- I ユーザからの要求を受け取り、ユーザエージェントがジョブスクリプトを作成する。ユーザエージェントは、データベースからジョブの特性情報や、ジョブを実行可能なクラスのリストをジョブスクリプト中に書き込む。データベースに無いジョブや、上位層のワークグループで配送先を探すジョブに関しては、リソース管理サーバに問い合わせ、この情報を得る。
- II ユーザエージェントは、実行可能な各計算機クラスのクラス管理サーバに対してジョブの特性情報を送る。情報を受け取ったクラス管理サーバは、管理している計算機クラス内で、そのジョブを処理するのにもっとも適していると予想されるホストのアドレスとその評価値を返却する。
- III ユーザエージェントは、各クラスのクラス評価値を比較し、もっとも良い値をつけたホストのジョブ実行サーバにスクリプトを配送する。
- IV スクリプトを受け取ったジョブ実行サーバは、ジョブスクリプトの記述をもとにジョブを実行するとともに、自分自身の負荷を定期的にクラス管理サーバに報告する。ジョブの実行が終了すると、ジョブの処理結果をユーザエージェントに返却し、実行時に計測したジョブのリソース使用量も、ユーザエージェントに報告される。

### 4 まとめ

以上で、簡単にジョブ配送システム JAM/JC について説明した。現在このプロトタイプを実装中である。本研究では、実際に動くものを作成し、このシステムをユーザに評価してもらうことを目的としている。今後、その結果をもとに本システムの問題点を検討し、改良を加えていく予定である。

### 参考文献

- [1] S. Zhou, X. Zheng, J. Wang and P. Delisle, "Utopia: A Load Sharing Facility for Large, Heterogeneous Distributed Computer Systems," *Software-Practice and Experience*, Vol.23(12), pp.1305-1336, (1993).
- [2] T. P. Graf, R. G. Assini, J. M. Lewis, E. J. Sharpe, J. J. Turner, M. C. Ward, "Task Broker で、ネットワークのコンピュータ資源を有効活用する", *日経 エレクトロニクス*, 12.20 号, pp.143-154 (1993).
- [3] 谷内, 太田, 渡辺, 水野, "分散システムにおけるジョブ特性を考慮したジョブの配送先決定法の性能評価", *情報処理学会研究報告*, Vol.96, No.63, pp.43-48 (1996).