7H-7

階層型スターコンピュータネットワークにおける最適負荷分散*

町田 浩之　　亀田 壽夫
電気通信大学 情報工学科

1 はじめに

本研究では、スター型のコンピュータネットワークが幾層にも重なった階層構造を持つ図1のようなネットワークでの、静的な最適負荷分散を考える。最適負荷分散は、システム全体でのジョブの応答時間を最小化する方式である。スター型ネットワークにおける最適負荷分散については、単一ジョブクラスに関しTantawiら[1]によって、複数クラスに関しては、亀田とZhang[3]によって研究されている。この研究でのモデルは、これらのモデルを形状的に多重階層へと拡張、一般化したものである。

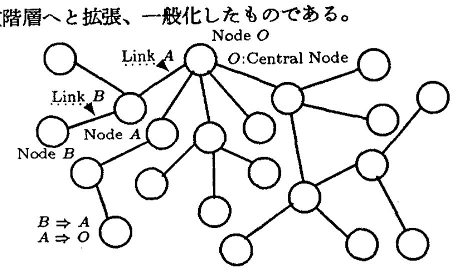

図1 階層型スターネットワークの構成例

2 モデル

本研究で対象とするモデルは、スター型を幾層にも積みあげて構成された全体でツリー構造をなすようなネットワークである。**各ノードに到着したジョブは、親のノードに転送されるか、自ノードで処理される。その逆は行われないとする。** ノードAがその親にのばしているリンクのLink名を、Aと定義する。Node名の集合をN_dと、Link名の集合をL_kと表す。(L_kは、N_dより中央ノードを除いたものとなる。)
上の集合上に次のような関係を定める。

- $B \Rightarrow A$:　ノードAがBの親である。
- $B \stackrel{+}{\Rightarrow} A$: (即ち $B \Rightarrow A_1 \Rightarrow A_2 \Rightarrow \cdots \Rightarrow A_u \Rightarrow A,\ A_i \in N_d(i>0)$) ノードAがBの祖先である。
- $B \stackrel{*}{\Rightarrow} A$:　$B \stackrel{+}{\Rightarrow} A$または$B = A$が成り立つ。

記号を以下のように定義する。

- $\phi_A^{(k)}$:　ノードAへのクラスkのジョブ到着率 (ポアソン到着と仮定)
- Φ :　全外部到着の総和
- m :　総ジョブクラス数
- $\beta_A^{(k)}$:　ノードAでのクラスkのジョブ処理率
- $\boldsymbol{\beta}_A = (\beta_A^{(1)}\beta_A^{(2)},\ldots,\beta_A^{(m)})$:　ノードAでのロードベクタ
- $\lambda_A^{(k)}$:　通信リンクAにおけるクラスkジョブ通信量
- $\boldsymbol{\lambda}_A = (\lambda_A^{(1)},\lambda_A^{(2)},\ldots,\lambda_A^{(m)})$:　リンクAでのロードベクタ
- $F_A^{(k)}(\boldsymbol{\beta}_A)$:　ノードAでのクラスkジョブに対する遅延の期待値の関数 (Fは、微分可能な凸型の増加関数)
- $G_A^{(k)}(\boldsymbol{\lambda}_A)$:　リンクAでのクラスkジョブに対する遅延の期待値の関数 (Gは、微分可能な凸型の非減少関数)

あるノード$A \in L_k$の内部のモデルを図2に示す。(なお中央ノードのモデルは、これから親へのリンクを取り除いたものである。)

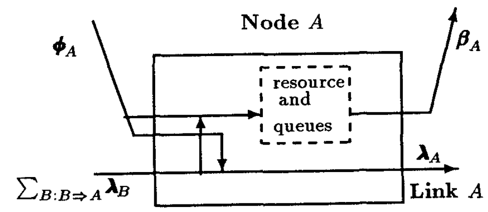

図2 ノード内部のモデル

3 平均応答時間

システム全体での平均応答時間 T は、

$$T(\beta) = \frac{1}{\Phi}\sum_{k=1}^{m}[\ \sum_{A:A\in Nd}\ \beta_A^{(k)}F_A^{(k)}(\boldsymbol{\beta}_A) + \sum_{B:B\in L_k}\ \lambda_B^{(k)}G_B^{(k)}(\boldsymbol{\lambda}_B)],$$

と表される。ただし、β は次の制約条件を満足しなければならない。

$$\begin{aligned} \beta_B^{(k)} + \lambda_B^{(k)} &= \sum_{C:C\Rightarrow B}\lambda_C^{(k)} + \phi_B^{(k)}, B \in Lk, \\ \beta_O^{(k)} &= \sum_{C:C\Rightarrow O}\lambda_C^{(k)} + \phi_O^{(k)},\ O\text{:Central node}, \\ \beta_B^{(k)} &\geq 0, \qquad B \in L_k, \\ \lambda_B^{(k)} &\geq 0, \qquad B \in L_k, \\ \phi_A^{(k)} &\geq 0, \qquad A \in N_d \end{aligned}$$

4 全体最適化方式

marginal node delay fと marginal communication delay gを、それぞれ

$$f_A^{(k)}(\beta_A) = \frac{\partial}{\partial\beta_A^{(k)}}(\beta_A^{(k)}F(\boldsymbol{\beta}_A)), A \in N_d$$

および

$$g_A^{(k)}(\boldsymbol{\lambda}_A) = \frac{\partial}{\partial\lambda_A^{(k)}}(\lambda_A^{(k)}G(\boldsymbol{\lambda}_A)), A \in L_k$$

と表すことにする。ノードAに到着したクラス k ジョブをBで処理する場合の marginal response time $t_{A\rightarrow B}^{(k)}$を、到着の種類(外部、内部)によらず、以下のように表す。

---

*Optimal load balancing in star network hierarchies.
Hiroyuki MACHIDA and Hisao KAMEDA
The University of Electro-Communications.

$$t^{(k)}_{A\to B}(\beta) = f^{(k)}_B(\boldsymbol{\beta}_B) + \sum_{C:A\overset{+}{\Rightarrow}C\overset{+}{\Rightarrow}B} g^{(k)}_C(\boldsymbol{\lambda}_C), A \overset{*}{\Rightarrow} B$$

定理 $T(\beta)$を最小にするノードAのクラスk負荷$\beta^{(k)}_A$は、以下の条件を満たす。

$$f^{(k)}_A(\boldsymbol{\beta}_A) \geq \alpha^{(k)}_B + g^{(k)}_A(\boldsymbol{\lambda}_A), \qquad \beta^{(k)}_A = 0, \qquad A \in R_d,$$

$$f^{(k)}_A(\boldsymbol{\beta}_A) = \alpha^{(k)}_B + g^{(k)}_A(\boldsymbol{\lambda}_A),\ 0 < \beta^{(k)}_A < (\sum_{C:C\Rightarrow A} \lambda^{(k)}_C) + \phi^{(k)}_A, \quad A \in R_a,$$

$$f^{(k)}_A(\boldsymbol{\beta}_A) \leq \alpha^{(k)}_B + g^{(k)}_A(\boldsymbol{\lambda}_A), \qquad \beta^{(k)}_A = (\sum_{C:C\Rightarrow A} \lambda^{(k)}_C) + \phi^{(k)}_A, \quad A \in S,$$

ただし、

$$\alpha^{(k)}_B = \min_{D:B\overset{+}{\Rightarrow}D} (t^{(k)}_{B\to D}(\beta)), \text{ および } \quad A \Rightarrow B,$$

であり、制約条件

$$\Phi = \sum_{k=1}^{m} \sum_{A\in N_d} \beta^{(k)}_A$$

が、満たされなければならない。

証明 省略。基本的には、亀田と櫨山の一般モデルの証明[2] を適用したものである。[4] を参照。

この最適状態において、ネットワーク内の各ノードは次の3通りに分類される。

Idel nodes. (R_d) このノードでは、自分の子ノードから送られてきたジョブと外部から到着したジョブを両方とも全く処理せず自分の親ノードに送る。(送られたジョブがその親のノードで処理されるかによらない。)

Active nodes. (R_a) このノードでは、自分の子ノードから送られてきたジョブと外部から到着したジョブの一部を処理し、残りを自分の親ノードに送る。(送られたジョブがその親のノードで処理されるかによらない。)

Sink nodes. (S) このノードでは、自分の子ノードから送られてきたジョブと外部から到着したジョブの全てを処理する。

5 計算アルゴリズム

亀田と櫨山[2] は、複数ジョブクラスへの拡張のなかで、対象とするクラス以外を固定して、各クラスごとにシングルクラスのアルゴリズムを繰り返すことにより、最適解が求められることを示した。

ここでは、ネットワーク内の兄弟関係にあるノードグループと元のジョブクラスを含めた以下のような新しいジョブクラスを考え、単一レベルのアルゴリズムの考え方を拡張する。

階層スター型ネットワークでは、中央ノードを除く全てのノードは、親ノードを持っている。この性質に注目して、共通の親ノードAを持つノード群を1つのグループと考え、これをC_Aとよぶ。**仮想ジョブクラス$C(A,k)$を、グループC_A内のノードに到着したクラスkのジョブの集合と定義する。** このような再クラス化により、全体での最適解が変わることはない。さらに仮想クラスの定義より、外部到着率$\hat{\phi}$に関してもし$B \in C_A$ならば $\hat{\phi}^{C(B,k)}_A = \phi^{(k)}_A$, それ以外では0という性質が成り立つ。これより仮想クラス$C(A,k)$内での最適化を行なう場合、C_Aとその祖先のノードだけ考えればよいことがわかる。

具体的なアルゴリズム次のようになる。

1. **初期設定** 仮想ジョブクラスの設定、およびそれにともなうパラメータの拡張を行う。拡張されたパラメータには、$\hat{\beta}$のようにハットをつける。さらに、実行可能なロードベクタ初期値を求め$\hat{\beta}'$に格納する。

2. **新しいロードベクタ$\hat{\beta}'$の計算** $\hat{\beta} = \hat{\beta}'$とする。さらにジョブクラス $k = 1, 2, ...m$ について、ステップ3を繰り返し行う。

3. **ジョブクラス k 内の最適解の計算** ノードの階層 $d = 2, 3, ...L$について、 ステップ3.1. を繰り返し行う。ここでLは、階層数を表す。

   3.1. 各ノード階層内のクラスk最適負荷分散の計算 与えられた階層dの各ノードAごとに以下を繰り返す。

   3.2. 仮想ジョブクラス$C(A,k)$内の最適負荷の計算 与えられたノードAとジョブクラスkのもとで、仮想ジョブクラス $C(A,k)$ 内の最適解 $\hat{\beta}^{C(A,k)}$ を計算する。ただし、ここでは他の仮想ジョブクラスは固定されている。このアルゴリズムは、単一レベルスター型ものの拡張として 示すことができる。

4. **全体平均応答時間の最適解への収束の検査** もし $|T(\hat{\beta}) - T(\hat{\beta}')| < \varepsilon$ ならアルゴリズムを終了する。そうでなければ、ステップ2を繰り返す。ここで ε は、前もって決めておいた誤差許容値とする。

6 まとめ

本研究において、階層構造を持つスター型ネットワークにおける最適負荷分散が、単一レベルスター型でのそれの一般化として示されることが明らかにされた。また、階層構造に基づく仮想クラスの導入により、最適解の計算アルゴリズムを単一レベルスター型の拡張として構成した。

参考文献

[1] A.N.Tantawi and D.Towsley. "A general model for optimal static load balancing in star network configurations.", PERFORMANCE '84 (Paris, Dec. 19-21, 1984) North-Holland, New York, 277-291.

[2] H.Kameda and A.Hazeyama. "Individual vs. overall optimization for static load balancing in distributed computer systems", Computer Science Report, The University of Electro-Communications, (1988).

[3] 亀田 壽夫, 張 勇兵. "スター型コンピュータネットワークにおける負荷分散", 情報処理学会オペレーティングシステム研究会研究報告, 38-2, (1988).

[4] 町田 浩之, 亀田 壽夫. "多重スター階層型コンピュータネットワークにおける負荷分散", 情報ネットワークのトラフィック評価に関する基礎的研究シンポジウム報文集, (Jan. 25-27, 1989).