

Latency や Gap のゆらぎを考慮した LogP モデルの検討

新家正総

(株)富士通研究所
〒211-8588 川崎市中原区上小田中 4-1-1
E-mail: ninom@flab.fujitsu.co.jp

本論文では、LogP モデル上で最適と評価される並列アルゴリズムの計算 / 通信スケジュールが、latency や gap に大きなゆらぎがある場合も最適なスケジュールとなるかどうかを検討する。検討は並列加算アルゴリズムを例にゆらぎの計算時間への影響を解析的に求めて行う。結果として、ゆらぎがある場合、従来の LogP モデル上で最適と評価される計算 / 通信スケジュールが実際は必ずしも良いスケジュールではなくなるを示し、更にゆらぎに応じて LogP モデルに用いる L の値を変えればより良いスケジュールが得られることを示す。

LogP、ゆらぎ、並列アルゴリズム、インターネット

Investigation of LogP model with fluctuation of Latency and Gap

Tadafusa Niinomi

Fujitsu Laboratories Ltd.
1-1, Kamikodanaka 4-Chome, Nakahara-ku, Kawasaki, 211-8588 Japan
E-mail: ninom@flab.fujitsu.co.jp

In this paper, we investigate whether optimal communication and computation schedules of parallel algorithms on LogP model are really optimal in case there is fluctuation of latency and gap. As an example of parallel algorithms we use a parallel summation, and analyze effect of the fluctuation to its calculation time mathematically. We found that the optimal communication and computation schedules on conventional LogP model are not always good if there is the fluctuation of the latency and the gap. We also found that the communication and computation schedules on LogP model can be improved if we choose the value of L according to the value of standard deviation of fluctuation.

LogP, fluctuation, parallel algorithm, internet

1. はじめに

LogP モデル[1]は並列アルゴリズムを検討するために提案された並列計算機のモデルである。ネットワークの latency、プロセッサが通信に要する overhead、メッセージを通信路に注入できる最小間隔 gap をプロセッサのサイクルを単位として定量的に表現することで、並列計算機をモデル化する。

従来 LogP モデルが対象としてきた並列計算機システムの多くは室内や構内に設置され専用の高速ネットワークを用いて CPU を接続するものだった。専用ネットワークで限られた規模の CPU を接続する場合、latency や gap を固定した値としてモデル化しても有益な並列アルゴリズムの検討ができた。

しかし近年、並列計算機システムを更に拡張するために、インターネットを用いて並列計算機を拡張する検討がはじまりつつある[2]。このようなシステムは米国 NGI[3] などが進めているインターネットの高速化が進むに伴って、実用化されるだろう。

では、LogP モデルはこのような、超高速インターネットを用いて規模を拡大した並列計算機システムにもそのまま適用可能であろうか。

ネットワークの規模が拡大すれば、latency や gap のゆらぎはそれに伴って大きくなるため、ゆらぎが計算時間に大きな影響を与える可能性がある。更には LogP モデル上で評価すると最適と考えられる計算スケジュールや通信スケジュールが、実は最適なスケジュールからは大きくはずれるかもしれない。しかしながら、このようなことがおきるかどうかは定かではなく、またゆらぎを考慮することでより良い計算/通信スケジュールが検討できるのかも現在明らかではない。

そこで本論文では、latency や gap のゆらぎが LogP モデル上で最適と評価された計算/通信スケジュールにどのような影響を与えるのか、またゆらぎを考慮すればより良い計算/通信スケジュールを得られるのかどうか検討する。

検討では並列計算機を用いて多くの数をなるべく短時

間で加算するアルゴリズム（並列加算アルゴリズム）を例にゆらぎの影響の検討を行い、ゆらぎの考慮が有効かどうか検証する。ゆらぎは標準偏差の正規分布をすると仮定し計算時間とゆらぎの関係を解析的に求める。更に関係式から、ゆらぎがある場合に latency の値としてどのような値を代表させて用いると、最適な計算/通信スケジュールが得られるようにできるのかを示す。

以下においては、2 節で検討の準備として LogP モデルと LogP モデルによる並列マシンを用いた加算のアルゴリズムを示す。3 節ではゆらぎをどのようにモデル化し、LogP モデルに反映させるべきかを検討する。4 節では加算のアルゴリズムにゆらぎがどのように影響するか検討する。5 節で結果をまとめる。

2. 準備

2.1. LogP モデル

LogP モデルは並列計算機を以下のパラメータで抽象的に表現する。

- L: latency。従来の LogP ではネットワークに負荷がかかっていない状態での latency の上限を用いる¹。
- o: overhead。プロセッサがメッセージの送受信にかかわる時間。時間の上限を用いる。
- g: gap。メッセージを最大スループットで連続送信/受信した場合のインターバル²。
- P: プロセッサ数。

ここで L、o、g はプロセッサのサイクルを単位として表現する。ここでサイクルとはプロセッサがローカルなオペレーション（例えば加算を一回するなど）をする場合の単位時間のことを言う。つまり LogP モデルでは、通信に関する時間をプロセッサの能力を基準として表現することで並列計算機をモデル化している。

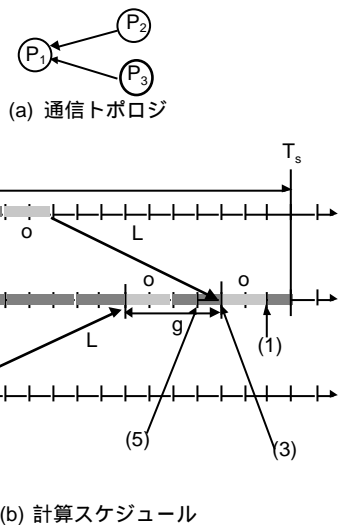
1 文献[5]においては平均を測定し L としている。

2 文献[1]において gap の定義はメッセージを連続送信/受信した場合の下限となっている。しかしこの下限という意味は最大スループットで連続転送する時の間隔という意味である。本論文では最大スループットが時間的にゆらぎ、文献[1]での「下限」がゆらぐことを考えているため、あえてこのように表現した。

2.2. LogP モデルによる並列加算

では、LogP モデルを用いてアルゴリズムがどのように検討されるかを並列加算を例として述べる。ここでは簡単に説明するために3つのプロセッサの場合を例とする。並列加算するには各プロセッサにデータを配置し計算するが、ここでは最初にデータを配置する方法の決定方法は説明から除き、通信のトポロジも図1(a)のように決まっていることを前提とする。4節で行う解析では最初にデータを配置する時間は検討の対象から外しており、通信のトポロジも決まっていると仮定している。また、以降の説明では加算を一回行う時間を1として考え、 L 、 o 、 g など全ての時間は加算を一回行う時間を単位として考える。

並列加算は n 個の値を最も短時間で加算するアルゴリズムである。加算に要する計算時間 T を考えるには、逆向きに考えて、 T の間にできるだけ多くの数を加算するにはどうすればよいか考えればよい。



図一 通信トポロジと計算スケジュール

今、もし T がプロセッサ間の通信遅延 $L+2o$ と同じか小さいとすると、他のプロセッサからデータを受け取る十分な余裕が無いので、最適な方法は $T+1$ 個の値を一つのプロセッサで加算することである。次に T が $L+2o$ より大きい場合に最適な方法を図1(b)を用いて説明する。この場合加算の最後のステップは、 $T-1$ において最後に加算を行うプロセッサ P_1 が加算した数と他のプロセッサ P_2 が加算した数を加えあわせることである(1)。従って P_2 は $T-1-2o-L$ の時点でデー

タをルートプロセッサに送らねばならない(2)。一方 P_1 は $T-1-o$ の時点で自分の加算を済ませる必要がある(3)。 P_1 は g サイクルでメッセージを受信できるから、 P_3 の加算は $T-1-2o-g-L$ で終了させねばならない(4)。ルートプロセッサ P_1 は P_3 からメッセージが到着するまでに自分のデータを加算する他、 P_2 と P_3 のメッセージが到着する間に $g-o-1$ の加算をしなければならない(5)。更に条件として、 P_1 は P_2 、 P_3 からの部分和を受信するのに o の処理時間を費やすから、少なくとも P_2 、 P_3 における和は o 以上でなければならない。これらの条件から、時間 T が与えられ、 L 、 o 、 g が与えられれば通信/計算のスケジュールとデータの配置をどのようにすべきかを決定することができる。

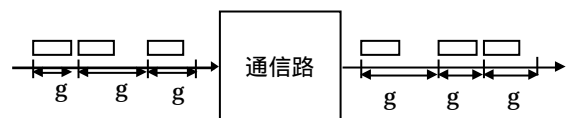
3. ゆらぎを考慮した LogP モデル

3.1. ゆらぎのモデル化

以下ではゆらぎをどのようにモデル化するか考える。

まず latency のゆらぎは正規分布すると仮定して解析を行う。現実のインターネットにおいても、遅延が正規分布することが観測されている[4]。

次に gap のゆらぎについて考える。gap はメッセージを最大スループットで連続送信/受信した場合のインターバルである。従来の LogP では、gap が時間的にゆらぐことや、ゆらぎの影響は考慮していなかった。しかし現実の通信路において最大スループットで通信を行う場合、フローコントロールのために送信側が一定間隔で送れない場合がある。また送信側でパケット間隔のゆらぎが無かったとしても、受信側に到達するまでに latency のゆらぎによってパケットの間隔がゆらぐことがある。



図一 2 gap のゆらぎ

今回の検討では、検討を簡単にするために送信側の gap のゆらぎは検討に入れず、gap のゆらぎは全て latency のゆらぎによるものとする。このように考えると、ゆらぎの検討は latency に絞って行うことがで

きる。但し gap の値には送信側、受信側で最小値があり、パケットはそれ以下の間隔では送信 / 受信できないことを検討の考慮に入れる。

3.2. ゆらぎの考慮

では latency のゆらぎがある場合、ゆらぎのパラメータをどのように定義し、どのように LogP モデルに取り込むことができるだろうか。

一つの考え方は、ゆらぎの大きさから L の値として LogP モデルで使うべき代表値を決定できるようにすることである。この代表値はゆらぎの下で計算時間を最小にできるよう選ぶべきである。具体的には以下のようにして選ぶことができるだろう。

1. L の代表値を L_s とする。まずゆらぎが無い場合 ($L=L_s$ の時) の計算時間 T_s を求める。
2. 実際の latency L が L_s から ずれた時を考え、その時の計算時間 T を求める。 T は T_s と の関数となる。
3. L が平均 μ 、標準偏差 σ の正規分布に従うとして、 T の平均を μ と の関数として求める。
4. T の平均 μ は L の平均値 L_{ave} と L_s の差である (図3) から、 T が最小になるような μ を求めれば、 T を最小にする L_s が求まる。

以降においては、並列加算のアルゴリズムについてこの計算を実際に行う。

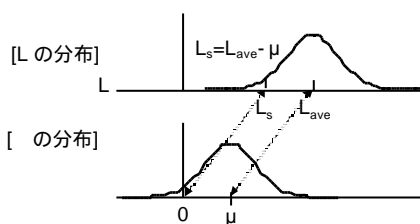


図 - 3 L_s 、 μ 、 L_{ave} の関係

4. 並列加算アルゴリズムへのゆらぎの影響

4.1. 検討の条件

以降の計算はプロセッサにデータが配置されたあとの計算時間に絞って行い、latency のゆらぎと計算時間の関係を解析的に示す。なおデータを転送するための通

信トポロジは既に決定していると考え、計算の前にデータをプロセッサに配置する時間は検討から除いている。

4.2. プロセッサが2つの場合

まず最も単純な場合である、プロセッサが2つしか無い場合を考える。

n 個の数を加算する計算時間 T を求めるには、2.2 節で説明したように T の間に計算できる最大数 n を求めるように考えれば良い。

ゆらぎがある場合を求める前に、まずゆらぎが無く、 $L=L_s$ となる場合の計算時間 T がどうなるかを考える。このとき、2.2 節のように考えると最適な計算スケジュールは図4のようになる。

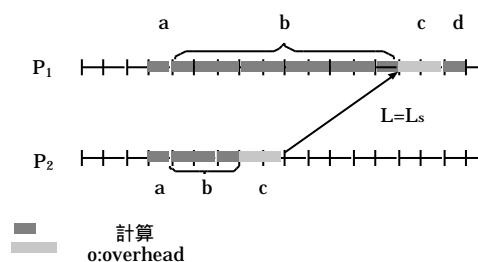


図 - 4 加算スケジュール ($L=L_s$ 、 $P=2$)

ここで T の間に P_1 、 P_2 でそれぞれ加算した数はいくつか考える。 P_1 で計算できる数の個数は、 a の時点で2つの数を加算し、 b の間は前の結果に1つ数を加算していく。 c の時点では通信のため加算はできない。 d の時点では P_2 からもらった数を自分の計算結果に加えるため、計算は行うものの自分のもつデータを新たに加えることはしていない。したがって P_1 自身で加算したデータの数は $T-o-1+1=T-o$ となる (最後の+1 は a の時点で2つ加算しているためである)。一方 P_2 で計算できる数を同様に考えると $T-2o-L-1+1=T-2o-L$ となる。よって、 T の間に計算できる最大数 n はこれらの合計であり

$$n = 2T - 3o - L \quad (4-1)$$

となる。さてここで考え方をもとに戻し、 n 個の数を解くための最大時間を T と考えると、上式を T について解けば良く

$$T = \frac{n+3o+L}{2} \quad (4-2)$$

となる。

では今度は L にゆらぎがある場合を考える。ある値 L_s を L の代表値として選んだとして、実際の L が $L_s +$ となった場合、T は L_s をもとにした計算からどのように変化するのであろうか。

L が L_s より小さかった場合 (図5)、つまり予定より L が小さかった場合、 P_1 が P_2 のデータを受け取る時間は予定より早くなるものの、もともと P_1 で計算する数の量は決まっており、 P_1 の計算が早まるわけではない。したがって全計算時間 T に変化は無い。

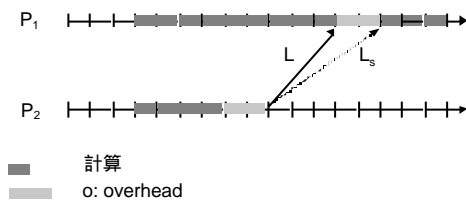


図 - 5 予定より遅延が小さい場合

逆に L が L_s より大きかった場合、 P_1 は自分が計算すべき数の計算は全て終了し P_2 からの計算結果を待つことになる。結果として最終的に計算が終わるのは、予定時間よりもだけ遅れてしまう。

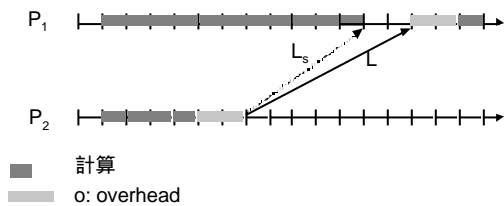


図 - 6 予定より遅延が大きい場合

つまり、L が L_s より大きくなると T が増加するが、逆の場合は T は変化しないことになる。従って を考慮した計算時間 T は以下ようになる。

$$T = \begin{cases} T_s + \mu & (\mu > 0) \\ T_s & (\mu < 0) \end{cases} \quad (4-3)$$

但し T_s は、 μ が 0 で $L=L_s$ の時の T であり、

$$T_s = \frac{n+3o+L_s}{2} \quad (4-4)$$

である。

さて、今度は T の平均を求めよう。 μ は平均、標準偏差 σ の正規分布をすると仮定したので、 $p(x)$ の確率密度関数は以下ようになる。

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4-5)$$

T の平均は次の式で求められる。

$$E[T] = \int_0^{\infty} T p(x) dx \quad (4-6)$$

この積分は x が 0 以上の場合と 0 より小さい場合に別けて実行することで求めることができ、結果は以下のようなになる。

$$E[T] = T_s + \mu \left(\frac{\mu}{\sigma^2} + \frac{1}{\sigma} \right) \quad (4-7)$$

となる。但し $\phi(x)$ は標準正規分布関数、 $\Phi(x)$ はその累積分布関数であり、以下のように定義される。

$$\begin{aligned} \phi(x) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \\ \Phi(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt \end{aligned} \quad (4-8)$$

さて、 $E[T]$ が最小になるような μ を求める。まず T_s が L_s の関数であり、 L_s は $L_{ave} - \mu$ であることを考慮する。 $E[T]$ は以下のように書ける。

$$E[T] = \frac{n+3o+L_{ave}}{2} + w \quad (4-9)$$

但し w を以下のように定義した。

$$w = \frac{\mu}{\sigma} \left(\frac{\mu}{\sigma} + 1 \right) - \frac{1}{\sqrt{2\pi}} \quad (4-10)$$

図 8 に w を (μ/σ) の関数としてグラフ化したものを示す。 μ が 0 の時、w が最小となることがわかる。また $E[T]$ の最小値は $1/\sqrt{2}$ (約 0.4) となる。

つまりプロセッサが2つの場合の並列加算では、 L_s として L の平均を用いれば最適な計算スケジュールが得られるのである。ただし、計算時間 T は L の平均を用いて LogP モデルで計算したものより、0.4 大きくなることわかる。

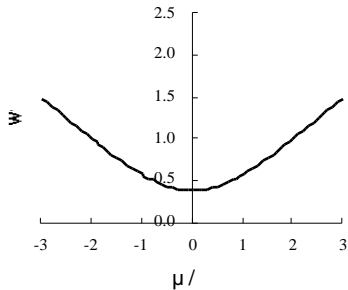


図 - 8 w と $(\mu/)$ の関係

L の平均を用いることが最適な計算スケジュールをもたらすというこの結果がもし一般に成り立つならば、計算スケジュールの検討においてゆらぎは検討しなくてもよいことになり、検討する立場からはとても好ましいことになる。しかし残念ながらこの結果は一般には言えない。プロセッサが3つの場合で考えると、最適な計算スケジュールをもたらす L_s は L に依存することになる。次にこのことを示す。

4.3. プロセッサが3つの場合

図7にプロセッサが3つの場合の計算スケジュールを示す。プロセッサ3つの場合はプロセッサ2つの場合に比べモデル化が相当複雑になる。 P_1 は P_2 と P_3 の2つから計算結果を受け取る。 P_2 から P_1 への latency と P_3 から P_1 への latency は一般的には異なるのでそれぞれ L_{21} 、 L_{31} とし、latency の代表値を L_{21s} 、 L_{31s} とする。

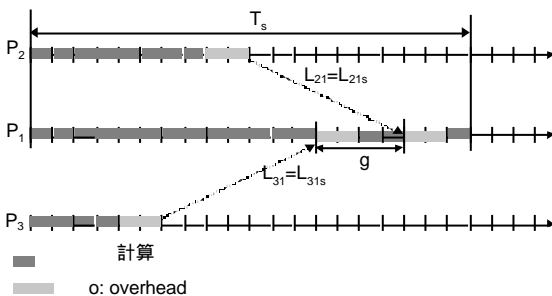


図 - 7 加算スケジュール (P=3)

まずゆらぎが無いときの計算時間 T_s を前節と同様に求めると以下ようになる。

$$T_s = \frac{n+6o+1+L_{21}+g+L_{31}}{3} \quad (4-11)$$

次にメッセージがゆらいだ場合を考える。 L に対するゆらぎを σ_{21} 、 L に対するゆらぎを σ_{31} とする。 σ_{21} と σ_{31} の値によって6つのケースが考えられる。

1. $\sigma_{31} < \sigma_{21} < 0$ の場合 (図9) は、4.1 節の図5の場合と同様で P_1 で計算する時間に変化は無いので全体の計算時間 T は σ_{31} 、 σ_{21} に影響されない。

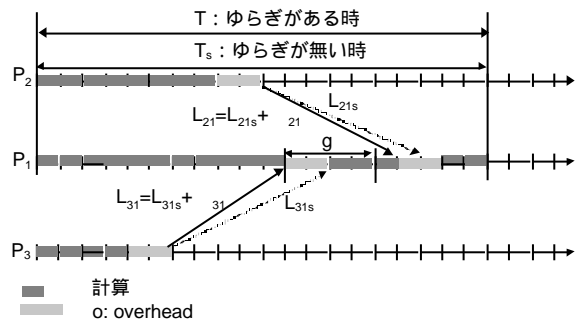


図 - 9 $\sigma_{31} < \sigma_{21} < 0$ の場合

2. $\sigma_{21} < \sigma_{31}-g < \sigma_{31} < 0$ の場合 (図10) は、 L_{21} は $L_{21s} + \sigma_{21}$ となる。また gap の存在により、 L_{31} は $L_{31s} + \sigma_{31}$ とならず、 $L_{21}+g$ となり、メッセージの順序は入れ替わることになる。しかし全体の計算時間 T は変わらない。

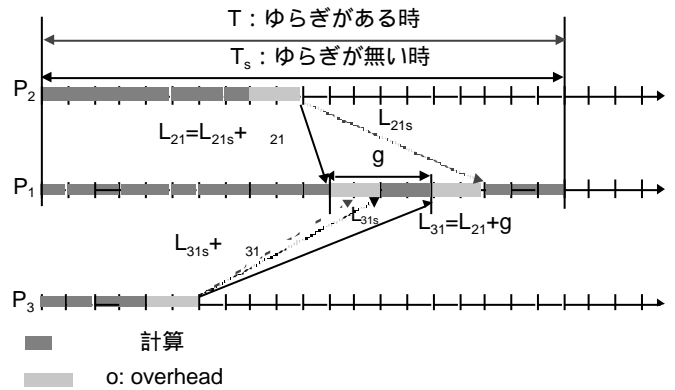


図 - 10 $\sigma_{21} < \sigma_{31}-g < \sigma_{31} < 0$ の場合

3. $\sigma_{31}-g < \sigma_{21} < \sigma_{31} < 0$ の場合 (図11) は、gap の存在により、観測される L_{21} は $L + \sigma_{21}$ とならず $L_{31}+g$ になってしまう。しかしこの場合も、 P_1 で計算時間に変化は無いから全体の計算時間 T

は変わらない。

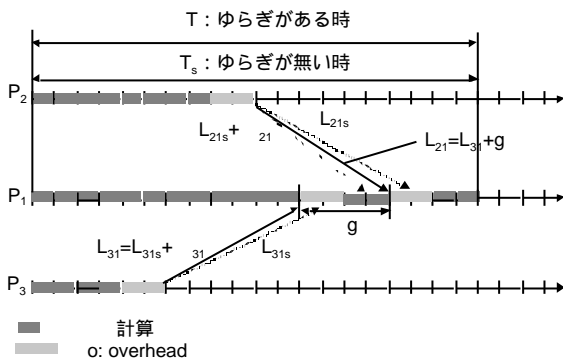


図 - 1 1 $31 < g < 21 < 31 < 0$ の場合

4. $21 > 0$ 、 $21 > 31$ の場合 (図 1 2)、 P_1 にあるデータと P_3 のデータの加算が全部終わっても P_1 は P_2 からのデータを待つことになり、全体の計算時間は 21 だけ大きくなる。

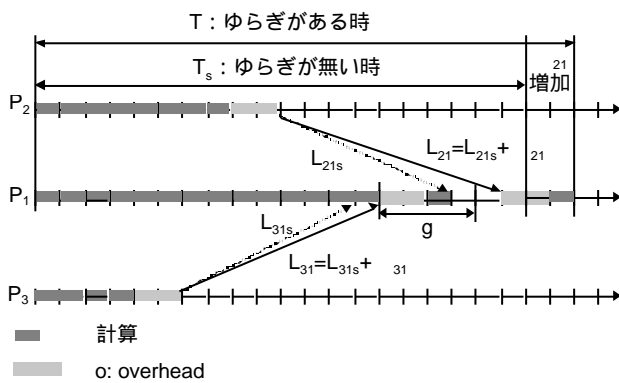


図 - 1 2 $21 > 0$ 、 $21 > 31$ の場合

5. $31 > 0$ 、 $31 > 21+g$ の場合 (図 1 3)、メッセージが入れ替わり、全体の計算時間は 31 だけ増える。

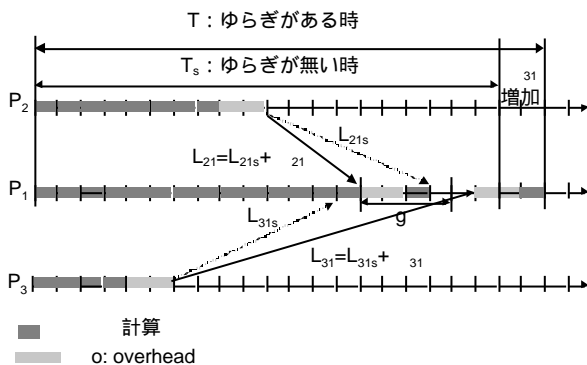


図 - 1 3 $31 > 0$ 、 $31 > 21+g$

6. $31 > 0$ 、 $21+g > 31 > 21$ の場合 (図 1 4)、メッセージが入れ替わる。更に gap の影響により、

L_{31} は $L_{31s} + 31$ とはならず、 $L_{21}+g$ になる。この時全体の計算時間は $21+g$ だけ大きくなる。但し計算にあたっては、この場合の計算時間を 31 だけ増えるとして近似的に考える。従って計算時間は小さ目の見積もりとなる。このように考えても g が latency のゆらぎに対して小さければ、結果に大きな影響は出ないと考えられる。この簡略化により、前の 5 のケースと 6 のケースは一緒にでき、 $31 > 0$ 、 $31 > 21$ の場合計算時間が 31 だけ増えると考えることができる。

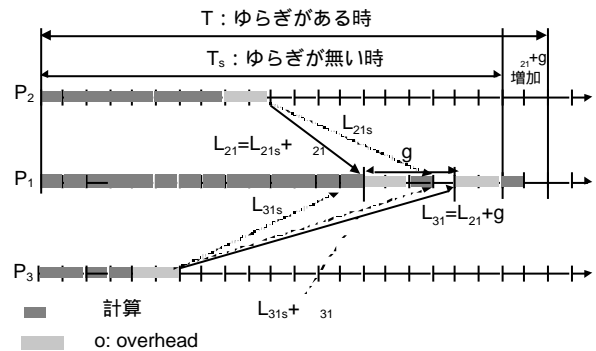


図 - 1 4 $31 > 0$ 、 $21+g > 31 > 21$

さて、以上の場合分けをふまえてゆらぎがある場合の T を定式化すると、結局 T は以下ようになる。

$$T = \begin{cases} T_s + 21 & (21 > 31 \text{ and } 21 > 0) \\ T_s + 31 & (31 > 21 \text{ and } 31 > 0) \\ T_s & (21 = 0 \text{ and } 31 = 0) \end{cases}$$

(4-12)

T の平均は次の式で表される。

$$E[T] = \int_0^{\infty} T p_1(t) dt + \int_0^{\infty} T p_2(t) dt$$

(4-13)

但し $P_1(\cdot)$ 、 $P_2(\cdot)$ は正規分布関数で以下のように表される。

$$p_1(t) = \frac{1}{\sqrt{2\pi} \sigma_1} e^{-\frac{(t-\mu_1)^2}{2\sigma_1^2}}$$

$$p_2(t) = \frac{1}{\sqrt{2\pi} \sigma_2} e^{-\frac{(t-\mu_2)^2}{2\sigma_2^2}}$$

(4-14)

この積分を行うにあたり、以下においては 31 と 21 の標準偏差が同じ場合を考える。つまり $31 = 21 = \sigma$ と置く。また T は 31 と 21 について対称な形をして

おり、標準偏差が同じならば $E[T]$ を最小にする μ_{31} 、 μ_{21} は等しいと考えられるので $\mu_{31} = \mu_{21} = \mu$ と置く。この条件の下に積分を実行して T を求めると、結果は以下ようになる

$$E[T] = T_s + 2 \frac{\mu}{\sqrt{2}} + 2 \frac{\mu}{\sqrt{2}} + \frac{\mu^2}{2} \quad (4-15)$$

さて、 $E[T]$ が最小になるような μ を求める。まず T_s を μ の関数として表すと、 $E[T]$ は以下のように書ける。

$$E[T] = \frac{n+6o+1+2L_{ave}+g}{3} + w \quad (4-16)$$

但し

$$w = 2 \frac{\mu}{\sqrt{2}} + 2 \frac{\mu}{\sqrt{2}} + \frac{\mu^2}{2} - \frac{2}{3} \mu \quad (4-17)$$

とした。図 15 に w のグラフを示す。プロセッサ 2 つの場合に比べ、左上にグラフがシフトしている。

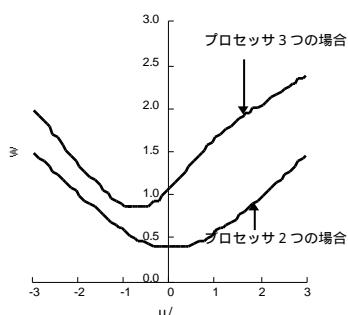


図 - 15 w と (μ/λ) の関係

w を最小にする μ は数値計算で求められ、 -0.76 となる。このとき w の最小値は 0.87 となる。

これらの結果から、プロセッサが 3 つの場合には、 L_s として選ぶべき値は L の平均値ではなく、平均よりも 0.76 大きいことがわかる。プロセッサ 3 つの場合、

μ_{21} と μ_{31} のどちらかが 0 以上ならば T が増加するため、プロセッサ 2 つの場合よりも T に対する L の寄与が大きくなる。このために L の平均がモデルを作る上で最適とはならなくなると考えられる。また、プロセッサ数が多くなればなるほど L の寄与が更に大きくなるため、更に平均よりも大きな値を使う必要があ

ると予想される。

5. まとめ

Latency や gap のゆらぎにより、従来の LogP モデル上では最適な計算 / 通信スケジュールが実際は最適でなくなる場合があるのかどうか、またゆらぎの考慮により従来の LogP モデル上で検討するよりも良い計算 / 通信スケジュールの検討が可能かどうかを並列加算のアルゴリズムを例として検討した。検討の結果として以下のことがわかった。

- プロセッサが 2 つの場合は latency の平均値を用いれば最適な計算 / 通信スケジュールとなる。
- プロセッサが 3 つの場合は latency のゆらぎをとした時、latency の値として平均より 0.76 大きな値を用いるべきである。
- プロセッサ数が多くなれば更にゆらぎの影響が大きくなると予想される。

これらの結果から、インターネットのような大規模なネットワークを用いて並列計算機の規模を拡大する場合、従来の LogP モデル上で最適となる計算 / 通信スケジュールでもゆらぎがあると良いスケジュールとは言えなくなる場合があることがわかり、ゆらぎの考慮が重要になることがわかった。

この結果を踏まえ、今後はよりプロセッサが多いケースでも利用できるような方法を検討していきたいと考えている。

参考文献

- [1] D. E. Culler, R. M. Karp, D. A. Patterson, A. Sahay, K. E. Schauer, E. Santos, R. Subramonian, and T. von Eicken "LogP: Towards a Realistic Model of Parallel Computation", Proceedings of 4th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, May 1993.
- [2] 陣崎、林、中村、村井：大規模広域並列分散システムの実現を目指す超高速インターネットの構想、信学技報 CPSY97-61, 1997 年 8 月
- [3] <http://www.ngi.gov/>
- [4] 古賀、陣崎：Comet による IEEE1394 を利用した計算機ネットワークの構築、SWoPP'98 CPSY, 1998 年 8 月
- [5] D. Culler, L. T. Liu, R. P. Martin, and C. Yoshikawa "LogP Performance Assessment of Fast Network Interfaces", IEEE Micro, Feb. 1996