分散pub/subにおけるsubscriber配置に関する一検討

A Study of Optimal Placement of Subscribers on Distributed Pub/Sub Systems

坂野遼平 [1] 竹内亨 [1] 川野哲生 [1] 武本充治 [1]
Ryohei Banno Susumu Takeuchi Tetsuo Kawano Michiharu Takemoto

日本電信電話株式会社 NTT 未来ねっと研究所 [1]
NTT Network Innovation Laboratories, NTT Corporation

1 はじめに

我々は Multi-key Skip Graph [1] を用いたスケーラブルな分散トピックベース pub/sub 機構を提案している [2]. データセンタ等における活用として，各々が 1 つのトピックに属する多数の subscriber（スマートフォン等）に対し，それらを収容するために必要十分な台数のサーバ（ノード）を用意し，各ノードに均等な数を収容する状況を想定した場合，ノード群に対する subscriber の割り当て方が効率に影響を及ぼす.

割り当て方には，大きく 2 通りの方式が考えられる. 同一トピックの subscriber を可能な限り同じノード上に集約する**集約配置**と，同一トピックの subscriber を可能な限り多くの異なるノードに散在させる**分散配置**である.

集約配置の場合，以下の利点が得られると考えられる.

- publish 時の同報先減少による，ネットワークリソース（帯域，スイッチ負荷等）の消費抑制.
- 各ノードの経路表サイズ縮小による，メモリ消費及び管理コストの抑制.

即ち，ノードやネットワークのリソース消費の観点からは，集約配置が望ましい.

一方で，個々のノード内における処理（subscriber 毎のキューへの書き出し等）の時間を短縮するためには，分散配置によって処理を分散化することが望ましいが，分散配置の場合 publish 時の転送経路長が長くなるため，publish が全 subscriber に届くまでの総遅延時間は必ずしも短縮されるとは限らない.

従って，subscriber の最適な配置を検討するには，まず処理遅延に関する 2 方式間の優劣を明確化する必要がある. 本稿では，これら 2 方式間の処理遅延の差を定式化し，subscriber 配置の最適化について考察を加える.

2 処理遅延のモデル化

本稿で用いる記号を表 1 のように定義する. なお以降では，Multi-key Skip Graph によって，トピック毎に publisher を根ノードとした二分木が構成されていると仮定する.

表 1 記号の定義

s_{all}	subscriber の総数
s_t	トピック t の subscriber 数
n	ノード数
p_t	トピック t の人気度（$s_t \div s_{all}$）
s_{unit}	ノードあたりの subscriber 収容数（$s_{all} \div n$）
t_c	ノード間の通信遅延
t_p	ノード内の 1 subscriber あたりの処理遅延

トピック t の subscriber を集約配置した場合に，publish の配送と各ノードにおける内部処理が完了するまでの遅延時間 T_{aggr} は，$s_t \geq s_{unit}$ の時，次式で表すことができる.

$$T_{aggr} = t_c * \log_2 \lceil s_t / s_{unit} \rceil + t_p * s_{unit}$$

一方，subscriber を分散配置した場合の遅延時間 T_{dist} は，$s_t \geq n$ の時，次式で表される.

$$T_{dist} = t_c * \log_2 n + t_p * s_t / n$$

簡略化のため天井関数を外して考えると，集約配置と分散配置の差 T_{diff} を次式で表すことができる.

$$T_{diff} = T_{aggr} - T_{dist} = t_c * \log_2 p_t + t_p * (1 - p_t) * s_{unit}$$

3 考察

以降では，$p_t \geq 1/n$ かつ $p_t \geq 1/s_{unit}$ であるものとする. $t_c = 1msec$, $t_p = 0.001msec$ とし，異なる複数の s_{unit} について上式を p_t の関数としてプロットすると，図 1 のようになる.

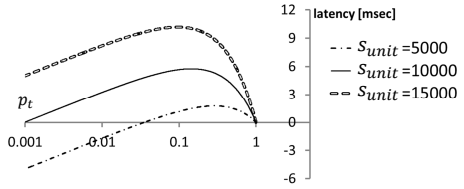

図 1 p_t に対する T_{diff} の遷移

p_t が大きければ T_{diff} は正の値をとり，特に s_{unit} が大きいほど T_{diff} が増大する. T_{diff} が正である範囲では，遅延時間に関しては分散配置に優位性があるため，集約配置によるリソース消費面での優位性とのトレードオフを考慮して subscriber 配置を設計する必要がある.

一方，p_t が小さくなると t_c 項が支配的になり，T_{diff} が負の値をとるようになる. この場合は，リソース消費と遅延時間の双方の点で集約配置に優位性があると言える. トピック集合に対する p_t の分布がべき乗則に従うことを想定すると，p_t の値に応じて集約配置と分散配置とを切り替えられることが望ましい. 閾値となるのは $p_t < 1$ において $T_{diff} = 0$ となる点であるが，各ノードは，t_c, t_p, s_{unit} の値が所与であれば，これを以下のように算出することができる.

$$p_t = W(z * e^z)/z \text{ where } z = -(\log_e 2/t_c) * t_p * s_{unit}$$

なお，W はランベルトの W 関数である. 例えば，$t_c = 1$ msec, $t_p = 0.001$ msec, $s_{unit} = 10,000$ の場合，$p_t \approx 0.000983$ となる.

4 おわりに

本稿では，Multi-key Skip Graph を用いた分散トピックベース pub/sub において，subscriber の集約配置と分散配置との処理遅延時間の差を定式化し，両方式の優位性が切り替わる閾値の存在を示した.

参考文献

[1] 小西他, “単一ノードに複数キーを保持可能とする Skip Graph 拡張,” 情報処理学会論文誌, 2008.

[2] 坂野他, “ストリームデータの配送に向けた分散トピックベース Pub/Sub 手法の提案,” 電子情報通信学会技術研究報告, 2013.