2007－HPC－112 (1)
2007／9／9

インターネット上での分散時刻認証グリッドの タイムスタンプ発行スケーラビリティの評価

西川 武志[†] 松岡 聡[†,‡]

[†]東京工業大学学術国際情報センター 〒152-8550 東京都目黒区大岡山 2-12-1

[‡]国立情報学研究所 〒101-8430 東京都千代田区一ツ橋 2-1-2

E-mail: [†]t.nishikawa@gsic.titech.ac.jp, [†‡]matsu@is.titech.ac.jp

これまで我々は既存の単一点時刻認証局や分散時刻認証局の持つ問題を解決する K=L+M among N for G 世代分散時刻認証法を提唱し,プログラムを実装し, 毎秒百万タイムスタンプ発行が可能であることを, LAN 環境等, 低遅延, 高バンド幅の環境で示して来た. 今回, NTT 東日本 B-Flets や欧州の WiFi インターネット接続サービス等のネットワーク上に時刻認証ユニット(TSU)を設置し, インターネット上での分散時刻認証グリッドシステムの動作実験を行った.その結果, 突発的なネットワーク遅延や Java VM のガーベージコレクションによる応答遅延が存在しても,十分な数の TSU が存在すれば, インターネット上でも毎秒百万タイムスタンプ取得発行の可能性があることを示した.

Evaluation of the issue of time stamps scalability of the distributed time stamping authority grid on the Internet

Takeshi Nishikawa[†] and Satoshi Matsuoka[†,‡]

[†]Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550

[‡]National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430

E-mail: [†]t.nishikawa@gsic.titech.ac.jp, [†‡]matsu@is.titech.ac.jp

We have previously proposed a distributed time stamping scheme called "K=L+M among N for G generation" that solved the problem of scalability that both a centralized TSA (Time Stamping Authority) as well as other previous distributed time stamping schemes exhibited, and moreover implemented and tested its viability in issuing one million time stamps per second on a LAN testbed environment which has the low latency and the high bandwidth. To verify the globlal scalability of our approach, we install the distributed time stamping units (TSU) in various locations on the Internet with varying access characteristics, such as the NTT East B-Flets network (regional shared optical 100Mbps best effort) as well as a European WiFi Internet service provider network. There, realistic operational experiment of the distributed time stamping grid system exhibited good scalability in that sufficient number of TSUs distributed on the Internet allows issuance of one million time-stamps per second even if there are the unpredictable network delay and/or the response delay by garbage collection of Java VM, just as was the case under a LAN environment.

1. はじめに

デジタル時刻認証とは、その時刻に該当デジタルデータが存在していたこと（存在性証明）、その時刻以降に該当デジタルデータが不正に改ざんされていないこと（完全性証明）を行うものであり，IETF が RFC3161 として策定した PKI を利用したタイムスタンプのプロトコル・フォーマットが広く商用化され利用され始めている[1]. その重要性は法律で規定された保存電子帳票の正当性証明や内部統制での種々の記録の非改ざん証明手段として用いられている.

デジタル時刻認証は科学技術分野でも知的財産権の優先権確保や不正が行われていないことの証明に役立つことが期待されて、デジタル実験ノートなどのシステムが開発されている.ところが１タイムスタンプあたりの費用が数円程度のため,１ヶ月に数百万ー数千万のタイムスタンプを必要とするようなデジタル実験ノートでは膨大なコストがかかってしまう. また, 既存のデジタル時刻認証局はネットワーク上の単一点でサービスが提供されるため, 性能スケーラビリティ, 耐障害性等に問題が潜んでいる. 性能スケーラビリティ, 耐障害性等の対策として様々な分散時刻認証法が提唱されているが[2-4] , これらの分散時刻認証は, 複数の時刻認証ユニット(TSU)を用いて時刻認証局を構築することで性能スケーラビリティや耐障害性を確保するものの, 実際の時刻認証局の運用において必須とされている, TSU が時刻源と時刻は正しく同期されているか, 運用規定に従って正しく運用されているかを担保する監査のプロセスに触れていない.この監査プロセスが高コストであることが, 既存の単一拠点時刻認証局のタイムスタンプ発行コストを押し上げていることへの解決案を提供していない.

これまで我々は複数 TSU が一つの時刻認証要求に対し, 複数世代に渡って相互に発行・再要求する K=L+M among N for G 世代法を考案し, それを用いて相互監査機能を内在した分散時刻認証グリッドの仕組みを実装（ソフトウェア名 tsagrid）し, 高速内蔵ネットワークを有したクラスタ, Gigabit Ethernet LAN 上のクラスタ, 産総研および東工大間のグリッド実験線上のクラスタ等を使い基本動作や毎秒百万タイムスタンプ発行の可能性等を検討し十分な数の TSU を用意すれば, 信頼出来る TSU 同士で毎秒百万タイムスタンプ発行が可能であることを示したてきた[5-7].

さらに TSU 群を一般のインターネット環境, NTT B-Flets（日本国内）, SINET（東工大,阪大）, WIDE（慶大）上に固定設置するとともに, 著者の西川がドイツ, スイス,フランスを訪れた際に持参したノートパソコン上でも tsagrid により TSU を動作させ実験を行った. その結果, SFR-WiFi（パリ,フランス）から NTT B-Flets 上の TSU に対し tsagrid クライアントプログラムによるタイムスタンプ要求の送受信時間測定では 412ms であった.これにより, 十分な数の TSU がインターネット上に分散配置されていれば, 毎秒百万タイムスタンプが可能であることが分かった（図１）[8]. ただし、このとき TSU の内部動作に依存する遅延, すなわち, Java VM のガーベージコレクション（GC）による応答遅延やインターネット上のネットワーク遅延が時折発生し, 毎秒百万タイムスタンプ発行を阻害するような動作例も観測された.

今回, 我々はインターネット上での分散時刻認証グリッドのタイムスタンプ発行スケーラビリティの評価を,一歩進めて, TSU の内部動作に依存する遅延やインターネット上のネットワーク遅延の生起確率を勘案して行った.

その結果, TSU の内部動作に依存する遅延やインターネットのネットワーク遅延を考慮しても毎秒百万タイムスタンプ発行のスケーラビリティを持つことが出来る条件設定が可能と推定される結論を得た.

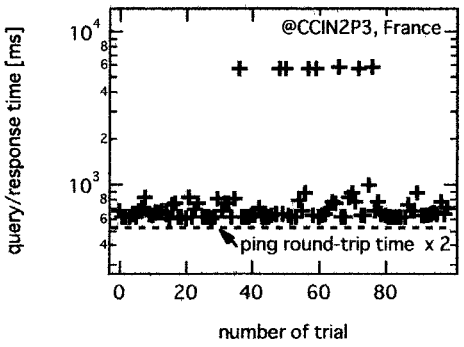

図1 タイムスタンプの発行要求からログの記録までの時間測定例(フランスCCIN2P3にて)

2. K=L+M among N for G 世代法

我々が提唱するK=L+M among N for G世代分散時刻認証手法の発行プロセスについて説明する．クライアントからのタイムスタンプの発行要求からログの記録までの詳細は次の通りである．

mk hash：G=0すなわちクライアントにおいてデジタルデータからハッシュを作成する．

dispatch tsq：利用者はRFC3161に従って時刻認証を行いたいファイルのハッシュからタイムスタンプクエリを作成し，RFC3161のタイムスタンプ要求(TSQ)をG=1である root TSUに発行する．発行を要求した時刻を t_0 とする．

Stamping tsr：root TSUは，時刻 t_1 を署名したタイムスタンプ応答(TSR)を生成し，a)非同期にTSRを要求元に返す．次世代のK個のTSUはL個の信頼できるものと(N-1-L)個のTSUからランダムに選んだM個のものから選択しTSQの発行準備をするとともに，拡張プロトコルで保持している現在何世代目かを記録するフラグを1減じ，次世代TSUへTSQを発行する．

Logging：次世代TSUによるTSR生成，応答が行われ，非同期にTSRを要求元に返す.受け取った前世代TSUはTSRを手元に保存し，どの次世代TSUからのTSRであったかを記録する.従ってタイムスタンプ時刻はTSQがTSUに届いて処理される過程で認証されるため，TSQ要求からTSR処理，ログの記録までの全体の処理が終わった時刻よりも前の時刻が記録される.タイムアウトになっていないか，世代数が上限に達していないかを確認し，いずれでも無い場合，再び次世代のTSU，K個をこれまで述べたアルゴリズムに従って選択しTSQの発行準備をするとともに，現在何世代目かを記録するフラグを1減ずる.

以上のmk hash, dispatch tsq, stamping tsr, loggingををG上限まで繰り返していく.

以上のG=2までを図示したものを図2に示す.

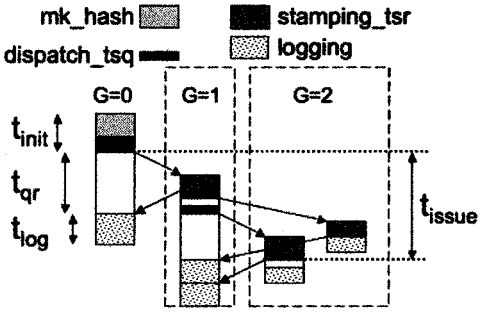

図2 タイムスタンプの発行要求からログの記録までの動作概要

G=0におけるmk hashからdispatch tsqまでを t_{init}，dispatch tsq終了時からstamping tsrを経てloggingの直前までを t_{qr}，logging時間を t_{log} とした。一方で、毎秒百万タイムスタンプ発行のスケーラビリティを評価する場合は，t_{init} 後から最後のG世代でのstamping tsrが終了するまでの時間 t_{issue} を発行時間として定義した．Gが3以上の時，各々のG世代のTSUは(G-1)世代から受け取ったタイムスタンプ要求を自己と(G-1)世代の直接上流のTSUを除いた(N-2)のTSUの内からK=(L+M)個を選んでこれまでと同様に世代数を減じて転送要求するその結果，各G世代で1,K,K2, ..., KG-1個のタイムスタンプ要求がなされ，全体では $(K^G-1)/(K-1)$ 個のタイムスタンプ要求がなされる．その結果，最大で $(K^G-1)/(K-1)$ 個のタイムスタンプが生成されるが，現実の実装では途中のネットワーク遮断等なんらかの理由により応

答が無い TSU があることが考えられる．3 世代以上の間ではタイムスタンプ要求がループすることがあり得る．

本 TSA Grid で拡張したプロトコルでは各 TSU はどの TSU からのタイムスタンプ要求を受け応答したか，そのタイムスタンプ要求をどの子世代の TSU に転送したか，それが何世代目だったかを記録する．

このようにして各 TSU は親 TSU にタイムスタンプ応答を返し，子 TSU からタイムスタンプを受け取る．受け取った子 TSU からの最大 K 個のタイムスタンプに署名された時刻を使って信頼出来る L 個の TSU のリストを更新する．

TSA Grid としての時刻認証は root TS に署名された時刻 t_1 に対し Δt の各種統計処理値（平均値，分散，最頻値，etc.）によって表現された時刻によってなされる．

以上の仕様を満たす tsagrid プログラムは Java および Bouncy Castle API を用いて Servlet として実装し Tomcat 上で動作するようにした．Tomcat は Servlet/JSP アドオンとして Apache 上で動作させたり，単独で Web サーバとして動作させたりすることが可能である．

3. 動作実験

インターネット上での分散時刻認証グリッドのタイムスタンプ発行スケーラビリティの評価実験では，毎秒百万タイムスタンプ発行の阻害要因となる TSU の内部動作由来の遅延とネットワーク遅延の発生頻度を調査した．なお今回の動作実験では，設置可能な TSU 数が少ないため，全ての世代に渡ってのタイムスタンプ発行時間 t_{issue} の直接測定での百万タイムスタンプ発行の可能性を検討は困難である．従って，TSU の内部動作由来の遅延とネットワーク遅延の発生頻度および遅延時間の大きさから各世代間での t_{qr} を推定し，G t_{qr} 秒以内百万タイムスタンプ発行スケーラビリティを検討した．

3.1. 実験概要と実験環境

今回，我々は TSU 群を一般のインターネット環境，NTT B-Flets（日本国内: Mac Mini Intel Core Duo 1.26GHz x 9, SunFire X2100 Opteron 2.0GHz x 2[ただし Tomcat のインスタンスはそれぞれ 11 稼働]），SINET（東工大，阪大:SunFire X2100 Opteron 2.0GHz x2），WIDE（慶大: Mac Mini Intel Core Duo 1.26GHz x1, Mac Mini PowerPC G4 x1）上に設置し，動作実験を行った．Ping コマンドによる round-tirp タイム測定によるネットワーク遅延平均時間測定結果は，日本国内 TSU 間が 2ms 弱から数十 ms であった．

実験においては 1 試行当たり TSU とクライアント間で 1 対 1 では 1,000 タイムスタンプ要求を行った．

3.2. 実験結果と考察

図 3 に大阪大学サイバーメディアセンター内に設置した TSU へ同一ネットワーク上 2 ホップ先のクライアント(MacBook 2.0GHz 2GB RAM 200GB HDD)からタイムスタンプ要求を 1,000 回行った時のタイムスタンプ発行要求応答時間（t_{qr}：単位 ms）を示す．

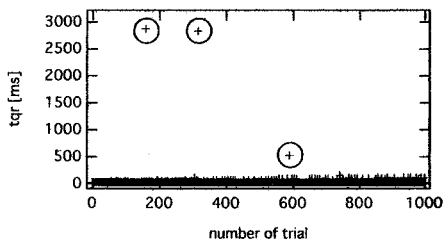

図 3　タイムスタンプ発行要求応答時間

丸で囲んだように 1,000 試行中 2 回 2,500ms を超え，1 回は 500ms を超えていることが分かる．Tomcat のログから 2,500ms 超は何らかのネットワーク遅延、500ms は GC 発生によるものであることが判明した．

同様にして SWoPP 旭川 2007 会場に持ち込んだ同一 MacBook により測定した際のタイムスタンプ発行要求応答時間（t_{qr}：単位 ms）およびログ保存時間（t_{log}：単位 ms）の関係を図 4 に示す．

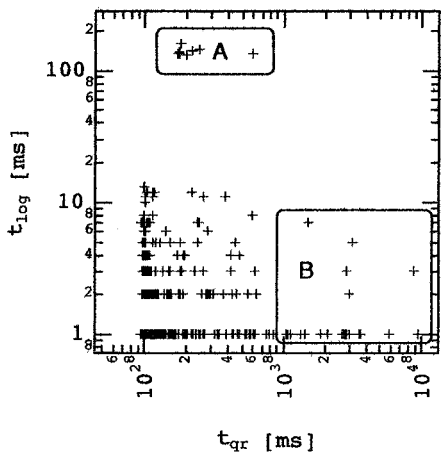

図4　タイムスタンプ発行要求応答時間とログ保存時間の関係

ログの解析よりこの実験ではGCが発生しなかった．図4のA領域ではログのディスクへの保存に関係する遅延が発生しており，B領域では何らかのネットワーク遅延が発生していると推測される．この他の今回の測定結果から1秒間に，200ms以上のネットワーク遅延が発生する確率は 0.00676，GC が発生する確率は 0.000278 となった．これらの遅延が発生する過程はポアソン過程であるとみなせるので，例えば十分な数のTSUがありTSU間ループバック無しでK=32, G=5としたとき毎秒百万タイムスタンプ発行に成功する確率は，1世代当たり 200ms 以内に GC 遅延が起きない確率(0.9668)と1世代当たり 200ms 以内に 200ms 以上のネットワーク遅延が起きない確率（0.9998）の積(0.9663)の5乗から 84.3% と推定される．

4. まとめ

今回，我々は TSU 群を一般のインターネット環境に設置し，動作実験を行い，十分な数の TSU がインターネット上に分散配置されていれば，毎秒百万タイムスタンプが84%という高い確率で可能な条件設定が可能であることを示した．

5. 今後の課題

今回は TSU の世代間でのループバックを行わない条件で動作実験を実施した。ループバックを許すと少ない TSU で大量のタイムスタンプ発行が可能とが，一定時間内に大量のタイムスタンプ発行要求や検証要求がなされた場合，TSU の応答速度が低下することが分かっている．この時、毎秒百万タイムスタンプ発行を実現できるような条件はどのようなものかを検討することが課題である．

謝辞

本研究の一部は国立情報学研究所による「最先端学術情報基盤の構築推進委託事業」（CSI 委託事業）の支援を受けてなされた．
また本実験の一部と担った TSU の設置各機関に感謝する．

参考文献

[1] C. Adams, P. Cain, D. Pinkas and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)," RFC3161, 2001.

[2] A. Takura, S. Ono and S. Naito, "Secure and Trusted Time Stamping Authority, " Proc. of IWS'99, pp.123-128, Feb.1999.

[3] A. Bonnecaze, P. Liardet, A. Gabillon, and K. Blibech, "Secure Time Stamping Schemes: A Distributive Point of View, " Annals of Telecommunications., Vol. 61, no.5-6, pp.662-681, 2006.

[4] Daniela Tulone, "A Scalable and Intrusion-tolerant Digital Time-stamping System, " Proc. 2006 IEEE Int'l Conf. Communications (ICC'06)., Vol.-5, pp.2357-2363, Jun.2006.

[5] 西川武志，松岡聡，"分散時刻認証局グリッドとパラメータ依存性の解析，" 情報処理学会論文誌, Vol.48 No.SIGx ACS19, pp.xx-xx, 2007(印刷中).

[6] 西川武志，松岡聡，"時刻認証グリッドの構築と基礎実験，" 信学技報，Vol.107，No16, pp.13-18, Apr. 2007.

[7] 西川武志，松岡聡，"ハイパフォーマンス分散時刻認証局：毎秒百万タイムスタンプ発行の実現，" 情報処理学会研究報告，2007-HPC-109(HOKKE2007)，pp221-226, Mar. 2007.

[8] 西川武志，松岡聡，"分散時刻認証グリッドのインターネット上での動作実験，" 信学技報, CPSY2007-16, pp.61-64, Aug. 2007.