

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2008-206158

(P2008-206158A)

(43) 公開日 平成20年9月4日(2008.9.4)

(51) Int.Cl. F I テーマコード (参考)
H O 4 L 9/32 (2006.01) H O 4 L 9/00 6 7 5 Z 5 J 1 0 4

審査請求 未請求 請求項の数 22 O L (全 20 頁)

(21) 出願番号 特願2008-37918 (P2008-37918)
(22) 出願日 平成20年2月19日 (2008.2.19)
(31) 優先権主張番号 11/677, 458
(32) 優先日 平成19年2月21日 (2007.2.21)
(33) 優先権主張国 米国 (US)

(71) 出願人 000006747
株式会社リコー
東京都大田区中馬込 1 丁目 3 番 6 号
(74) 代理人 100070150
弁理士 伊東 忠彦
(72) 発明者 ステフェン サビツキー
アメリカ合衆国, カリフォルニア 940
25-7054, メンロ・パーク, サンド
・ヒル・ロード 2882番, スイート
115 リコー イノベーション インク
内

最終頁に続く

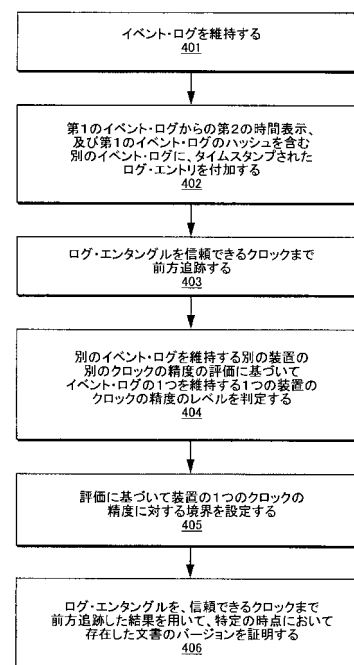
(54) 【発明の名称】 暗号ハッシュによってリンクされたログを用いた信頼すべきタイムスタンプ及び証明できるクロック

(57) 【要約】

【課題】 暗号ハッシュによってリンクされたログを用いて、信頼すべきタイムスタンプ及び証明できるクロックを作成及び／又は使用方法及び装置。

【解決手段】 一実施例では、方法は、第1のチェインハッシュ・ログを維持する工程と、第1のクロックをチェインハッシュ・ログと関連付け、タイムスタンプされた同期化エントリをチェインハッシュ・ログに加えることによって第1のログを第2のログとエンタングルする工程とを備え、同期化エントリは、第2のログと関連付けられた第2の時間表示、及び第1のログにおける1つ又は複数のエントリのハッシュを有する。

【選択図】 図4



【特許請求の範囲】**【請求項 1】**

方法であって、

第 1 のチェインハッシュ・ログを維持する工程と、

第 1 のクロックを前記第 1 のチェインハッシュ・ログと関連付ける工程と、

タイムスタンプされた同期化エントリを前記第 1 のチェインハッシュ・ログに付加することによって、前記第 1 のチェインハッシュ・ログを第 2 のログとエンタングルする工程とを備え、前記同期化エントリは、前記第 2 のログに関連付けられた第 2 の時間表示、及び前記第 1 のチェインハッシュ・ログにおける 1 つ又は複数のエントリのハッシュを有する方法。

10

【請求項 2】

請求項 1 記載の方法であって、前記第 2 のログに関係付けられたハッシュが、前記第 1 のチェインハッシュ・ログにおけるエントリ全てにハッシュを施した結果である方法。

【請求項 3】

請求項 1 記載の方法であって、前記第 2 のログに関連付けられたハッシュが、前記第 1 のチェインハッシュ・ログにおける先行エントリにハッシュを施した結果である方法。

【請求項 4】

請求項 1 記載の方法であって、信頼すべきクロックまでログ・エンタングルを前方追跡する工程を更に備える方法。

【請求項 5】

20

請求項 4 記載の方法であって、信頼すべきクロックまでログ・エンタングルを前方追跡した結果を用いて、識別された時間において文書のバージョンが存在したことを証明する工程を更に備える方法。

【請求項 6】

請求項 4 記載の方法であって、前記第 2 のログに関連したクロックの精度の評価に基づいて前記クロックの精度のレベルを求める工程を更に備える方法。

【請求項 7】

請求項 4 記載の方法であって、観察されるクロック精度、及び前記第 2 のログを維持する少なくとも一装置の評判に基づいてタイムスタンプの信頼度を判定する工程を更に備える方法。

30

【請求項 8】

請求項 4 記載の方法であって、信頼すべきクロックまでログ・エンタングルを前方追跡した結果に基づいて、先行発行タイムスタンプに対する補正によって、前記第 1 のチェインハッシュ・ログにおけるエントリに注釈を入れる工程を更に備える方法。

【請求項 9】

請求項 4 記載の方法であって、前記第 2 のログに関連したクロックの精度の評価に基づいて、前記第 1 のチェインハッシュ・ログに関連したクロックの精度に対する境界を判定する工程を更に備える方法。

【請求項 10】

40

請求項 4 記載の方法であって、クロック・ドリフトの尺度を判定する工程、並びに、前記第 1 のチェインハッシュ・ログ及び前記第 2 のログの同期化によって生じる同期化点を用いて、前記クロックと、前記第 2 のログに関連付けられたクロックとの間のオフセットを推定する工程を更に備える方法。

【請求項 11】

請求項 1 記載の方法であって、前記第 1 のチェインハッシュ・ログの前記クロックの表示、及び前記第 1 のチェインハッシュ・ログのハッシュを含む第 2 のログに、前記第 1 のチェインハッシュ・ログ及び前記第 2 のログにおけるログ・エントリがエンタングル状態にあるように、タイムスタンプされたログ・エントリを付加する工程を更に備える方法。

【請求項 12】

請求項 11 記載の方法であって、第 2 のログそれぞれは、ハッシュ接続されたエントリ

50

・チェーンを備える方法。

【請求項 13】

請求項 1 記載の方法であって、前記第 1 のチェーンハッシュ・ログは、1 つ又は複数のエントリを備え、前記 1 つ又は複数のエントリの少なくとも 1 つは、1 つ又は複数のイベント時間及び文書ハッシュを備える方法。

【請求項 14】

請求項 1 記載の方法であって、前記第 1 のチェーンハッシュ・ログにおける 1 つ又は複数のエントリを補正によって注釈を入れる工程を更に備える方法。

【請求項 15】

請求項 14 記載の方法であって、前記補正は、前記クロックと、第 2 のクロックの時間の表示との間の識別された差異に基づく方法。

10

【請求項 16】

方法であって、

第 1 のハッシュチェーン・ログを第 2 のハッシュチェーン・ログと、前記第 2 のハッシュチェーン・ログに関連付けられたクロックの第 1 の時間表示を含むタイムスタンプされたログ・エントリを前記第 1 のハッシュチェーン・ログに付加することによって同期化させる工程と、

同期化処理の一部として付加された同期化エントリを介して相互接続された前記第 1 のハッシュチェーン・ログ及び前記第 2 のハッシュチェーン・ログを用いて、信頼できるソースまで追跡可能な 1 つ又は複数の文書タイムスタンプを生成する工程とを備える方法。

20

【請求項 17】

請求項 16 記載の方法であって、前記タイムスタンプされたログ・エントリが前記第 2 のハッシュチェーン・ログのハッシュを含む方法。

【請求項 18】

請求項 16 記載の方法であって、前記文書タイムスタンプが、前記第 1 のログ及び前記第 2 のログを用いた証明可能なエラー境界を有する方法。

【請求項 19】

方法であって、

それぞれがクロックを有するハッシュチェーン・ログ対を同期化させるようログ情報を交換する工程と、

30

前記ハッシュチェーン・ログ対における一方のログに関連付けられたクロックの精度に対する境界を、前記ハッシュチェーン・ログ対における他方のログに関連付けられたクロックの精度の評価に基づいて設定する工程とを備える方法。

【請求項 20】

方法であって、

ハッシュチェーン・ログにおけるエントリをタイムスタンプするために用いるクロックを更新する工程と、

前記更新されたクロックに基づいてログの 1 つ又は複数のログ・エントリを補正する工程とを備える方法。

【請求項 21】

40

請求項 20 記載の方法であって、前記更新されたクロックに基づいてログの 1 つ又は複数のログ・エントリを補正する工程が、前記 1 つ又は複数のログ・エントリに補正によって注釈を入れる工程を備える方法。

【請求項 22】

請求項 20 記載の方法であって、信頼すべきと考えられる第 2 の装置に接続する工程を更に備え、前記ログにおけるエントリをタイムスタンプする場合に第 1 の装置によって用いられるクロックを更新する工程が、前記第 2 の装置に関連付けられたクロックを用いて行われる方法。

【発明の詳細な説明】

【発明の詳細な説明】

50

【0001】

(関連出願との相互参照)

本出願は、本発明の、企業の譲受人に譲渡されている「Synchronizing Distributed Work Through Document Logs」と題する、西暦2004年7月9日付け出願の米国特許出願第10/887,998号、「Refining Based on Log Content」と題する、西暦2005年12月29日付け出願の米国特許出願第11/323,543号、「Always On and Updated Operation for Document Logs」と題する、西暦2005年12月29日付け出願の米国特許出願第11/323,542号、「Log Integrity Verification」と題する、西暦2005年12月29日付け出願の米国特許出願第11/323,629号、及び「Paper-Based Document Logging」と題する、西暦2005年8月31日付け出願の米国特許出願第11/514,711号の出願に関する。 10

【技術分野】

【0002】

本発明は、文書処理の分野に関する。特に、本発明は、信頼すべきタイムスタンプ及び証明できるクロックを提供するためにハッシュとリンクされたエンタングル状態のログに関する。

【背景技術】

【0003】

今日、時には、文書の特定のバージョンが、特定の時点以前に存在していた旨を証明する必要がある。従来、これは、タイムスタンプを用いて達成された。文書が存在していた時点と判定しようとする個人は、タイムスタンプ自体を自らの判定のベースにする。しかし、判定が正確であるためには、タイムスタンプは、信頼すべきものでなければならない。 20

【0004】

ログ（イベント・ログ、バージョン・コントロール変更ログ、及びワークフロー文書ログを含む）を組み入れた何れのシステムにおいても、しかじかのログ・エントリが特定の時点で作成されたことを証明する（例えば、文書の特定のバージョンが、だれそれによって、遅くとも現時点以前に見られており、かつ、その時点よりも前に見られていないことを証明する）ことが時には重要である。すなわち、偽造不能のタイムスタンプが必要である。 30

【0005】

この課題に対処する通常のやり方は、偽造不能のタイムスタンプを提供する集中化された「デジタル公証」サービスを備えることである。これはその場合、文書又はログ・エントリに施すことが可能である。これは、エントリをハッシングし、ハッシュを公証サービスに送出し、特定の時点において公証サービスによってエントリのハッシュが見られたことを証明するために後に用いることが可能な証明できる受領書を受け取ることによって行われる。随時、公証サービスが、記録の新聞などの印刷媒体においてその現在の「累計ハッシュ」を発行する。あるいは、信頼できるソースは、デジタル署名された証明書を発行する認定されたタンパリング防止クロック装置を供給することができる。何れの集中化手法においても、サービス又は装置が、特定のハッシュを備えた文書が特定の時点で存在したことを証明するために、遡って用いることが可能な、ハッシュ接続ログを維持することが肝要である。 40

【0006】

ピアツーピア・システム、「git」やその他の分散化システムのような分散化バージョン・コントロール・システムによって、集中化手法が不可能になる。このことの理由はいくつかある。別個の装置はそれ自身のクロック（これは、不正確であり得る）を有し、それ自身のログを保持する。上記装置は、長期間にわたってネットワークと接触していないことがあり得るものであり、多くの場合、オフにされ得る。

【0007】

公開鍵暗号が、過去において、解決策として用いられている。前述の場合、公開鍵を用 50

いて認証している。これは、特定のサーバをユーザが信頼することを可能にする。しかし、どのサーバを信頼するかについての決定は、事前に行わなければならない。

【0008】

NTP（ネットワーク時間プロトコル）を用いて、ネットワーク化された装置上のクロックを同期化させる。NTPは、サーバとクライアントとの間の同期化メッセージを伴う。NTPバージョン4は、タイムスタンプの出所を、信頼できるソースまで前方追跡して、タイムスタンプの信頼できる出所を確かめることを可能にする、オートキー（Autokey）公開鍵認証プロトコルと呼ばれている分散セキュリティ・アーキテクチャを含む。しかし、これは、NTPクライアントが現在の時間を正確に判定することしか可能にしない。このこと自体は、NTPによって生成されたタイムスタンプの精度を後日に確認することを可能にしない。

10

【発明の開示】

【発明が解決しようとする課題】

【0009】

暗号ハッシュによってリンクされたログを用いて、信頼すべきタイムスタンプ及び証明できるクロックを作成及び／又は使用方法及び装置。

【課題を解決するための手段】

【0010】

一実施例では、方法は、第1のチェインハッシュ・ログを維持する工程と、第1のクロックをチェインハッシュ・ログと関連付け、タイムスタンプされた同期化エントリをチェインハッシュ・ログに加えることによって第1のログを第2のログとエンタングルする工程とを備え、同期化エントリは、第2のログと関連付けられた第2の時間表示、及び第1のログにおける1つ又は複数のエントリのハッシュを有する。

20

【発明を実施するための最良の形態】

【0011】

本発明は、以下に表す詳細な説明から、かつ、本発明の種々の実施例の添付図面から、より徹底的に分かるであろう。しかし、それらは、特定の実施例に本発明を限定するものと解されるべきでなく、説明及び理解の目的に過ぎない。

【実施例】

【0012】

暗号ハッシュによってリンクされたログを用いて信頼すべきタイムスタンプ及び証明できるクロックを提供する方法及び装置について説明する。一実施例では、装置は、ハッシュによって接続された時間ログを維持し、これを用いてその他のログ・エントリをタイムスタンプ（エンタングル）する。2つの装置は、ログ情報を交換するために接触する場合、それらの時間ログを同期化させることが可能である。すなわち、各装置は、別のログの時間、及びログ・ハッシュを含む、タイムスタンプされたログ・エントリを作成する。これによって、「信頼の輪」が形成される。よって、累計ハッシュを交換することによって、ログがエンタングルされ、それによって、因果配列を表す、ログ・エントリ（例えば、バージョン、イベント等）に対する部分配列がもたらされる。一実施例では、エンタングルされたログは、例えば、特定のログ・エントリが作成された時点において特定の装置によって見られた文書又はイベントを証明するために用いられる。

30

40

【0013】

一実施例では、各装置は、別の装置の精度のその評価に応じて、そのクロックの精度に対する境界を設定し、これは、今度は、それらの介在タイムスタンプの精度に境界を設定する。一部の装置は、クロックをGPS衛星システム、NTPティア1時間サーバ、又は特定の他の信頼すべきソースにクロックを同期化させていることがあり得る。一実施例では、前述の信頼すべきソースに基づいて、タイムスタンプを備えたログ・エントリから前方追跡することによって、絶対的な境界が、ログ記録された何れかのイベントの時間に対して設定される。

【0014】

50

装置は、そのクロックを失うのに十分な期間、中断又はダウンしていることがあり得る。この場合にも、境界は計算可能であり、装置の過去の履歴を用いて、そのタイムスタンプにおくことが可能な信頼度を更に数量化する。

【0015】

以下の説明では、数多くの詳細を記載して、本発明のより徹底的な説明を提供する。しかし、前述の具体的な詳細なしで本発明を実施することができることは当業者に明らかになる。一方、周知の構造及び装置は、本発明を分かりにくくしないように詳細にではなく、ブロック図で示す。

【0016】

以下に続く詳細な説明の特定部分は、コンピュータ・メモリ内のデータ・ビットに対する演算のアルゴリズム及び記号表現によって提示する。前述のアルゴリズム記述及び表現は、成果の本質を他の当業者に効果的に伝えるためにデータ処理技術分野における当業者によって用いられる手段である。アルゴリズムは本明細書では、かつ、一般には、所望の結果につながる、自己矛盾のない工程系列であるとして考え出されている。工程は、物理的数量の物理的操作を必要とするものである。通常、必然的ではないが、前述の数量は、記憶し、転送し、合成し、比較し、他のやり方で操作することができる電気信号又は磁気信号の形式を呈する。時には、前述の信号をビット、値、エレメント、シンボル、文字、項、数字等として表すことが、特に、一般的に使用されているという理由で好都合であることが明らかになっている。

【0017】

しかし、前述及び同様な語は、適切な物理的数量と関連付けられるものとし、前述の数量に付される好都合なラベルに過ぎない。以下の記載から明らかであるように、別途明記しない限り、本明細書を通して、「処理」あるいは「計算」あるいは「算出」あるいは「判定」あるいは「表示」等などの語を利用した記載は、コンピュータ・システムのメモリ又はレジスタやその他の前述の情報記憶装置、情報伝送装置、若しくは情報表示装置内の物理的数量として同様に表される他のデータに、コンピュータ・システムのレジスタ及びメモリ内の物理的（電子的）数量として表されるデータを操作し、変換する、コンピュータ・システムや同様な電子計算装置の動作及び処理を表す。

【0018】

本発明は、本明細書及び特許請求の範囲記載の動作を行う装置にも関する。この装置は、必要な目的で特別に構成することができるか、又は、コンピュータに格納されたコンピュータ・プログラムによって選択的に起動又は再構成される汎用コンピュータを備え得る。前述のコンピュータ・プログラムは、コンピュータ読み取り可能な記憶媒体（電子命令を格納するのに適しており、それぞれがコンピュータ・システム・バスに結合されている何れかのタイプのディスク（フロッピー（登録商標）・ディスク、光ディスク、CD-ROM及び光磁気ディスクを含む）、リード・オンリ・メモリ（ROM）、ランダム・アクセス・メモリ（RAM）、EPROM、EEPROM、磁気カード若しくは光カード、又は何れかのタイプの媒体（これらに限定されない）など）に格納することができる。

【0019】

本明細書及び特許請求の範囲記載のアルゴリズム及びディスプレイは、何れの特定のコンピュータやその他の装置にも固有に関係していない。種々の汎用システムを本明細書及び特許請求の範囲記載の教示によってプログラムとともに用いることができるか、又は、要求される方法工程を実行するために、より専用の装置を構成することが好都合であることが明らかになり得る。前述の種々のシステムに要求される構造は以下の記載から明らかになる。更に、本発明は、何れの特定のプログラミング言語を参照しても説明されていない。各種のプログラミング言語を用いて、本明細書及び特許請求の範囲記載の本発明の教示を実現することができる。

【0020】

マシン読み取り可能な媒体は、マシン（例えば、コンピュータ）によって読み取り可能な形式で情報を記憶するか又は伝送する何れかの機構を含む。例えば、マシン読み取り可

10

20

30

40

50

能な媒体は、リード・オンリ・メモリ（「ROM」）、ランダム・アクセス・メモリ（「RAM」）、磁気ディスク記憶媒体、光記憶媒体、フラッシュ・メモリ装置、電氣的、光學的、音響的やその他の形式の伝搬信号（例えば、搬送波、赤外線信号、デジタル信号等）等を含む。

【0021】

ログの概要

本明細書及び特許請求の範囲の目的では、ログは、先行するエントリ系列全体に各エントリが依存する、タイムスタンプされた「エントリ」系列である。これは、先行エントリのハッシュを各エントリに含ませることによって達成される。エントリやそのハッシュは、そのハッシュを変更することなく、かつ、帰納法により、後続エントリ全てのハッシュを変更することなく変更することが可能でないようにチェーンを形成する。すなわち、各ログは、「累計ハッシュ」（一実施例では、最も直近のエントリのハッシュである）を有する。最も直近のエントリは累計ハッシュの先行値を含むので、現在の累計ハッシュが、先行エントリ全ての内容及び配列によって求められることを帰納法によって証明することが可能である（エントリのハッシュを用いて、エントリを一意に識別することができる。）

10

一実施例では、ログ内のエントリは文書に関する。例えば、ログは、種々のアプリケーションに適切であり得るデジタル・ファイル、文書のバージョン履歴、文書のアクセスや、何れかの他のタイプの情報に施される処理の系列を表し得る。本明細書及び特許請求の範囲の目的で、「文書」の語は、テキスト・ファイル、オーディオ・ファイル、GRS情報、電話通話ログ等を含み得る。処理は、文書が印刷されたか、若しくは視られた時点、又は他の情報が文書に追加された時点の表示を含み得る。

20

【0022】

一実施例では、エントリ及び文書をブラブ（データベース技術では「バイナリ・ラージ・オブジェクト」）として知られている）として格納することが可能である。各ブラブは、その暗号ハッシュによって識別される不変のデータ・オブジェクトである。

【0023】

ログに対して行うことができる処理はいくつか存在している。限定されないが、これらには、例えば、以下を含む。

【0024】

30

1) ログへのエントリの追加

2) 相互参照エントリを用いることによる2つのログのエンタングル（ログが何れも時間ログ（すなわち、時間スタンプ・エントリを含むログ）の場合、これは、「同期化イベント」と呼ばれる

3) エントリに注釈を入れる（例えば、注釈が入れられるエントリのハッシュIDを表す新たなエントリを作成する）

4) より正確であるとアサート又は判定される改訂タイムスタンプを示すための一定範囲のエントリの注釈

前述の処理を以下に更に詳細に説明する。

【0025】

40

ログ形式及びイベント

一実施例では、ログ・エントリの形式は、実現形態によって特有である。一実施例では、形式は、XMLエンティティである。別の実施例では、形式は、MIME「メッセージ」である。一実施例では、ログ・イベントは以下の形式を有する。

【0026】

LogEvent (eventType, bodyText, 属性)

これは、特定のタイプ及び本文（自由形式）、並びに更なる属性（属性・値対のリスト）によってログ・エントリを作成する。

【0027】

一実施例では、イベントは、名前・値対（XML属性又は「名前：値」ヘッダ項目）の形

50

式における以下の情報を含む。

【0028】

1) 先行エントリ 先行ログ・エントリのハッシュ（及び場合によってはシーケンス番号）。別の実施例では、常に現在のエントリのシーケンス番号より1少ない所では、シーケンス番号は含まれない。

【0029】

2) シーケンス番号 現在のエントリのシーケンス番号。現在時間データ（すなわち、localTime、オフセット、driftRate、syncTime、syncEvent）は、別個の属性に含まれ得るか、単一のタイムスタンプ列として含まれ得るか、又は両方の特定の組み合わせで含まれ得る。

10

【0030】

3) イベント・タイプ logEvent処理において表すようなイベント・タイプ

4) 本文 「本文」（自由形式列）。ログ・エントリがMIMEメッセージ形式の場合、Content-Sizeヘッダが存在しており、Content-Typeヘッダは任意である。

【0031】

5) 参照 タイムスタンプされている文書、又は参照されている先行ログ・イベントは、本文においてではなく、属性において参照される。

【0032】

用いることができるイベント・タイプがいくつか存在している。一実施例では、これらは以下の通りである。

20

【0033】

1) 記録 （タイムスタンプされた）コメントを本文中に記録する。

【0034】

2) タイムスタンプ 文書をタイムスタンプする。一実施例では、文書はそのハッシュによって規定される。

ブラブは、利用可能であり、文書ログを有する場合、注釈が入れられる。

【0035】

3) シンク 別の時間ログ・サーバ又はNTPサーバに同期化させる。どのサーバが要求を行っているか、及び、サーバのクロックが調節されているか否かに応じてこのバージョンがいくつか存在し得る。このイベントは、現在の（旧い）時間、及び調節された新しい時間をともに記録する。

30

【0036】

4) アンシンク 同期化の喪失（例えば、電力の喪失）、及び適切な時間サーバを見つけることができないことを示す。

【0037】

5) 正確 正確なオフセットが分かると、一定範囲におけるタイムスタンプに注釈を入れる。一実施例では、上記範囲は通常、アンシンクとシンクとの間である。このイベントでは、タイムスタンプされた文書ブラブは、アクセス可能であり、文書ログを有する場合、注釈が入れられる。

【0038】

6) 要求 別のサーバに対して行われる要求を文書化する。

40

【0039】

ログへのイベントの追加

図1は、イベントをログ記録するための処理の一実施例のフロー図である。処理は、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータ・システム上若しくは専用マシン上で実行されるものなど）、又は両方の組み合わせを備え得る処理ロジックによって実行される。

【0040】

処理を例証する目的で、ファイル名HEADを有するログを用いる。一実施例では、ログが、時間ログ・サーバによって維持される。一実施例では、時間ログ・サーバは、ウェブ・

50

サーバを介してアクセスされるCGIである。前述の場合、クライアント及びピアはGET要求を用いて、ハッシュによって、又は時間間隔によってログ・エントリを取り出し、POST要求を用いてエントリを作成する。

【0041】

図1を参照すれば、処理は、処理ロジック（例えば、プロセッサ612）が、ファイルHEADに対する書き込みロックを取得して（処理ブロック101）、ログにイベントが追加されている間にファイルHEADがアクセスされることを阻止することによって始まる。これは任意である。一実施例では、HEADは、最も直近のエントリのファイル・パス（シーケンス番号及びハッシュを含む）を含む。

【0042】

特に、ロックは、2つのPOST要求の処理が重なることを阻止する。このことは、単一スレッド・ウェブ・サーバにおいて、又は、資源に対する単一スレッド・アクセスを保証することが可能なものにおいて行うことができる。一方、アパッチのようなマルチスレッド・サーバは、ファイル・ロック、又は別個の単一スレッド・コサーバを必要とする。一実施例では、最も直近のログ・エントリを指し示す（例えば、最も直近のエントリを含む）ファイルに対する排他的なロックを得るためのシステム・コール。それ自身の内部ロックを行うジャバ・ミニサーバ、又はパールCGIを用いることができる。

【0043】

次に、処理ロジック（例えば、プロセッサ612）は、ファイルENTRYが存在するか否かを検査する（処理ブロック102）。ファイルENTRYがファイルHEAD内に存在しない場合、処理ロジック（例えば、プロセッサ612）は処理ブロック106に移行する。ファイルENTRYがファイルHEAD内に存在する場合、処理ロジック（例えば、プロセッサ612）はそのシーケンス番号を確認し、処理ブロック103に移行することによってファイルHEADをハッシュ・マッチングする。処理ブロック103では、処理ロジック（例えば、プロセッサ612）は、エントリのハッシュ（hash（ENTRY）として表す）が、ファイルHEAD内にあるか否かを検査する。いいえの場合、処理ロジック（例えば、プロセッサ612）は、ログにイベントを追加しようとする者にエラーを通知し、処理ブロック105に移行する。エントリのハッシュ（hash（ENTRY））がファイルHEAD内にある場合、処理ロジック（例えば、プロセッサ612）は処理ブロック105に移行する。処理ブロック105では、ENTRYがファイルHEADから削除され、処理ロジック（例えば、プロセッサ612）は処理ブロック106に移行する。

【0044】

処理ブロック106では、処理ロジック（例えば、プロセッサ612）は、ログ内の次のエントリを指し示すようシーケンス番号を増やす。その後、処理ロジック（例えば、プロセッサ612）は、（シーケンス番号を含む）新たなエントリを含む新たなENTRYファイルを構築し（処理ブロック107）、ENTRYにハッシュを施すことによってENTRYのハッシュを計算する（処理ブロック108）。エントリのハッシュを計算した後、処理ロジック（例えば、プロセッサ612）はその新たなパスを計算し、ENTRYをブラブ記憶装置に移動させ（処理ブロック109）、hash（ENTRY）及びシーケンス番号をファイルHEADに記憶する（処理ブロック110）。その後、処理ロジック（例えば、プロセッサ612）は、ファイルHEADに対するロックを解除し（処理ブロック111）、処理は終了する。

【0045】

ログの再同期化

一実施例では、ログは、別のクロック又は時間を用いてログ内のエントリを同期化させる同期化イベントを経る。新たな時間が別のログから導き出され、現在のログの時間が、別のログの同期化に用いられている場合、相互参照エントリによって、2つのログがエンタングル状態になる。

【0046】

図2は、別のソース（例えば、別のログ、信頼すべきソース等）のクロック又は時間を用いてログを再同期化させる処理の一実施例のフロー図である。処理は、ハードウェア（

10

20

30

40

50

回路、専用ロジック等)、ソフトウェア(汎用コンピュータ・システム上若しくは専用マシン上で実行されるものなど)、又は両方の組み合わせを備え得る処理ロジックによって実行される。

【0047】

一実施例では、以下の時間データ・パラメータを処理において用いる。

【0048】

1) long clockTime(c, clock)。ログを維持するハードウェア・クロック又は装置によって現在示されている時間

2) long offset(d, delta)。最後の同期化の際のログの「実」時間と局所時間との間の差を表す。一実施例では、dは、同期化処理の一部としてハードウェア・クロックが「実」時間にリセットされた場合にd=0であるように設定される。

10

【0049】

3) double driftRate(r, rate)。ログによって用いられる局所時間が実時間から逸脱するレート。これは、時間に対するクロック周波数の一次導関数である。

【0050】

4) long baseTime(c0)。最も直近の同期化イベントの際のclockTimeを表す。

【0051】

5) long fixup(f)。再同期化イベントに施す対象のフィックスアップを表す。

【0052】

図2を参照すれば、処理は、処理ロジック(例えば、プロセッサ612)が、新たな時間t'を得ることによって始まる(処理ブロック201)。新たな時間は、信頼すべきソース(例えば、別のログ、実時間クロック源等)で得ることができる。

20

【0053】

次に、処理ロジックは、クロックc、オフセットd、及び一定期間にわたるドリフト・レートrに基づいて現在時間tを設定する。時間データを用いて、一実施例では、システム(例えば、プロセッサ612)は、現在の「実」時間(t)を以下によって求める。

【0054】

$realTime = clockTime + offset + driftRate * (localTime - syncTime)$

これは、

$t = c + d + r(c - c0)$ によって表すことができる(処理ブロック202)。

30

【0055】

次いで、処理ロジック(例えば、プロセッサ612)は、以下の式によって、ドリフト・レートr及び長いフィックスアップfを設定する。

【0056】

$r = (t' - c - t) / (c - c0)$

$f = t' - t$

(処理ブロック203)

この時点で、一実施例では、ハードウェア・クロックを、正確な値に再設定することが可能である。次に、処理ロジック(例えば、プロセッサ612)は、ハードウェア・クロックを再設定するか否かを検査する(処理ブロック204)。

40

【0057】

別の実施例では、実効時間を正確にするようオフセットを調節することが可能である。一実施例では、前述の実施例のうちのどれを用いるかは、多くの要因(例えば、ハードウェア・クロックに対する変更によって影響を及ぼされ得る他のアプリケーションの要件を含む)に依存する、システム構成の決定事項である。

【0058】

一実施例では、ハードウェア・クロックは変更されない。あるいは、ハードウェア・クロックは、新たな時間t1に再設定される。

【0059】

ハードウェア・クロックを再設定する場合、処理ロジック(例えば、プロセッサ612

50

）は、処理ブロック205に移行する。処理ブロック205では、クロックcが、現在時間tに等しくなるように設定され、オフセットdが、ゼロに等しくなるように設定され、c0が、現在時間tに等しくなるように設定され、次いで、処理は、処理ブロック207に移行する。ハードウェア・クロックが再設定されていない場合、処理ロジック（例えば、プロセッサ612）は、オフセットdが $t' - c$ に等しくなるように設定し、c0がcに等しくなるように設定し（処理ブロック206）、次いで、処理は処理ブロック207に移行する。よって、再同期化を行う場合、局所クロック時間cにおける正確な時間t'を前提として、新たなパラメータc0、c、d及びr'が計算され、これらは、c0、d及びrの新たな値になる。

【0060】

10

処理ブロック207では、処理ロジック（例えば、プロセッサ612）は、ログ・エントリを作成する。

【0061】

その後、処理は完了する。

【0062】

同期化は、非同期化状態で実行される期間後に行うこともできる。この場合、行うことが可能な最良のことは、 $r' = r$ になるように設定することである。再計算されることが可能になる、c0の正確な値が存在しないからである。パラメータc0は、非同期化状態での実行の開始の際のクロック時間である。

【0063】

20

先行タイムスタンプt（元のパラメータc、d、c0及びrを備える）を補正するために、より正確であると思われる新たなレートr'、及び現在のフィックスアップfを前提とすれば、古いレート補正が取り去られ、新たなものが代入される。これは、以下を用いて表される。

【0064】

$$t' = t + f - r(c - c0) + r'(c - c0)$$

一実施例において、システム・クロックが、NTPを用いて同期化状態に維持され、平均して $r=0$ であるように位相ロック・ループにおいて常に調節された場合、再同期化は、あるとしても、非同期化状態で実行される期間後以外、まず必要でなくなり、補正は一定のフィックスアップを施すことを含む。

30

【0065】

ログ・エントリに注釈を入れる

一実施例では、ログ・エントリには、改訂された時間によって注釈を入れることができる。これは、より信頼すべきクロック（すなわち、より信頼すべきソースからのクロック）が利用可能になった場合に行うことができる。前述の場合、ログにおける1つ又は複数のエントリを、より信頼すべきクロックに再同期化させることができる。

【0066】

図3は、1つ又は複数のログ・エントリに注釈を入れるための処理の一実施例のフロー図である。処理は、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータ・システム上若しくは専用マシン上で実行されるものなど）、又は両方の組み合わせを備え得る処理ロジックによって実行される。

40

【0067】

図3を参照すれば、処理は、処理ロジック（例えば、プロセッサ612）が、先行ログ・エントリを得ることによって始まる（処理ブロック301）。先行ログ・エントリを得た後、処理ロジック（例えば、プロセッサ612）は、ログ内のこのエントリの場所が、再同期化点にあるか否かをチェックする（処理ブロック302）。はいの場合、再同期化処理は完了している。

【0068】

ログ内のこのエントリの場所が、再同期化点にない場合、処理ロジック（例えば、プロセッサ612）は、エントリの時間(t)、ドリフト・レート(r)、クロック(c)、及び開始

50

時間(c0)をエントリから取得し（処理ブロック303）、

$$t' = t + f - r(c - c0) + r'(c - c0)$$

によって求められる新たな時間t'によってエントリに注釈を入れる（処理ブロック304）。

【0069】

次いで、処理ロジック（例えば、プロセッサ612）は、文書ブラブが存在しているか否かを判定する（処理ブロック305）。はいの場合、処理ロジック（例えば、プロセッサ612）は、改訂時間t'を示すよう文書ブラブに注釈を入れ（処理ブロック306）、処理ブロック307に移行する。文書ブラブが存在しない場合、処理ロジック（例えば、プロセッサ612）は、処理ブロック307に直接移行する。処理ブロック307では、処理ロジック（例えば、プロセッサ612）は、補正が行われた旨を示す新たなログ・エントリをログに対して行う。その後、処理ロジック（例えば、プロセッサ612）は、処理ブロック301に移行する。処理ブロック301では、処理が繰り返される。

10

【0070】

一実施例では、元の文書、及びログ・エントリ・ブラブを維持するやり方で注釈が入れられる。それらのハッシュ（及び、よって、それらの識別情報）が、時間ログのインテグリティに不可欠であるからである。

【0071】

一実施例では、注釈のハッシュ接続ログは、注釈が入れられている、元の文書、又はログ・エントリのハッシュに「.log」を付加することによって命名されるディレクトリ内に保持される。

20

【0072】

信頼すべきタイムスタンプ

汎用タイムスタンプ

一実施例では、時間ログ・サーバは、信頼すべきタイムスタンプを生成する。一実施例では、時間ログ・サーバは、時間ログ・サーバにタイムスタンプ送出される対象のアイテム（例えば、文書ブラブ、ディレクトリツリー・ブラブや、ハッシュ・ベースの改訂識別子）のハッシュを含む要求を受けることによって、信頼できるタイムスタンプを生成するようにプロンプトされ得る。要求は、要求側のエンティティ（例えば、要求側のプログラム）から来る。時間ログ・サーバは、ハッシュ及び時間を含むその時間ログにエントリを作成し、エントリのハッシュを、要求側のエンティティに返す。

30

【0073】

一実施例では、時間ログ・サーバは最低限、テキスト・ブロックを受け入れ、これをタイムスタンプ、及び先行エントリのハッシュとともにログに記録する。一実施例では、時間ログ・サーバは、ハードウェア・クロックによる現在時間と、最も直近のエントリとの間のスキューを検出し、かつ、再同期化のための特定の処理を行うことができる。

【0074】

一実施例では、時間も返される。ログ・エントリから、そのハッシュを用いて取り出すことが可能であるので、このことは必要でないが、要求側のエンティティの労力を節減するために、いずれにせよ、時間を返すことが通常、好都合になる。結果として生じるタイムスタンプを利用する1つのやり方は、タイムスタンプされる対象に関連したログ・エントリやその他の特定のメタデータを作成することである。このことは、時間ログ及びアプリケーションが、同じ形式でログを保持している場合に、特にうまく実施される。

40

【0075】

エンタングル状態のログ

一実施例では、信頼すべきタイムスタンプは、ハッシュチェイン・ログを用いてイベント時間及び文書（ブラブ）ハッシュをログ記録し、そのログ・エントリを、他の装置のものとエンタングルする装置（例えば、時間ログ・サーバ）によって提供される。そのログと、他のクロックのログが更にエンタングルされるほど、クロックにタンパリングすること、又はログ・エントリを作成することが難しくなる。エンタングル状態のログは、PGP

50

のような暗号システムにおいて用いられる信頼の輪と同様な「信頼の輪」を設定する。

【0076】

図1をもう一度参照すれば、付加する対象のエントリは、タイムスタンプされた1つ又は複数のエントリを既に含むイベント・ログに付加する対象の、タイムスタンプされたログ・エントリであり得る。前述の通り、一実施例では、タイムスタンプされたエントリは、同期化処理の一部として付加される（例えば、同期化エントリ）。前述の同期化エントリを付加することによって、イベント・ログは、相互接続されるか、又はエンタングルされる。

【0077】

一実施例では、文書タイムスタンプは、同期化の一部として付加された同期化エントリによって相互接続された相互接続イベント・ログを用いて、信頼できる機関まで追跡可能である。

【0078】

ログ・エンタングルの追跡による境界設定

ログ・エンタングルを、周知の時間標準（例えば、暗号認証プロトコル（例えば、NTPなど）によって設定される）との既知の関係によって、信頼できるクロックまで追跡することによって、クロックに対していくつかの種類の境界を設定することが可能になる。これらには、絶対境界及び近似境界が含まれる。絶対境界によって、エンタングルに関連したタイムスタンプに基づいて、考えられるクロック・スキューの絶対範囲を、エンタングルの時点で求めることが可能である。近似境界によって、エンタングル間で、クロックのオフセット及びドリフト・レートを計算することが可能である。クロックがタンパリングされていない限り保証することが可能なタイムスタンプを、これを用いて計算することが可能である。これは、フロー図に更に詳細に表している。

【0079】

図4は、ログを処理する方法の別の実施例のフロー図である。処理は、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータ・システム上若しくは専用マシン上で実行されるものなど）、又は両方の組み合わせを備え得る処理ロジックによって実行される。

【0080】

図4を参照すれば、処理は、イベント・ログを維持することによって始まる（処理ブロック401）。特定時点において、処理ロジック（例えば、プロセッサ612）は、別のイベント・ログからの第1の時間表示（例えば、クロック）、及び第2のイベント・ログのハッシュを含む第1のイベント・ログに、タイムスタンプされたログ・エントリを付加することによって、別のイベント・ログとイベント・ログを同期化させる。これは、別のログの累計ハッシュを表す。

【0081】

2つ以上のログを併せてエンタングルするために、処理ロジック（例えば、プロセッサ612）は、第1のイベント・ログに関連したクロックの時間表示、及び第1のイベント・ログのハッシュを含む別のイベント・ログに、タイムスタンプされたログ・エントリも付加する（処理ブロック402）。この処理を行った後、イベント・ログそれぞれは、ハッシュ接続されたエントリ・チェーンを備える。ハッシュ接続されたエントリ・チェーンは、1つ又は複数のイベント時間及び文書ハッシュを備える。

【0082】

次に、処理ロジック（例えば、プロセッサ612）は、ログ・エンタングルを信頼できるクロックまで前方追跡する（処理ブロック403）。一実施例では、信頼できるクロックは、周知の時間標準との既知の関係を有する。一実施例では、信頼できるクロックにログ・エンタングルを前方追跡することは、観察されたクロック精度、及び別のイベント・ログを維持する装置の評判に基づいてタイムスタンプの信頼度を判定することを含む。

【0083】

ログ・エンタングルを、信頼できるクロックまで前方追跡した後、処理ロジック（例え

10

20

30

40

50

ば、プロセッサ 612) は、別のイベント・ログを維持する別の装置の別のクロックの精度の評価に基づいてイベント・ログの 1 つを維持する 1 つの装置のクロックの精度のレベルを判定し (処理ブロック 404)、その評価に基づいて装置の 1 つのクロックの精度に対する境界を設定する (処理ブロック 405)。クロックの精度はいくつかのやり方で評価することができる。例えば、時間ログが、時点 t_1 (その時点は、 $\pm a_1$ 以内で正確であることが知られている) において別のログと同期化されていると仮定する。更に、上記ログが、 $\pm a_2$ の精度の時点 t_2 において同期化されていると仮定する。その場合、時間 $t_1 \leq t < t_2$ を有するログにおける各エントリが、間隔 $(t_1 - a_1) < t < (t_2 + a_2)$ において生じたはずであることを証明することが可能である。多くの場合、精度の前述の証明が十分になる (例えば、支払いが行われた日を特許の優先日として証明するため)。クロックがタンパリングされていないことを更に仮定した場合、時間 t に対する境界を $(t - a_1) < t < (t + a_2)$ に狭くすることが可能である。一実施例では、これは、相互接続されたイベント・ログを用いて、証明可能なエラー境界を備えた文書タイムスタンプを生成することを可能にする。

10

【0084】

一実施例では、クロックの精度に対して境界を設定することは、クロック・ドリフトの尺度を判定し、ログにおけるエントリにタイムスタンプするために用いられるクロックと、イベント・ログの同期化によって生じる同期化点を用いた (イベント・ログの同期化によって生じる同期化点に同期化された) 別の、より高精度のクロックとの間のオフセットを推定することを含む。

20

【0085】

任意的には、処理ロジック (例えば、プロセッサ 612) は、ログ・エンタングルを、信頼できるクロックまで前方追跡した結果を用いて、識別された特定の時点において存在した文書のバージョンを証明することもできる (処理ブロック 406)。一実施例では、特定の時点は、特定の日の時点、又は特定の日より前の時点を含む。

【0086】

一実施例では、より信頼すべきクロックとのエンタングルによって、イベント・ログを維持する装置のクロックが再同期化された後、ログにおけるエントリに、より正確な時間の注釈が入れられる。これが行われると、注釈を入れたエントリ範囲を示す新たなエントリがログに入れられる。

30

【0087】

図5は、ログを処理する方法の別の実施例のフロー図である。処理は、ハードウェア (回路、専用ロジック等)、ソフトウェア (汎用コンピュータ・システム上若しくは専用マシン上で実行されるものなど)、又は両方の組み合わせを備え得る処理ロジックによって実行される。

【0088】

図5を参照すれば、処理は、イベント・ログを維持することによって始まる (処理ブロック 501)。特定時点において、処理ロジック (例えば、プロセッサ 612) は、別のイベント・ログからの第 1 の時間表示 (例えば、クロック)、及び第 2 のイベント・ログのハッシュを含む第 1 のイベント・ログに、タイムスタンプされたログ・エントリを付加することによって、別のイベント・ログと、イベント・ログを同期化させる。

40

【0089】

2 つ以上のログを併せてエンタングルするために、処理ロジック (例えば、プロセッサ 612) は、第 1 のイベント・ログからの第 2 の時間表示、及び第 1 のイベント・ログのハッシュを含む別のイベント・ログに、タイムスタンプされたログ・エントリも付加する (処理ブロック 502)。2 つのログの 1 つのみが相互参照を含む場合にのみ、ログがエンタングルされる。エントリが第 2 のログに付加された場合、これは、第 1 のログを見つけるための好都合なやり方であり、実際に、完全に任意である。この処理を行った後、イベント・ログそれぞれは、ハッシュ接続されたエントリ・チェーンを備える。ハッシュ接続されたエントリ・チェーンは、1 つ又は複数のイベント時間及び文書ハッシュを備える

50

。

【0090】

次に、処理ロジック（例えば、プロセッサ612）は、ログ・エンタングルを、信頼すべきクロックまで前方追跡する（処理ブロック503）。一実施例では、信頼すべきクロックは、周知の時間標準との既知の関係を有する。

【0091】

ログ・エンタングルを、信頼すべきクロックまで前方追跡した後、処理ロジック（例えば、プロセッサ612）は、ログ・エンタングルを、信頼すべきクロックまで前方追跡した結果に基づいて、一イベント・ログにおける先行発行タイムスタンプに対する、信頼すべき補正を生成する（処理ブロック504）。これは、エントリのエラー及び／又はその補正を示す注釈を各エントリに付加することができる。

10

【0092】

タイムスタンプされたサーバ・ログ・ファイル

ウェブ・サーバやその他のプログラムは、タイムスタンプを含むログ・ファイルを生成する。ログ記録された特定のイベントが生起したおおよその時点を証明することができるが、場合によっては、有意義であり得る。一実施例では、この情報を用いて、ログ・ファイルにタンパリングが存在していないことを証明する。ウェブ・サーバの場合、サーバの要求ログへの読み出しアクセスを用いることによって、サーバを修正することなく、これを行うことができる。

【0093】

20

一実施例では、以下の手順を行う。まず、時間ログ・サーバは、その現在の累計ハッシュを含むユニバーサル・リソース識別子（URI）（例えば、URL）を用いた要求をウェブ・サーバに対して行うと同時に、同じハッシュをそれ自身のログに記録する。一実施例では、これは、クエリ列内にハッシュを備えたHTTP GET要求を用いて行う。第2に、時間ログ・サーバは次いで、ウェブ・サーバのログを読み出し、それ自身の要求（累計ハッシュの第1のインスタンスを含むので認識可能である）を含むハッシュを計算する。第3に、時間ログ・サーバは次いでそれ自身のログにおいて、サーバのログの（新たな）現在の累計ハッシュ及び新たに計算されたハッシュを含む第2の要求を行い、記録する。

【0094】

30

この処理は、時間ログ・サーバ又は外部クライアント（ウェブ・サーバでない）によって起動される。外部クライアントは、基本的に時間ログ・サーバのクライアントであり、何れのやり方においても修正しなくてよい。一実施例では、時間ログ・サーバ及びそのプロトコルについてウェブ・サーバは分かっているので、処理は簡素化される。この場合、以下の手順を行う。まず、ウェブ・サーバは、そのログの累計ハッシュを計算し、これをそのログに、認識可能な形式で入れる。第2に、ウェブ・サーバはこの計算された累計ハッシュを時間ログ・サーバに送出する。この時間ログ・サーバはこれを、その自らのログに入れ、エントリをもう一度ウェブ・サーバに送出して、そのログ内に挿入する。一実施例では、ウェブ・サーバによって、エントリが、先行処理における第3の処理におけるように、GET要求を用いてもう一度送出される。当然、ウェブ・サーバが時間ログ・サーバである場合にこれは最もうまく機能する。

40

【0095】

実現形態の更なる詳細

ファイル及びディレクトリの形式

一実施例では、以下のディレクトリ形式を用いる。

【0096】

yyyy/mmdd/hhmm/ss.fraction-hash

別の実施例では、以下のディレクトリ形式を用いる。

【0097】

/iii/jjj-hash

後者の形式は、クロックに対する調節にかかわらずシーケンス番号が単調増加するとい

50

う利点を有する。

【0098】

一実施例では、エントリは、サブディレクトリ毎に1000に制限され（000-999）、レベルは必要に応じて追加される。各エントリは、タイムスタンプ、現在のオフセット、及びそのシーケンス番号を含む。付加された精緻化として、一実施例では、古いディレクトリは、連結によってファイル（例えば、ジップ・ファイル）に集約され、それによって、ハッシュの一貫性の検査が容易になる。

【0099】

何れの場合にも、ハッシュ・テーブルを用いて、ハッシュをシーケンス番号、及び、少なくとも、主要イベント（例えば、再同期化、再起動、及び日次チェックポイント）のリストにマッピングすることができる。ハッシュ・テーブルをチェックポイントすることができることによって、必要な場合に、再構築することが可能になる。

【0100】

アプリケーション・スタイル

アプリケーション・スタイルは、CGI、コマンドライン、又はサーバであり得る。一実施例では、CGIアプリケーションは、パールで実現される。CGIとともに行われる可能性が高い単純なコマンド行アプリケーションを用いることができ、クロン・ジョブ、makeファイル（例えば、リリースが作成された時点をログ記録する）や同様なシェル・レベルの機能によく適している。サーバ・アプリケーション・スタイルが、アクセスに、より好適である。前述の場合、サーバは、データベースを保持することが可能であり、ログに対するアクセスを直列化することができる。マイナス面では、クラッシュや停電を考えると、サーバをロバストにするための留意がない限り、サーバの使用は、あまり信頼できないことがあり得る。

【0101】

NTPとの関係

一実施例では、時間ログ・システムを用いて、特定文書のタイムスタンプの生成に、NTP追跡可能なタイムスタンプを用いたことを事後に証明することが可能である。これは、NTP時間と、文書に対するタイムスタンプとの間の非常に必要な信頼できるリンクを提供する。

【0102】

コンピュータ・システムの例

図6は、本明細書及び特許請求の範囲記載の1つ又は複数の動作を実行することができる例示的なコンピュータ・システムを示すブロック図である。図6を参照すれば、コンピュータ・システムは、例示的なクライアント又はサーバ・コンピュータ・システム（埋め込みサーバなど）を備え得る。コンピュータ・システムは、情報を通信する通信機構すなわちバス611と、バス611に結合され、情報を処理するプロセッサ612とを含む。プロセッサ612は、マイクロプロセッサ（例えば、ペンティアム（登録商標）、パワーPC（商標）、等など）を含むが、マイクロプロセッサに限定されない。

【0103】

コンピュータ・システムは、バス611に結合され、情報、及びプロセッサ612によって実行される対象の命令を記憶するランダム・アクセス・メモリ（RAM）やその他のダイナミック記憶装置604（主メモリと表す）を更に備える。主メモリ604は、プロセッサ612による命令の実行中の一時変数やその他の中間情報の記憶にも用いることができる。

【0104】

コンピュータ・システムは、バス611に結合され、スタティック情報と、プロセッサ612のための命令とを記憶するリード・オンリ・メモリ（ROM）及び／又はその他のスタティック記憶装置606、並びに、データ記憶装置607（磁気ディスク又は光ディスク、及びその対応するディスク・ドライブなど）も備える。データ記憶装置607は、バス611に結合され、情報及び命令を記憶する。

【0105】

10

20

30

40

50

バス611に結合することができる別の装置は、媒体（紙、フィルムや同様なタイプの媒体など）上に命令、データやその他の情報を印刷するために用いることができるハードコピー装置624である。更に、コンピュータ・システムとのオーディオ・インタフェースのために、音響記録及び再生装置（スピーカ及び／又はマイクロフォンなど）を任意的にバス611に結合することができる。バス611に結合することができる別の装置は、電話機、又はハンドヘルド型パーム・デバイスに通信するための有線／無線通信機能部625である。

【0106】

本発明の多くの改変及び修正が、当業者が前述の記載を読んだ後に明らかになることに疑いの余地はないが、例証として示し、説明している特定の実施例は何れも、限定的であるとみなされることを意図するもので決してない。したがって、種々の実施例の詳細への参照は、本発明に不可欠なものとみなす構成のみをそれ自体が記載した特許請求の範囲記載の範囲を限定することを意図するものでない。

【図面の簡単な説明】

【0107】

【図1】 エントリをログ記録する方法の一実施例を示すフロー図である。

【図2】 ログにおけるログ・エントリを再同期化させる方法の一実施例のフロー図である。

【図3】 ログ・エントリを調節する方法の一実施例のフロー図である。

【図4】 ログを処理する方法の一実施例のフロー図である。

【図5】 ログを処理する方法の別の実施例のフロー図である。

【図6】 コンピュータ・システムの一実施例のブロック図である。

【符号の説明】

【0108】

604 主メモリ

606 スタティック・メモリ

607 大容量記憶メモリ

611 バス

612 プロセッサ

620 外部ネットワーク・インタフェース

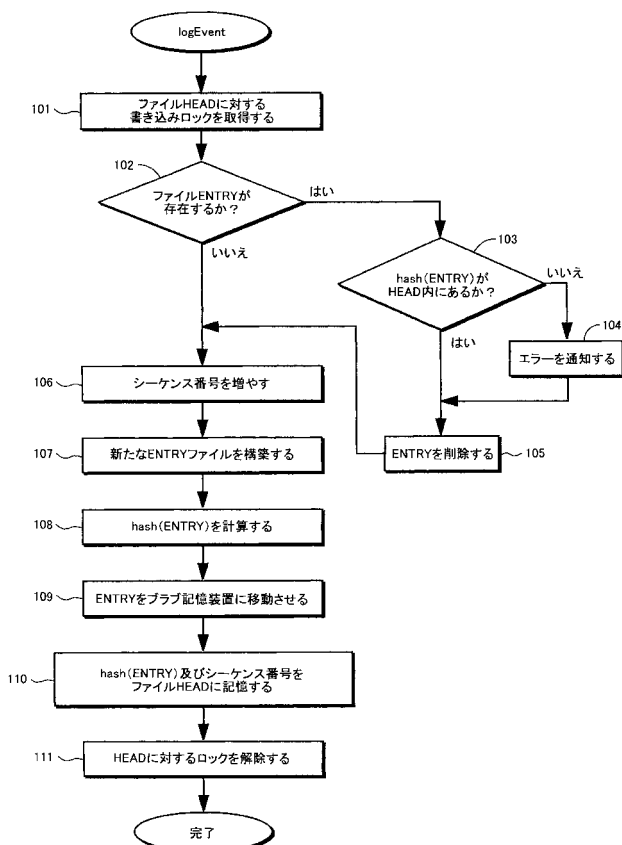
624 ハードコピー装置

10

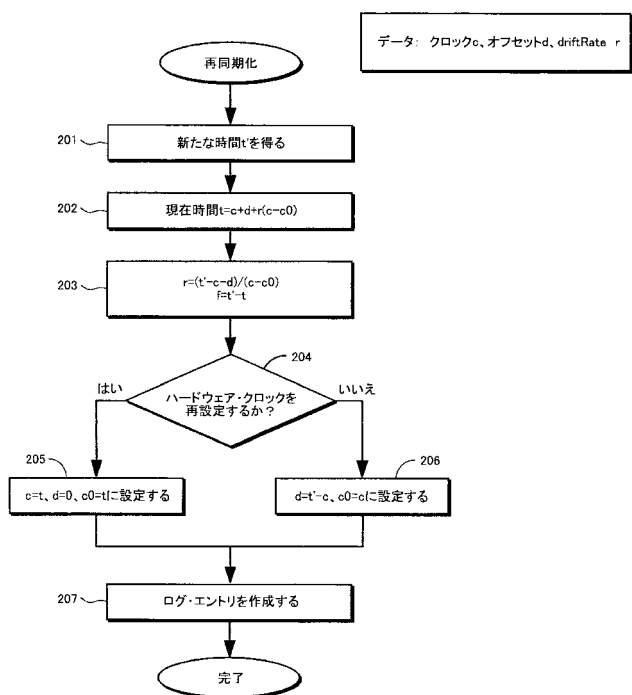
20

30

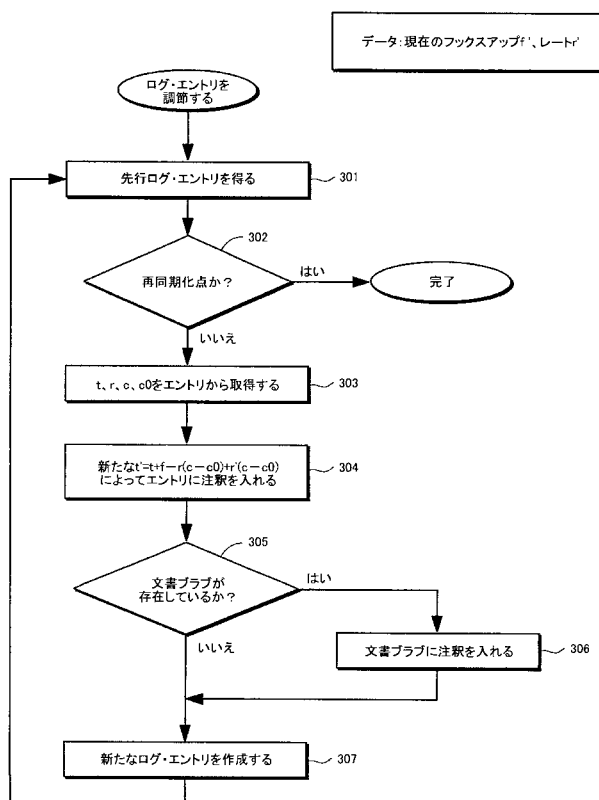
【図 1】



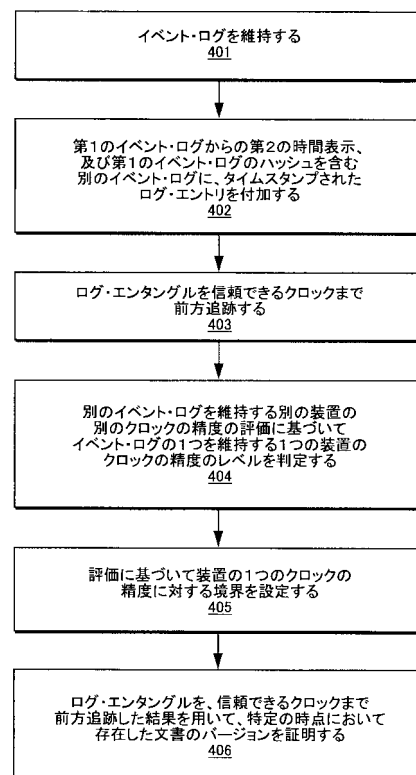
【図 2】



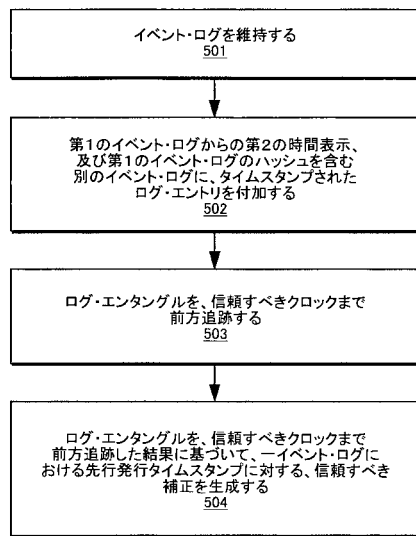
【図 3】



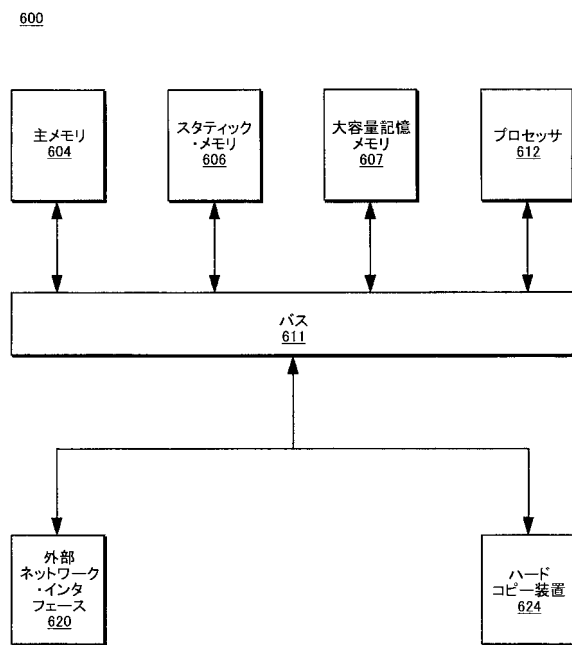
【図 4】



【図 5】



【図 6】



フロントページの続き

(72)発明者 カート ピアソル

アメリカ合衆国, カリフォルニア 94025-7054, メンロ・パーク, サンド・ヒル・ロード 2882番, スイート 115 リコー イノベーション インク内

Fターム(参考) 5J104 AA09 AA11 GA03 GA05 JA21 LA03 LA06 NA02 NA12 NA27
NA37 NA38