

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7335502号  
(P7335502)

(45)発行日 令和5年8月30日(2023.8.30)

(24)登録日 令和5年8月22日(2023.8.22)

(51)国際特許分類	F I			
G 0 6 F 3/06 (2006.01)	G 0 6 F	3/06	3 0 4 F	
	G 0 6 F	3/06	3 0 1 S	
	G 0 6 F	3/06	3 0 4 N	

請求項の数 8 (全33頁)

(21)出願番号	特願2019-184256(P2019-184256)	(73)特許権者	000005223
(22)出願日	令和1年10月7日(2019.10.7)		富士通株式会社
(65)公開番号	特開2021-60780(P2021-60780A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43)公開日	令和3年4月15日(2021.4.15)	(74)代理人	110002918
審査請求日	令和4年6月9日(2022.6.9)		弁理士法人扶桑国際特許事務所
		(72)発明者	高橋 英一
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72)発明者	司波 章
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72)発明者	岡林 美和
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

(54)【発明の名称】 情報処理システム、情報処理方法および情報処理プログラム

(57)【特許請求の範囲】

【請求項1】

バッファに格納された入力データに対して実行される処理に対応する状態データを記憶する第1の記憶装置と、

前記バッファに対する前記入力データの入力レートから前記バッファのオーバーフローが生じる第1の時刻を計算し、前記入力レートと単位時間当たりの前記状態データの更新量と前記第1の記憶装置から第2の記憶装置への前記状態データのバックアップ処理の第1のスループットと前記第2の記憶装置から前記第1の記憶装置への前記状態データの復元処理の第2のスループットと前記処理の第3のスループットとに基づいて、前記状態データのバックアップ処理を開始する第2の時刻であって、前記第1の時刻よりも前の前記第2の時刻を決定する処理装置と、

を有する情報処理システム。

【請求項2】

前記処理装置は、前記第2の時刻に達する前に、前記入力レート、単位時間当たりの前記状態データの更新量、前記第1のスループット、前記第2のスループットおよび前記第3のスループットの少なくとも1つが更新されると、前記第2の時刻を更新する、請求項1記載の情報処理システム。

【請求項3】

前記処理装置は、

単位時間当たりの前記状態データの更新量と前記第1のスループットとに基づいて、前

記第 1 の時刻に前記バックアップ処理を完了させる場合の前記バックアップ処理の第 1 の開始時刻を計算し、

前記入力レートと前記第 2 のスループットと前記第 3 のスループットと前記バックアップされたサイズとに基づいて、前記第 1 の開始時刻に、前記状態データのバックアップによる前記第 1 の記憶装置の前記状態データの復旧を完了させる場合の前記復元処理の第 2 の開始時刻を計算し、

単位時間当たりの前記状態データの更新量と前記第 1 のスループットとに基づいて、前記第 2 の開始時刻に前記バックアップ処理を完了させる場合の前記バックアップ処理の第 3 の開始時刻を計算し、

前記第 3 の開始時刻または前記第 3 の開始時刻よりも所定時間前の時刻を、前記第 2 の時刻とする、

請求項 1 記載の情報処理システム。

【請求項 4】

前記処理装置は、前記第 2 の時刻で前記バックアップ処理を開始し、前記バックアップ処理を終了すると、前記バッファに格納された前記入力データのうち、前記第 2 の時刻よりも前に前記処理に入力済である前記入力データを前記バッファから削除する、請求項 1 記載の情報処理システム。

【請求項 5】

前記処理装置は、前記処理に用いられる前記入力データが格納される複数の前記バッファそれぞれに対して前記第 2 の時刻の候補時刻を計算し、計算した複数の前記候補時刻のうち、最も早い前記候補時刻を前記第 2 の時刻とする、請求項 1 記載の情報処理システム。

【請求項 6】

前記処理は、前記入力データを処理する複数のタスクを含むストリーム処理である、請求項 1 記載の情報処理システム。

【請求項 7】

コンピュータが、

入力データを格納するバッファに対する前記入力データの入力レートから前記バッファのオーバーフローが生じる第 1 の時刻を計算し、

前記入力レートと前記入力データに対して実行される処理に対応する状態データの単位時間当たりの更新量と前記状態データを記憶する第 1 の記憶装置から第 2 の記憶装置への前記状態データのバックアップ処理の第 1 のスループットと前記第 2 の記憶装置から前記第 1 の記憶装置への前記状態データの復元処理の第 2 のスループットと前記処理の第 3 のスループットとに基づいて、前記状態データのバックアップ処理を開始する第 2 の時刻であって、前記第 1 の時刻よりも前の前記第 2 の時刻を決定する、

情報処理方法。

【請求項 8】

コンピュータに、

入力データを格納するバッファに対する前記入力データの入力レートから前記バッファのオーバーフローが生じる第 1 の時刻を計算し、

前記入力レートと前記入力データに対して実行される処理に対応する状態データの単位時間当たりの更新量と前記状態データを記憶する第 1 の記憶装置から第 2 の記憶装置への前記状態データのバックアップ処理の第 1 のスループットと前記第 2 の記憶装置から前記第 1 の記憶装置への前記状態データの復元処理の第 2 のスループットと前記処理の第 3 のスループットとに基づいて、前記状態データのバックアップ処理を開始する第 2 の時刻であって、前記第 1 の時刻よりも前の前記第 2 の時刻を決定する、

処理を実行させる情報処理プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は情報処理システム、情報処理方法および情報処理プログラムに関する。

10

20

30

40

50

## 【背景技術】

## 【0002】

情報処理システムとして、様々な端末装置からイベントを示す入力データを受信し、情報処理システムが保持するデータをイベントに応じて更新し、データの更新に応じた情報処理を実行するイベント処理システムがある。例えば、情報処理システムは、スマートフォンや車載装置などの端末装置から、センサデバイスを用いて測定されたセンサ情報を含む入力データを収集し、端末装置などの状態を示す状態データを保持する。例えば、情報処理システムは、状態データの更新を検出し、端末装置などの現在の状態に応じた情報を提供するという情報サービスを実現する。

## 【0003】

ここで、情報処理システムでは、障害時などのデータ消失に備えて、データのバックアップが行われることがある。例えば、所定のバックアップインターバルに基づいてサーバにおいて接続データセット（電子メールや連絡先など）のバックアップを生成するシステムの提案がある。

## 【0004】

また、ユーザ装置による通信サービスの実行要求が発生する時、ユーザ装置に対応付けられた仮想サーバを使用して通信サービスを実行するデータセンタ装置の提案がある。提案のデータセンタ装置は、通信サービスの実行によりバックアップが必要な場合、仮想サーバに対応するバックアップスケジュール及びCPU（Central Processing Unit）によりバックアップサービスを実行する。

## 【先行技術文献】

## 【特許文献】

## 【0005】

【文献】特表2009-501499号公報

特開2013-232807号公報

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0006】

情報処理システムで保持される状態データのバックアップを取得することがある。ここで、情報処理システムで受信された入力データは、バッファに保持される。ある時点の状態データのバックアップを取得すると、当該時点以前の入力データは、状態データの復元に不要になるので、バッファから削除できる。

## 【0007】

バッファの容量は有限である。そこで、状態データのバックアップ処理を比較的短い時間間隔で実行することでバッファオーバーフローの発生を抑止することが考えられる。しかし、状態データのバックアップ処理の頻度を高くするほど、バックアップ処理による負荷が高まり、情報処理システムの処理性能が低下する。

## 【0008】

一方、バックアップ処理の時間間隔を長くするほど、バッファに保持されるデータ量が大きくなる。このため、状態データの復旧時に、復旧処理の間に入力される入力データによりバッファオーバーが生じる可能性が高くなる。

## 【0009】

1つの側面では、本発明は、バックアップ処理を行うタイミングを適切に決定できる情報処理システム、情報処理方法および情報処理プログラムを提供することを目的とする。

## 【課題を解決するための手段】

## 【0010】

1つの態様では、情報処理システムが提供される。この情報処理システムは、第1の記憶装置と処理装置とを有する。第1の記憶装置は、バッファに格納された入力データに対して実行される処理に対応する状態データを記憶する。処理装置は、バッファに対する入力データの入力レートからバッファのオーバーフローが生じる第1の時刻を計算し、入力

10

20

30

40

50

レートと単位時間当たりの状態データの更新量と第 1 の記憶装置から第 2 の記憶装置への状態データのバックアップ処理の第 1 のスループットと第 2 の記憶装置から第 1 の記憶装置への状態データの復元処理の第 2 のスループットと処理の第 3 のスループットとに基づいて、状態データのバックアップ処理を開始する第 2 の時刻であって、第 1 の時刻よりも前の第 2 の時刻を決定する。

【 0 0 1 1 】

また、1つの態様では、情報処理方法が提供される。

また、1つの態様では、情報処理プログラムが提供される。

【発明の効果】

【 0 0 1 2 】

1つの側面では、バックアップ処理を行うタイミングを適切に決定できる。

【図面の簡単な説明】

【 0 0 1 3 】

【図 1】第 1 の実施の形態の情報処理システムの例を示す図である。

【図 2】第 2 の実施の形態の情報処理システムの例を示す図である。

【図 3】ノードのハードウェア例を示すブロック図である。

【図 4】情報処理システムの機能例を示すブロック図である。

【図 5】ストリーム処理部の例を示す図である。

【図 6】入力バッファ部の例を示す図である。

【図 7】入力バッファ部のページ例を示す図である。

【図 8】バックアップの例を示す図である。

【図 9】復旧の例を示す図である。

【図 10】入力レートの例を示す図である。

【図 11】バッファデータ量計測テーブルの例を示す図である。

【図 12】スループット計測テーブルの例を示す図である。

【図 13】ステート更新サイズ計測テーブルの例を示す図である。

【図 14】復元スループット計測テーブルの例を示す図である。

【図 15】チェックポイントスループット計測テーブルの例を示す図である。

【図 16】チェックポイント開始時刻決定例を示すフローチャートである。

【図 17】チェックポイント開始時刻の決定例を示す図である。

【図 18】チェックポイント開始時刻決定の他の例を示すフローチャートである。

【図 19】情報処理システムの他の例を示す図である。

【発明を実施するための形態】

【 0 0 1 4 】

以下、本実施の形態について図面を参照して説明する。

[第 1 の実施の形態]

第 1 の実施の形態を説明する。

【 0 0 1 5 】

図 1 は、第 1 の実施の形態の情報処理システムの例を示す図である。

情報処理システム 10 は、バッファ 11、第 1 の記憶装置 12、第 2 の記憶装置 13 および処理装置 14 を含む。

【 0 0 1 6 】

バッファ 11 は、入力データを記憶する。情報処理システム 10 が他の装置（図示を省略している）から受信する入力データは、バッファ 11 に格納される。バッファ 11 に対する入力データの書き込みや読み出しの手順には、例えば F I F O（First-In/First-Out）が用いられる。第 1 の記憶装置 12 は、R A M（Random Access Memory）などの揮発性記憶装置である。第 1 の記憶装置 12 は、処理装置 14 により実行される処理に対応する状態データを記憶する。第 2 の記憶装置 13 は、H D D（Hard Disk Drive）や S S D（Solid State Drive）などの不揮発性記憶装置である。第 2 の記憶装置 13 は、第 1 の記憶装置 12 に格納された状態データのバックアップを記憶する。

10

20

30

40

50

## 【 0 0 1 7 】

ここで、バッファ 1 1 には、第 1 の記憶装置 1 2 および処理装置 1 4 とは別個に電源が供給される。例えば、バッファ 1 1 が第 1 の情報処理装置（図示を省略している）に搭載され、第 1 の記憶装置 1 2 および処理装置 1 4 が第 2 の情報処理装置（図示を省略している）に搭載されてもよい。なお、第 2 の記憶装置 1 3 は、第 2 の情報処理装置の外部ストレージでもよいし、第 2 の情報処理装置の内部に搭載されてもよい。バッファ 1 1 には、第 1 の記憶装置 1 2 とは別個に電源が供給されるので、仮に、第 1 の記憶装置 1 2 や処理装置 1 4 がダウンしても、新たに受信した入力データが格納され続ける。

## 【 0 0 1 8 】

処理装置 1 4 は、例えば、CPU、GPU (Graphics Processing Unit)、DSP (Digital Signal Processor) などのプロセッサを含む。ただし、処理装置 1 4 は、ASIC (Application Specific Integrated Circuit) や FPGA (Field Programmable Gate Array) などの特定用途の電子回路を含んでもよい。プロセッサは、メモリ（第 1 の記憶装置 1 2 でもよい）に記憶されたプログラムを実行する。複数のプロセッサの集合を「マルチプロセッサ」または単に「プロセッサ」と言うことがある。

10

## 【 0 0 1 9 】

処理装置 1 4 は、バッファ 1 1 に格納される入力データに基づく処理を実行し、第 1 の記憶装置 1 2 に記憶された状態データを更新する。処理装置 1 4 において入力データに対して実行される処理の単位はタスクと呼ばれる。例えば、処理装置 1 4 は、入力データに対して複数のタスクを順次実行するストリーム処理を行ってもよい。ストリーム処理は、複数のタスクを含むパイプラインにより実現され得る。この場合、1 つのタスクは、パイプラインの 1 つのステージに相当する。状態データは、複数のタスクの状態を示す情報でもよい。

20

## 【 0 0 2 0 】

処理装置 1 4 は、処理実行部 1 4 a、バックアップ時刻決定部 1 4 b およびバックアップ実行部 1 4 c を有する。

処理実行部 1 4 a は、入力データに対するタスクを実行する。処理実行部 1 4 a は、ストリーム処理を実行する場合、入力データに対して複数のタスクを順番に実行することができる。処理実行部 1 4 a は、タスクの実行に応じて、第 1 の記憶装置 1 2 に記憶された状態データを更新することができる。

30

## 【 0 0 2 1 】

バックアップ時刻決定部 1 4 b は、第 1 の記憶装置 1 2 に記憶された状態データのバックアップ処理を開始する時刻を決定する。

まず、バックアップ時刻決定部 1 4 b は、バッファ 1 1 に対する入力データの入力レート  $R$  からバッファ 1 1 のオーバーフローが生じる第 1 の時刻  $T_0$  を計算する。入力レート  $R$  は、バッファ 1 1 に対して単位時間あたりに格納される入力データのデータ量を示す。また、バックアップ時刻決定部 1 4 b は、次の 3 つの種類の処理のスループットを取得する。スループットは、単位時間あたりに処理可能なデータ量を示す。

## 【 0 0 2 2 】

第 1 のスループットは、第 1 の記憶装置 1 2 から第 2 の記憶装置 1 3 への状態データのバックアップ処理のスループットである。第 1 のスループットは、バックアップ処理の所要時間に関するパラメータとして用いられる。例えば、バックアップ時刻決定部 1 4 b は、第 1 のスループットと該当の時刻におけるバックアップ対象のデータのサイズとに基づいて、バックアップ処理の所要時間を求めることができる。

40

## 【 0 0 2 3 】

第 2 のスループットは、第 2 の記憶装置 1 3 から第 1 の記憶装置 1 2 への状態データの復元処理のスループットである。第 2 のスループットは、復元処理の所要時間に関するパラメータとして用いられる。例えば、バックアップ時刻決定部 1 4 b は、第 2 のスループットと復元対象のデータのサイズとに基づいて、復元処理の所要時間を求めることができる。

50

## 【 0 0 2 4 】

第3のスループットは、入力データに対して、処理実行部14aにより実行される処理のスループットである。第3のスループットは、再計算処理の所要時間に関するパラメータとして用いられる。再計算処理は、バッファ11に保持されている入力データを用いてタスクを実行することで、復元後の状態データから障害発生時の直前の状態データを再計算する処理である。例えば、バックアップ時刻決定部14bは、第3のスループットとバッファ11に保持される入力データのサイズとに基づいて、再計算処理の所要時間を求めることができる。

## 【 0 0 2 5 】

そして、バックアップ時刻決定部14bは、入力レートRと単位時間当たりの状態データの更新量と第1のスループットと第2のスループットと第3のスループットとに基づいて、状態データのバックアップ処理を開始する第2の時刻 $t_d$ を決定する。第2の時刻 $t_d$ は、第1の時刻 $T_o$ よりも前の時刻である。

10

## 【 0 0 2 6 】

ここで、図1には、グラフG1が示されている。グラフG1は、時間 $t$ とバッファ11におけるバッファデータ量 $L$ との関係の例を示す。グラフG1の横軸が時間 $t$ であり、縦軸がバッファデータ量 $L$ である。時間 $t$ は、前回バックアップ終了時刻 $t_0$ を原点とし、 $t_0$ からの経過時間を表す。各時刻は時刻 $t_0$ からの経過時間によって表される。例えば、時刻 $T_o$ と言う場合、時刻 $t_0$ からの経過時間が $T_o$ である時刻を指す。各時刻におけるバッファデータ量 $L$ は、時刻 $t_0$ でのバッファデータ量 $L_0$ と入力レートRと時刻 $t_0$ からの経過時間により求めることができる。バッファデータ量 $L$ の上限は $L_{max}$ である。バッファデータ量 $L$ が上限 $L_{max}$ を超えると、バッファオーバーフローが発生する。 $T_o = (L_{max} - L_0) / R$ である。

20

## 【 0 0 2 7 】

例えば、バックアップ時刻決定部14bは、時刻 $t_i$ において、入力レートR、時刻 $T_o$ および上記の第1、第2、第3のスループットを取得する。

すると、まず、バックアップ時刻決定部14bは、単位時間当たりの状態データの更新量( $u$ とする)と第1のスループット( $TH_1$ とする)とに基づいて、時刻 $T_o$ よりも時間 $t_1$ だけ前の時刻 $t_a$ を求める。時間 $t_1$ は、時刻 $t_a$ における状態データのバックアップ処理に対して予測される所要時間である。時刻 $t_a$ は、バックアップ処理を時刻 $T_o$ に完了させる場合の当該バックアップ処理の開始時刻である。具体的には、 $t_1 = u \times t_a / TH_1$ である。 $T_o - t_1 = t_a$ を $t_a$ について解くことで、時刻 $t_a$ を得られる。

30

## 【 0 0 2 8 】

次に、バックアップ時刻決定部14bは、入力レートRと第3のスループット( $TH_3$ とする)に基づいて、時刻 $t_a$ よりも時間 $t_2$ だけ前の時刻 $t_b$ を求める。また、バックアップ時刻決定部14bは、前回バックアップされたサイズ( $U$ とする)と第2のスループット( $TH_2$ とする)とに基づいて、時刻 $t_b$ よりも時間 $t_3$ だけ前の時刻 $t_c$ を求める。時間 $t_3$ は、時刻 $t_c$ で障害が発生して、第1の記憶装置12の状態データが消失した場合に、時刻 $t_0$ で取得済のバックアップから第1の記憶装置12の状態データを復元する復元処理に対して予測される所要時間である。時間 $t_2$ は、時刻 $t_b$ で復元完了した状態データを、時刻 $t_b$ の状態まで再計算する再計算処理に対して予測される所要時間である。時刻 $t_c$ は、バックアップされた状態の復元処理および再計算処理を時刻 $t_a$ に完了させる場合の当該復元処理の開始時刻である。具体的には、 $t_2 = R \times t_b / TH_3$ である。 $t_3 = U / TH_2$ である。 $t_a - t_2 = t_b$ を $t_b$ について解くことで、時刻 $t_b$ を得られる。時刻 $t_c$ は、 $t_c = t_b - t_3 - ($ は0以上の実数)により得られる。この式で表されるように、時刻 $t_c$ は、 $t_b - t_3$ よりも時間 $t_3$ だけ前の時刻でもよい。例えば、 $t_3$ は、システム停止検知から復旧開始までにかかる時間に応じて予め見積もられる。

40

## 【 0 0 2 9 】

50

更に、バックアップ時刻決定部 14 b は、単位時間当たりの状態データの更新量と第 1 のスループットとに基づいて、時刻  $t_c$  よりも時間  $t_4$  だけ前の時刻  $t_d$  を求める。時間  $t_4$  は、時刻  $t_d$  における状態データのバックアップ処理に対して予測される所要時間である。時刻  $t_d$  は、バックアップ処理を時刻  $t_c$  に完了させる場合の当該バックアップ処理の開始時刻である。具体的には、 $t_4 = u \times t_d / TH\_1$  である。 $t_c - t_4 = t_d$  を  $t_d$  について解くことで、時刻  $t_d$  を得られる。

【0030】

そして、バックアップ時刻決定部 14 b は、例えば、時刻  $t_d$  を、次のバックアップ処理の開始時刻と決定する。ただし、バックアップ時刻決定部 14 b は、次のバックアップ処理の開始時刻を、時刻  $t_d$  よりも所定時間前の時刻としてもよい。

10

【0031】

バックアップ実行部 14 c は、バックアップ時刻決定部 14 b により決定された時刻  $t_d$  に、状態データのバックアップ処理を開始する。バックアップ実行部 14 c は、バックアップ処理において、第 1 の記憶装置 12 に記憶された状態データの複製を、第 2 の記憶装置 13 に格納する。バックアップ実行部 14 c は、時刻  $t_d$  で状態データのバックアップ処理を開始し、当該バックアップ処理を終了すると、当該時刻  $t_d$  よりも前に処理実行部 14 a に入力済である入力データをバッファ 11 から削除する。これにより、バッファ 11 の空き容量が増える。

【0032】

情報処理システム 10 によれば、バッファ 11 に対する入力データの入力レート  $R$  からバッファ 11 のオーバーフローが生じる第 1 の時刻  $T_o$  が計算される。第 1 の時刻  $T_o$  とバックアップ処理の第 1 のスループットと復元処理の第 2 のスループットとタスクによる処理の第 3 のスループットとに基づいて、バックアップ処理を開始する第 2 の時刻であって、第 1 の時刻  $T_o$  よりも前の第 2 の時刻  $t_d$  が決定される。

20

【0033】

これにより、バックアップ処理を行うタイミングを適切に決定できる。

ここで、時刻  $T_o$  は、バッファ 11 のデータサイズが上限サイズ  $L_{max}$  に到達する予測時刻である。時刻  $T_o$  までにバックアップ処理が完了すれば、バッファ 11 に空きができるため、バッファ 11 のオーバーフローを防げる。時刻  $t_a$  は、時刻  $T_o$  までにバックアップ処理を完了できる最終のバックアップ開始時刻である。

30

【0034】

また、情報処理システム 10 では、障害などにより、第 1 の記憶装置 12 の状態データが消失し、状態データの復旧処理が行われる可能性がある。復旧処理は、バックアップからの状態データの復元処理および復元された状態データによる再計算処理のセットである。時刻  $t_c$  は、時刻  $t_a$  に復旧処理を完了できる最終の復元開始時刻である。したがって、復旧処理が時刻  $t_c$  以降に起きないことを保証するためには、時刻  $t_c$  までにバックアップ処理が完了すればよい。時刻  $t_d$  は、時刻  $t_c$  までにバックアップ処理を完了できる最終のバックアップ開始時刻である。このため、例えば、処理装置 14 は、時刻  $t_d$  で次のバックアップ処理の実行を開始することで、仮に、復旧処理が生じて、バッファ 11 がオーバーフローを起こすことを防げる。

40

【0035】

特に、情報処理システム 10 では、入力レート  $R$  や第 1、第 2、第 3 のスループットで表されるデータ処理コストが変動しても、変動に応じてバックアップ処理のタイミングを調整可能になる。情報処理システム 10 におけるバックアップ処理を、バッファオーバーフローが生じない最低限の頻度とすることで、バックアップ処理のコストを低減し、情報処理システム 10 の本来のタスク処理の性能が低下することを抑制できる。

【0036】

[第 2 の実施の形態]

次に、第 2 の実施の形態を説明する。

図 2 は、第 2 の実施の形態の情報処理システムの例を示す図である。

50

## 【 0 0 3 7 】

第2の実施の形態の情報処理システムは、イベントメッセージをエッジデバイスから収集し、イベントメッセージをリアルタイムに処理してイベント駆動型のサービスを提供するイベント処理システムである。

## 【 0 0 3 8 】

第2の実施の形態の情報処理システムは、ノード100、メッセージサーバ200およびエッジ装置300、300a、300bを含む。メッセージサーバ200およびエッジ装置300、300a、300bは、ネットワーク20に接続されている。ネットワーク20は、例えばインターネットなどの広域通信網である。ノード100およびメッセージサーバ200は、例えばデータセンタに設けられる。ノード100およびメッセージサーバ200は、データセンタ内のLAN(Local Area Network)に接続されてもよい。

10

## 【 0 0 3 9 】

ノード100は、イベントメッセージに対してストリーム処理を行うサーバコンピュータである。ノード100は、メッセージサーバ200から次々に入力されるイベントメッセージを次々に処理する。ストリーム処理は複数の処理要素を含むパイプラインによって実現される。ストリーム処理におけるパイプラインの1ステージに相当するサブ処理はタスクと呼ばれる。

## 【 0 0 4 0 】

ストリーム処理では、タスク毎にデータ処理結果をステート(状態)としてタスク単位に保持し、後のデータ処理で当該ステートを用いることがある。このようなストリーム処理は、ステートフルなストリーム処理と呼ばれる。ノード100は、ステートフルなストリーム処理を実行する。第2の実施の形態の情報処理システムは、ノード100を複数有してもよい。ステートは、第1の実施の形態の状態データに対応する。

20

## 【 0 0 4 1 】

メッセージサーバ200は、エッジ装置300、300a、300bからイベントメッセージを受信し、イベントメッセージを保持するサーバコンピュータである。メッセージサーバ200は、イベントメッセージのバッファとして機能し、当該イベントメッセージをノード100に送信する。メッセージサーバ200に対して次々に入力されるイベントメッセージは、データストリームと呼ばれることがある。

## 【 0 0 4 2 】

エッジ装置300、300a、300bは、メッセージサーバ200にイベントメッセージを送信するエッジコンピュータである。エッジ装置300、300a、300bは、センサデバイスでもよいし、末端のセンサデバイスからイベントメッセージを集約してメッセージサーバ200に送信するエッジ側サーバコンピュータでもよい。例えば、エッジ装置300、300a、300bは、自動車の速度を示すイベントメッセージをメッセージサーバ200に送信する車載装置でもよい。

30

## 【 0 0 4 3 】

図3は、ノードのハードウェア例を示すブロック図である。

ノード100は、CPU101、RAM102、HDD103、画像信号処理部104、入力信号処理部105、媒体リーダ106およびNIC(Network Interface Card)107を有する。ノード100の上記ユニットは、ノード100のバスに接続されている。なお、CPU101は、第1の実施の形態の処理装置14に対応する。RAM102は、第1の実施の形態の第1の記憶装置12に対応する。HDD103は、第1の実施の形態の第2の記憶装置13に対応する。

40

## 【 0 0 4 4 】

CPU101は、プログラムの命令を実行するプロセッサである。CPU101は、HDD103に記憶されたプログラムやデータの少なくとも一部をRAM102にロードし、プログラムを実行する。なお、CPU101は複数のプロセッサコアを含んでもよい。また、ノード100は複数のプロセッサを有してもよい。以下で説明する処理は複数のプロセッサまたはプロセッサコアを用いて並列に実行されてもよい。また、複数のプロセッ

50

サの集合を「マルチプロセッサ」または単に「プロセッサ」と言うことがある。

【0045】

RAM102は、CPU101が実行するプログラムやCPU101が演算に用いるデータを一時的に記憶する揮発性の半導体メモリである。なお、ノード100は、RAM以外の種類のメモリを備えてもよく、複数個のメモリを備えてもよい。

【0046】

HDD103は、OS（Operating System）やミドルウェアやアプリケーションソフトウェアなどのソフトウェアのプログラム、および、データを記憶する不揮発性の記憶装置である。なお、ノード100は、フラッシュメモリやSSD（Solid State Drive）などの他の種類の記憶装置を備えてもよく、複数の不揮発性の記憶装置を備えてもよい。

10

【0047】

画像信号処理部104は、CPU101からの命令に従って、ノード100に接続されたディスプレイ21に画像を出力する。ディスプレイ21としては、CRT（Cathode Ray Tube）ディスプレイ、液晶ディスプレイ（LCD：Liquid Crystal Display）、プラズマディスプレイ、有機EL（OEL：Organic Electro-Luminescence）ディスプレイなど、任意の種類のディスプレイを用いることができる。

【0048】

入力信号処理部105は、ノード100に接続された入力デバイス22から入力信号を取得し、CPU101に出力する。入力デバイス22としては、マウス・タッチパネル・タッチパッド・トラックボールなどのポインティングデバイス、キーボード、リモートコントローラ、ボタンスイッチなどを用いることができる。また、ノード100に、複数の種類の入力デバイスが接続されていてもよい。

20

【0049】

媒体リーダ106は、記録媒体23に記録されたプログラムやデータを読み取る読み取り装置である。記録媒体23として、例えば、磁気ディスク、光ディスク、光磁気ディスク（MO：Magneto-Optical disk）、半導体メモリなどを使用できる。磁気ディスクには、フレキシブルディスク（FD：Flexible Disk）やHDDが含まれる。光ディスクには、CD（Compact Disc）やDVD（Digital Versatile Disc）が含まれる。

【0050】

媒体リーダ106は、例えば、記録媒体23から読み取ったプログラムやデータを、RAM102やHDD103などの他の記録媒体にコピーする。読み取られたプログラムは、例えば、CPU101によって実行される。なお、記録媒体23は可搬型記録媒体であってもよく、プログラムやデータの配布に用いられることがある。また、記録媒体23やHDD103を、コンピュータ読み取り可能な記録媒体と言うことがある。

30

【0051】

NIC107は、ネットワーク20に接続され、ネットワーク20を介して他のコンピュータと通信を行うインタフェースである。NIC107は、例えば、スイッチやルータなどの通信装置とケーブルで接続される。

【0052】

図4は、情報処理システムの機能例を示すブロック図である。

40

ノード100は、記憶部110、ストリーム処理部120、チェックポイントスケジューラ部130、スループット計測部140、ステート更新サイズ計測部150、復元スループット計測部160およびチェックポイント制御部170を有する。記憶部110には、例えば、RAM102の記憶領域が使用される。ストリーム処理部120、チェックポイントスケジューラ部130、スループット計測部140、ステート更新サイズ計測部150、復元スループット計測部160およびチェックポイント制御部170は、例えば、プログラムを用いて実装される。

【0053】

記憶部110は、各種の計測データを記憶する。計測データは、下記の情報を含む。第1には、メッセージサーバ200に対するイベントメッセージの入力レートである。第2

50

には、ストリーム処理部 120 によるストリーム処理のスループットである。第 3 には、ストリーム処理部 120 の各タスクによる状態更新サイズ増加率である。第 4 には、各タスクの状態の復元処理のスループットである。第 5 には、各タスクによる状態のバックアップ処理のスループットである。

【0054】

ストリーム処理部 120 は、メッセージサーバ 200 から入力されるイベントメッセージに対するストリーム処理を実行する。以下では、イベントメッセージを、単に「データ」と称する。ストリーム処理は、複数のタスクを含むパイプラインで実現される。複数のタスクそれぞれは、状態を保持する。状態は、ノード 100 の RAM 102 に格納される。

10

【0055】

チェックポイントスケジューラ部 130 は、各タスクの状態のバックアップを取得するタイミングを決定する。ここで、状態をバックアップする方法として、チェックポイントを用いる方法がある。この方法では、チェックポイントと呼ばれる特別なデータを、他のデータと同様に、ストリーム処理部 120 へ流す。ストリーム処理部 120 の各タスクは、チェックポイントを受け取ると、その時点の状態のコピーをバックアップとして HDD 103 の DB (DataBase) などへ保存する。タスクは状態のバックアップが完了したらチェックポイントを後段のタスクへ流す。最終段のタスクまでチェックポイントが流れた時点で、パイプライン全体のバックアップが完了したことになる。当該チェックポイントよりも前のデータは保持不要となるので、データを保持するバッファから当該データを破棄できる。

20

【0056】

ノード 100 が障害などで停止した後、復旧する際、まず前回保存済のバックアップを用いて、各タスクの状態をチェックポイント時点で復元する。復元後、入力バッファ部 210 の先頭からストリーム処理部 120 へデータを流す。このときの入力バッファ部 210 の先頭データは、チェックポイント直後のデータである。このようにすることで、ノード 100 は、復旧後の状態が、停止しなかった場合と同じ状態になることを保証する。

【0057】

チェックポイントスケジューラ部 130 は、後述するストリーム処理のスループット、状態更新サイズ増加率、復元処理のスループット、バックアップ処理のスループットに基づいて、チェックポイント開始時刻を決定する。チェックポイントスケジューラ部 130 は、次のチェックポイント開始時刻をチェックポイント制御部 170 にセットする。

30

【0058】

スループット計測部 140 は、ストリーム処理のスループットを計測する。ストリーム処理のスループットは、ストリーム処理部 120 のパイプライン処理を通過したデータの単位時間当たりのサイズである。スループット計測部 140 は、計測したスループットをチェックポイントスケジューラ部 130 に通知する。

【0059】

状態更新サイズ計測部 150 は、状態更新サイズを計測する。状態更新サイズは、ストリーム処理部 120 を構成する各タスクが更新した状態のデータサイズの総和である。各タスクが更新した状態のデータサイズは、各タスクの状態の更新前と更新後の差分のデータサイズとなる。状態更新サイズ計測部 150 は、状態更新サイズ増加率を求める。状態更新サイズ増加率は、単位時間当たりの状態の更新部分のサイズである。当該状態の更新部分のサイズは、状態の更新量に相当する。状態更新サイズ計測部 150 は、状態更新サイズ増加率をチェックポイントスケジューラ部 130 に通知する。

40

【0060】

復元スループット計測部 160 は、ストリーム処理部 120 の復旧処理時に、状態のバックアップサイズと状態の復元に要した時間から、復元処理のスループットを計

50

測する。復元処理のスループットを「復元スループット」と称する。復元スループット計測部 160 は、復元スループットをチェックポイントスケジュール部 130 に通知する。

【0061】

チェックポイント制御部 170 は、チェックポイントスケジュール部 130 から取得したチェックポイント開始時刻において、ストリーム処理部 120 に対してチェックポイントを発行する。チェックポイント制御部 170 は、チェックポイント発行をメッセージサーバ 200 に通知する。チェックポイント制御部 170 は、チェックポイントがストリーム処理部 120 を通過したことを検出すると、チェックポイント終了をメッセージサーバ 200 に通知する。また、チェックポイント制御部 170 は、今回のチェックポイントにおけるバックアップサイズをストリーム処理部 120 から取得し、チェックポイントの通過に要した時間からバックアップ処理のスループットを求める。ここで、バックアップ処理のスループットを、「チェックポイントスループット」と称する。チェックポイント制御部 170 は、チェックポイントスループットおよびバックアップサイズをチェックポイントスケジュール部 130 に通知する。

10

【0062】

メッセージサーバ 200 は、入力バッファ部 210、入力レート計測部 220 および入力バッファ管理部 230 を有する。入力バッファ部 210 は、例えば、RAM の記憶領域が使用される。入力レート計測部 220 および入力バッファ管理部 230 は、例えば、プログラムを用いて実装される。

【0063】

入力バッファ部 210 は、エッジ装置 300、300a、300b から受信したデータを、受信順に保持する FIFO キューである。入力バッファ部 210 に格納されたデータは、入力バッファ部 210 の先頭から順番にストリーム処理部 120 に入力される。入力バッファ部 210 は、第 1 の実施の形態のバッファ 11 に対応する。

20

【0064】

入力レート計測部 220 は、入力バッファ部 210 の入力レートを計測する。入力レートは、入力バッファ部 210 に単位時間に格納されたデータ量である。入力レート計測部 220 は、入力レートをチェックポイントスケジュール部 130 に送信する。

【0065】

入力バッファ管理部 230 は、入力バッファ部 210 内のチェックポイント発行直後のデータ位置 *cp*、ストリーム処理部 120 が次に処理するデータ位置 *offset* および入力バッファ部 210 に格納されている最後のデータ位置 *last* を保持し、更新する。例えば、データ位置 *cp*、*offset*、*last* は、入力バッファ部 210 の先頭を基準とするアドレスによって表される。

30

【0066】

入力バッファ管理部 230 は、チェックポイント制御部 170 からチェックポイント終了の通知を受け付ける。すると、入力バッファ管理部 230 は、入力バッファ部 210 の先頭からチェックポイント発行直後のデータの直前までのデータを破棄する。データ破棄後、入力バッファ部 210 の先頭データは、チェックポイント発行直後のデータとなる。

【0067】

第 2 の実施の形態の情報処理システムでは、ノード 100 が障害によりダウンしても、メッセージサーバ 200 は稼働を継続できる。このため、ノード 100 がダウンしていても、入力バッファ部 210 にはデータが格納され続ける。

40

【0068】

次に、ストリーム処理部 120 のタスク構成の例を説明する。

図 5 は、ストリーム処理部の例を示す図である。

例えば、ストリーム処理部 120 は、タスク 121、122、123 を有する。タスク 121、122、123 は、データに対して、この順に実行され、次々に入力されるデータをパイプライン処理する。ストリーム処理部 120 は、2 または 4 以上のタスクのパイプラインによって構成されてもよい。

50

## 【 0 0 6 9 】

タスク 1 2 1 , 1 2 2 , 1 2 3 は、それぞれステート  $s a 1$  ,  $s b 1$  ,  $s c 1$  を保持する。ステート  $s a 1$  ,  $s b 1$  ,  $s c 1$  は、ステート記憶部 1 8 0 に格納される。ステート記憶部 1 8 0 には、RAM 1 0 2 の記憶領域が用いられる。タスク 1 2 1 は、データに対する処理に応じて、ステート  $s a 1$  を更新することがある。タスク 1 2 2 , 1 2 3 も同様に、データに対する処理に応じて、それぞれステート  $s b 1$  ,  $s c 1$  を更新することがある。

## 【 0 0 7 0 】

ステート  $s a 1$  ,  $s b 1$  ,  $s c 1$  は、チェックポイント発行に応じて、それぞれタスク 1 2 1 , 1 2 2 , 1 2 3 によりバックアップされる。バックアップされたステート  $s a 1$  ,  $s b 1$  ,  $s c 1$  のセットを、スナップショット 1 9 1 と呼ぶ。スナップショット 1 9 1 は、スナップショット記憶部 1 9 0 に格納される。スナップショット記憶部 1 9 0 には、例えば、HDD 1 0 3 の記憶領域が用いられる。スナップショット記憶部 1 9 0 は、例えば、ノード 1 0 0 の外部ストレージによって実現されてもよい。

10

## 【 0 0 7 1 】

次に、入力バッファ部 2 1 0 の構成例を説明する。

図 6 は、入力バッファ部の例を示す図である。

入力バッファ部 2 1 0 は、前述のように FIFO の手順によって、データの書き込みや読み出しが行われる。入力バッファ部 2 1 0 の先頭のデータから順に、ストリーム処理部 1 2 0 に供給される。ストリーム処理部 1 2 0 に格納されたデータは、ページされるまで入力バッファ部 2 1 0 に残る。

20

## 【 0 0 7 2 】

図 6 では、ある時刻におけるチェックポイントの発行直後で、ページが行われる前の入力バッファ部 2 1 0 の状態を示している。データ位置  $c p$  は、チェックポイント発行直後のデータの位置である。データ位置  $o f f s e t$  は、ストリーム処理部 1 2 0 が次に処理するデータの位置である。データ位置  $l a s t$  は、入力バッファ部 2 1 0 に格納された最後のデータの位置である。

## 【 0 0 7 3 】

図 7 は、入力バッファ部のページ例を示す図である。

図 7 ( A ) は、図 6 の状態であって、ページの直前の入力バッファ部 2 1 0 の状態を示す。

30

## 【 0 0 7 4 】

図 7 ( B ) は、図 7 ( A ) の状態からのページ後の入力バッファ部 2 1 0 の状態を示す。ページにより、入力バッファ部 2 1 0 における図 7 ( A ) の先頭からデータ位置  $c p$  の直前までのデータが破棄される。これにより、図 7 ( B ) で示されるように、図 7 ( A ) のデータ位置  $c p$  のデータが入力バッファ部 2 1 0 の先頭データとなる。

## 【 0 0 7 5 】

次に、チェックポイントによるステートのバックアップの例を説明する。

図 8 は、バックアップの例を示す図である。

ステップ S 1 は、チェックポイント開始時刻におけるステップである。チェックポイント制御部 1 7 0 は、ストリーム処理部 1 2 0 にチェックポイントを発行する。ここで、入力バッファ部 2 1 0 に格納されているデータに番号を付して示す。図 8 では、チェックポイントのデータを  $c p$  と表す。ステップ S 1 の段階では、入力バッファ部 2 1 0 には、データ 1 ~ 6 が格納されている。チェックポイント  $c p$  は、データ 2 の後に発行されている。入力バッファ部 2 1 0 における次のデータ位置 (  $o f f s e t$  ) は、データ 3 の位置を示す。入力バッファ部 2 1 0 のチェックポイントのデータ位置 (  $c p$  ) は、データ 3 の位置を示す。

40

## 【 0 0 7 6 】

タスク 1 2 1 は、チェックポイント  $c p$  を受け付けたので、ステート  $s a 1$  をスナップショット記憶部 1 9 0 に保存する。ここで、タスク 1 2 1 に対して過去にバックアップさ

50

れたステートがスナップショット記憶部 190 に保存されている場合がある。この場合、タスク 121 は、過去にバックアップされたステートに対するステート s a 1 の差分をバックアップすればよい（以下のステートのバックアップについても同様）。タスク 122 はデータ 2 を処理している。タスク 122 は、データ 2 に対する処理に応じて、ステート s b 1 をステート s b 2 に更新する。タスク 123 はデータ 1 を処理している。タスク 123 は、データ 1 に対する処理に応じて、ステート s c 1 をステート s c 2 に更新する。

【0077】

ステップ S 2 は、ステップ S 1 からパイプラインが 1 段階進んだステップである。タスク 121 は、タスク 122 にチェックポイント c p を渡し、入力バッファ部 210 から次のデータ 3 を取得して、データ 3 を用いた処理を行う。タスク 121 は、データ 3 に対する処理に応じてステート s a 1 をステート s a 2 に更新する。入力バッファ部 210 における次のデータ位置 (offset) は、データ 4 の位置を示す。タスク 122 は、タスク 123 にデータ 2 を渡し、タスク 1 からチェックポイント c p を取得する。タスク 122 は、チェックポイント c p を受け付けたので、ステート s b 2 をスナップショット記憶部 190 に保存する。タスク 123 は、データ 1 の処理結果を出力し、タスク 122 からデータ 2 を取得し、データ 2 を用いた処理を行う。タスク 123 は、データ 2 に対する処理に応じて、ステート s c 2 をステート s c 3 に更新する。

10

【0078】

ステップ S 3 は、ステップ S 2 からパイプラインが 1 段階進んだステップである。タスク 121 は、タスク 122 にデータ 3 を渡し、入力バッファ部 210 から次のデータ 4 を取得して、データ 4 を用いた処理を行う。タスク 121 は、データ 4 に対する処理に応じて、ステート s a 2 をステート s a 3 に更新する。入力バッファ部 210 における次のデータ位置 (offset) は、データ 5 の位置を示す。タスク 122 は、タスク 123 にチェックポイント c p を渡し、タスク 121 からデータ 3 を取得して、データ 3 を用いた処理を行う。タスク 122 は、データ 3 に対する処理に応じてステート s b 2 をステート s b 3 に更新する。タスク 123 は、データ 2 の処理結果を出力し、タスク 122 からチェックポイント c p を取得する。タスク 123 はチェックポイント c p を受け付けたので、ステート s c 3 をスナップショット記憶部 190 に保存する。

20

【0079】

このタイミングで、チェックポイント c p に対して、チェックポイント対象のストリーム処理を構成する全てのタスク 121, 122, 123 のステート s a 1, s b 2, s c 3 の複製がスナップショット記憶部 190 に保存される。スナップショット記憶部 190 に保存されたスナップショット 192 は、ステート s a 1, s b 2, s c 3 のコピーのセットである。スナップショット 192 を取得したので、入力バッファ部 210 におけるチェックポイント c p よりも前のデータをパージする。チェックポイント c p よりも前のデータとは、入力バッファ部 210 の先頭からチェックポイント c p の発行タイミングの直前のデータ位置にあるデータであり、上記の例の場合、データ 1, 2 である。すると、入力バッファ部 210 の先頭データは、データ 3 になる。

30

【0080】

ステップ S 4 は、ステップ S 3 からパイプラインが 1 段階進んだステップである。タスク 121 は、タスク 122 にデータ 4 を渡し、入力バッファ部 210 から次のデータ 5 を取得して、データ 5 を用いた処理を行う。入力バッファ部 210 における次のデータ位置 (offset) は、データ 6 の位置を示す。タスク 122 は、タスク 123 にデータ 3 を渡し、タスク 121 からデータ 4 を取得して、データ 4 を用いた処理を行う。タスク 123 は、チェックポイント c p のデータを出力し、タスク 122 からデータ 3 を取得し、データ 3 を用いた処理を行う。

40

【0081】

次に、取得したスナップショット 192 を用いた復旧の例を説明する。

図 9 は、復旧の例を示す図である。

図 8 のステップ S 4 の後に障害が発生し、ステート記憶部 180 に記憶されたステート

50

s a 3 , s b 3 , s c 3 が消失した場合を考える。

【 0 0 8 2 】

ステップ S 5 は、障害が発生したため、ノード 1 0 0 を再起動させた直後の状態を示す。タスク 1 2 1 , 1 2 2 , 1 2 3 のプログラムは、RAM 1 0 2 上にロードされるが、ステートは失われている。

【 0 0 8 3 】

ステップ S 6 では、スナップショット記憶部 1 9 0 に保持されているスナップショット 1 9 2 により、タスク 1 2 1 , 1 2 2 , 1 2 3 それぞれのステート s a 1 , s b 2 , s c 3 が復元される。入力バッファ部 2 1 0 の次のデータ位置 ( o f f s e t ) は、入力バッファ部 2 1 0 の先頭を示すように更新される。

10

【 0 0 8 4 】

ステップ S 7 では、タスク 1 2 1 は、入力バッファ部 2 1 0 からデータ 3 を取得し、データ 3 を用いた処理を行う。タスク 1 2 1 は、データ 3 に対する処理に応じて、ステート s a 1 をステート s a 2 に更新する。入力バッファ部 2 1 0 の次のデータ位置 ( o f f s e t ) は、データ 4 の位置を示す。

【 0 0 8 5 】

ステップ S 8 では、タスク 1 2 1 は、タスク 1 2 2 にデータ 3 を渡し、入力バッファ部 2 1 0 から次のデータ 4 を取得して、データ 4 を用いた処理を行う。タスク 1 2 1 は、データ 4 に対する処理に応じて、ステート s a 2 をステート s a 3 に更新する。入力バッファ部 2 1 0 における次のデータ位置 ( o f f s e t ) は、データ 5 の位置を示す。タスク 1 2 2 は、タスク 1 2 1 からデータ 3 を取得し、データ 3 を用いた処理を行う。タスク 1 2 2 は、データ 3 に対する処理に応じて、ステート s b 2 をステート s b 3 に更新する。

20

【 0 0 8 6 】

このようにして、タスク 1 2 1 , 1 2 2 , 1 2 3 それぞれのステート s a 1 , s b 2 , s c 3 がスナップショット 1 9 2 から復元される。入力バッファ部 2 1 0 に格納されたデータを用いたステートの再計算によって、ステート s a 1 , s b 2 , s c 3 が障害直前の状態に復旧される。

【 0 0 8 7 】

次に、入力バッファ部 2 1 0 における入力レートの計測例を説明する。

図 1 0 は、入力レートの例を示す図である。

30

グラフ G 1 0 は、時間と入力バッファ部 2 1 0 に格納されているデータ量との関係の例を示す。グラフ G 1 0 の横軸は時間を示し、縦軸は入力バッファ部 2 1 0 に格納されているデータ量を示す。横軸では、時刻 t 0 を原点として、時刻 t 0 からの経過時間によって、時刻 t 1 , t 2 , t 3 , t 4 , t 5 を示している。

【 0 0 8 8 】

グラフ G 1 0 の左側には、入力バッファ部 2 1 0 が示されている。図 1 0 では、入力バッファ部 2 1 0 の下端が先頭であり、上端が最後尾である。入力バッファ部 2 1 0 が保持できる最大データ量を M A X とする。入力バッファ部 2 1 0 の先頭からデータが格納されている最後尾の位置が l a s t である。上限サイズ L \_ m a x は、入力バッファ部 2 1 0 が保持できる最大データ量 M A X からマージン分のサイズを減じたデータ量を示す。入力バッファ部 2 1 0 に格納されているデータ量が上限サイズ L \_ m a x を超えたときを、オーバーフローとする。最大データ量 M A X から減じるマージンは予め設定された値であり、バイト数で指定されてもよく、最大データ量 M A X に対する比率 (例えば、5 % など) で指定されてもよい。

40

【 0 0 8 9 】

入力レート計測部 2 2 0 は、入力バッファ部 2 1 0 に格納されているデータ量を繰り返し計測する。入力レート計測部 2 2 0 は、入力バッファ部 2 1 0 が保持する現データ量 ( l a s t ) を取得してもよい。入力バッファ部 2 1 0 にデータを格納する所定のプログラムが現データ量更新時に、更新後の現データ量を、入力レート計測部 2 2 0 に通知してもよい。図 1 0 は、時刻 t 1 , t 2 , t 3 , t 4 , t 5 それぞれで計 5 回、入力バッファ部

50

210に格納されているデータ量を計測した結果を示す。時刻t1でのデータ量はl1である。時刻t2でのデータ量はl2である。時刻t3でのデータ量はl3である。時刻t4でのデータ量はl4である。時刻t5でのデータ量はl5である。

【0090】

入力レート計測部220は、これらの計測結果に基づいて入力レートRを下記の式(1)で示される直線近似により求める。なお、数式中ではiの添え字を下付きとする。

【0091】

【数1】

$$R = \frac{\sum_i (t_i - t_{ave})(l_i - l_{ave})}{\sum_i (t_i - t_{ave})^2} \quad (1) \quad 10$$

【0092】

ここで、taveはtiの平均値である。laveはliの平均値である。図10の直線は式(1)で表される入力レートRを示す。

入力レート計測部220による計測は、入力バッファ部210がデータを受け取る時点で行われてもよく、予め設定された時間間隔で定期的に行われてもよい。

【0093】

あるいは、入力レート計測部220による計測は、入力バッファ部210のデータ量liについて、下記の式(2)で表される指標k[i]が、前回のデータ量計測時の指標k[i-1]を超えた時点(k[i] > k[i-1]の時点)で行われてもよい。

20

【0094】

【数2】

$$k[i] = \text{floor} \left( l_i \times \frac{dl}{L_{max}} \right) \quad (2)$$

【0095】

ここで、dlは予め設定された整数であり、例えば、dl = 100である。また、関数floor(x)は、xを超えない最大の整数を返す関数である。このように、チェックポイントスケジュール部130は、入力バッファ部210に格納されているデータサイズが入力バッファ部210の上限サイズに対して所定割合だけ増加するたびに、入力レートの計測結果を入力レート計測部から取得してもよい。

30

【0096】

更に、入力レート計測部220による計測は、前回のデータ量計測時から時間La/R[j]が経過した時点で行われてもよい。Laは、予め設定されたデータサイズであり、例えば、入力バッファ部210の上限サイズLmaxの1%である。R[j]は、前回計測された入力レートである。

【0097】

入力レート計測部220は、過去の計測結果をRAMなどに保存しておき、過去の計測結果から現在の計測結果と最も近い記録を選び、当該記録により、現在の計測結果を補完してもよい。過去の計測結果を用いて補完することで、入力レートの計測の頻度を下げられるので、入力レートの計測のための負荷を低減できる。例えば、入力レート計測部220は、次のように記録の選択を行う。

40

【0098】

図11は、バッファデータ量計測テーブルの例を示す図である。

バッファデータ量計測テーブル221は、入力レート計測部220により生成され、メッセージサーバ200のRAMに保持される。バッファデータ量計測テーブル221は、今回の入力レートの計測に用いられるテーブルである。入力レート計測部220は、過去の入力レートの計測に用いたバッファデータ量計測テーブルをRAM上に保持しておく。

50

## 【 0 0 9 9 】

バッファデータ量計測テーブル 2 2 1 は、前回チェックポイントからの経過時間および入力バッファ部のデータ量の項目を含む。前回チェックポイントからの経過時間の項目には、前回チェックポイントからの経過時間が登録される。入力バッファ部のデータ量の項目には、当該経過時間において計測されたデータ量が登録される。データ量はデータサイズと呼ばれてもよい。時刻は、前回チェックポイントからの経過時間によって表されるものとする。例えば、時刻  $t_x$  は、前回チェックポイントの完了時刻から時間  $t_x$  が経過した時刻を指す。

## 【 0 1 0 0 】

今回の入力レートの計測のために、入力レート計測部 2 2 0 は、時刻  $t_0 \sim t_n$  までデータ量を計測したとする。過去の各時刻におけるデータ量の記録それぞれについて、時刻  $t_0 \sim t_n$  までの記録と比較して距離を求め、その距離が最も小さい記録を選択する。入力レート計測部 2 2 0 は、距離を求めるために、まず該当の記録の時点列  $t_0, t_1, \dots, t_n$  と過去の記録の時点列  $t_0', t_1', \dots, t_n'$  を合わせた時点列  $T S$  を求める。時点列  $T S$  は、両時点列に属する全ての時刻を時間順に並べた列である。次に、入力レート計測部 2 2 0 は、当該記録について、時点列  $T S$  の各時刻  $t_k$  についてのデータ量  $l_k$  を求める。時刻  $t_k$  が記録にある時刻であれば記録されたデータ量を用い、記録にある時刻でなければ、時刻  $t_k$  の前後で最も近い時刻の値  $(t_a, l_a)$ 、 $(t_b, l_a)$  ( $t_a < t_k < t_b$ ) について、時刻  $t_k$  でのデータ量  $l_k$  を、例えば式 (3) により求める。

## 【 0 1 0 1 】

## 【数 3】

$$l_k = l_a + \frac{(l_b - l_a) \times (t_k - t_a)}{t_b - t_a} \quad (3)$$

## 【 0 1 0 2 】

入力レート計測部 2 2 0 は、過去の記録についても同様に、時点列  $T S$  の各時刻でのデータ量を求める。そして、入力レート計測部 2 2 0 は、距離  $d$  を式 (4) により求める。

## 【 0 1 0 3 】

## 【数 4】

$$d = \sum_i (l_i - l'_i)^2 \quad (4)$$

## 【 0 1 0 4 】

データ量  $l_i$  は、今回の記録について時刻  $t_i$  に対して求めたデータ量である。データ量  $l'_i$  は、過去の記録について時刻  $t'_i$  に対して求めたデータ量である。

入力レート計測部 2 2 0 は、入力バッファ部 2 1 0 のデータ量を計測するたびに入力レート  $R$  を再計算し、更新する。入力レート計測部 2 2 0 は、入力レート  $R$  を更新すると、更新後の入力レート  $R$  をチェックポイントスケジュール部 1 3 0 に通知する。

## 【 0 1 0 5 】

次に、スループット計測部 1 4 0 によるスループットの計測について説明する。

図 1 2 は、スループット計測テーブルの例を示す図である。

スループット計測部 1 4 0 は、ストリーム処理部 1 2 0 の単位時間当たりのデータ処理量を計測する。スループット計測部 1 4 0 は、スループットの計測に、例えばレイテンシマーカと呼ばれる特別なデータを用いる。

## 【 0 1 0 6 】

スループット計測部 1 4 0 は、ストリーム処理部 1 2 0 にデータ  $a$  が入力された時刻を、データ処理開始時刻  $t_s$  として記録する。スループット計測部 1 4 0 は、次にデータ  $b$  がストリーム処理部 1 2 0 に入力されると、データ  $b$  の次に、レイテンシマーカをストリ

10

20

30

40

50

ーム処理部 120 に入力する。ストリーム処理部 120 の各タスクは、レイテンシマーカを受け取ると、レイテンシマーカを直ちに後段タスクへ転送する。スループット計測部 140 は、レイテンシマーカがストリーム処理部 120 の最後尾のタスクを通過した時刻をデータ処理終了時刻  $t_e$  として記録する。スループット計測部 140 は、データ a からデータ b までの処理時間  $pt = t_e - t_s$  を求める。スループット計測部 140 は、データ a からデータ b までのデータ量総和  $pv$  を求める。スループット計測部 140 は、データ a からデータ b までのデータ量をストリーム処理部 120 から取得してもよく、ストリーム処理部 120 がデータ処理時にスループット計測部 140 に当該データ量を通知してもよい。

#### 【0107】

スループット計測部 140 は、処理時間  $pt$  およびデータ量総和  $pv$  の計測を、予め設定された時間間隔で定期的に、あるいは、ストリーム処理部 120 に入力されたデータ数が前回の計測時から予め設定された閾値を超えるたびに行う。スループット計測部 140 は、計測結果を、スループット計測テーブル 141 に記録する。

#### 【0108】

スループット計測テーブル 141 は、例えば記憶部 110 に格納される。スループット計測テーブル 141 は、データ処理開始時刻、データ処理終了時刻、データ処理時間およびデータ量総和の項目を含む。データ処理開始時刻の項目には、ストリーム処理部 120 によるデータ処理の開始時刻  $t_s$  が登録される。データ処理終了時刻の項目には、ストリーム処理部 120 によるデータ処理の終了時刻  $t_e$  が登録される。データ処理時間の項目には、ストリーム処理部 120 によるデータ処理時間  $pt$  が登録される。データ量総和の項目には、処理されたデータ量の総和  $pv$  が登録される。

#### 【0109】

スループット計測部 140 は、スループット計測テーブル 141 に基づいて、例えば式 (5) によりストリーム処理部 120 のスループット  $TH_p$  を求める。

#### 【0110】

【数 5】

$$TH_p = \frac{\sum_i (pv_i - pv_{ave})(pt_i - pt_{ave})}{\sum_i (pt_i - pt_{ave})^2} \quad (5)$$

#### 【0111】

ここで、 $pv_{ave}$  は、データ量総和  $pv_i$  の平均値である。 $pt_{ave}$  は、データ処理時間  $pt_i$  の平均値である。

スループット計測部 140 は、スループット計測テーブル 141 における新たなレコードを計測するたびに、スループット  $TH_p$  を再計算し、更新する。スループット計測部 140 は、スループット  $TH_p$  を更新すると、更新後のスループット  $TH_p$  を、チェックポイントスケジュール部 130 に通知する。

#### 【0112】

次に、ステート更新サイズ計測部 150 によるステート更新サイズの計測について説明する。

図 13 は、ステート更新サイズ計測テーブルの例を示す図である。

#### 【0113】

ステート更新サイズ計測部 150 は、ストリーム処理部 120 を構成する各タスクが更新したステートのデータサイズの総和  $uv$  を計測する。各タスクが更新したステートのデータサイズの総和を、ステート更新サイズと称する。例えば、ステート更新サイズ計測部 150 は、各タスクによるステートへの書き込み操作を監視し、更新サイズを累積することで、ステート更新サイズ  $uv$  を求める。

#### 【0114】

ステート更新サイズ計測部 150 は、ステート更新サイズの計測を、予め設定された時

10

20

30

40

50

間隔で定期的に行ってもよく、ストリーム処理部 120 に入力されたデータ数が前回の状態更新サイズの計測時から予め設定された閾値を超えるたびに行ってもよい。状態更新サイズ計測部 150 は、計測した状態更新サイズ  $uv$  を、状態更新サイズ計測テーブル 151 に記録する。

【0115】

状態更新サイズ計測テーブル 151 は、例えば記憶部 110 に格納される。状態更新サイズ計測テーブル 151 は、前回チェックポイントからの経過時間および状態更新サイズの項目を含む。前回チェックポイントからの経過時間の項目には、前回チェックポイントからの経過時間が登録される。状態更新サイズの項目には、当該経過時間で表される時刻で計測された状態更新サイズ  $uv$  が登録される。

10

【0116】

状態更新サイズ計測部 150 は、状態更新サイズ計測テーブル 151 に基づいて、例えば式(6)により時間に対する状態更新サイズ増加率  $D$  を求める。

【0117】

【数6】

$$D = \frac{\sum_i (t_i - t_{ave})(uv_i - uv_{ave})}{\sum_i (t_i - t_{ave})^2} \quad (6)$$

【0118】

ここで、 $t_{ave}$  は  $t_i$  の平均値である。 $uv_{ave}$  は  $uv_i$  の平均値である。

状態更新サイズ計測部 150 は、状態更新サイズ計測テーブル 151 における新たなレコードを計測するたびに、状態更新サイズ増加率  $D$  を再計算し、更新する。状態更新サイズ計測部 150 は、状態更新サイズ増加率  $D$  を更新すると、更新後の状態更新サイズ増加率  $D$  をチェックポイントスケジュール部 130 に通知する。

20

【0119】

なお、状態更新サイズ計測部 150 は、状態更新サイズ増加率  $D$  の過去の計測に用いた状態更新サイズ計測テーブルを保存しておき、入力レートの計測で述べた方法と同様の方法により、過去の計測結果で現時点までの計測結果を補完してもよい。

【0120】

次に、復元スループット計測部 160 による復元スループットの計測について説明する。

図14は、復元スループット計測テーブルの例を示す図である。

30

【0121】

復元スループット計測部 160 は、ストリーム処理部 120 が復旧処理を行うとき、復旧処理に用いられたバックアップサイズ  $bv$  と、状態をバックアップにより復元するためにかかった時間  $rt$  を計測する。復元スループット計測部 160 は、バックアップサイズ  $bv$  および時間  $rt$  を、ストリーム処理部 120 から取得してもよく、ストリーム処理部 120 が復旧処理を行うときに、復元スループット計測部 160 にバックアップサイズ  $bv$  および時間  $rt$  を通知してもよい。復元スループット計測部 160 は、バックアップサイズ  $bv$  および時間  $rt$  を復元スループット計測テーブル 161 に記録する。

40

【0122】

復元スループット計測テーブル 161 は、例えば記憶部 110 に格納される。復元スループット計測テーブル 161 は、バックアップサイズおよび復元時間の項目を含む。バックアップサイズの項目には、バックアップサイズ  $bv$  が登録される。復元時間の項目には、復元時間  $rt$  が登録される。

【0123】

復元スループット計測部 160 は、復元スループット計測テーブル 161 に基づいて、例えば式(7)により、単位時間当たりの復元データ量を復元スループット  $TH\_r$  として求める。

【0124】

50

【数 7】

$$TH\_r = \frac{\sum_i (rt_i - rt_{ave})(bv_i - bv_{ave})}{\sum_i (rt_i - rt_{ave})^2} \quad (7)$$

【0125】

ここで、 $rt_{ave}$  は  $rt_i$  の平均値である。 $bv_{ave}$  は  $bv_i$  の平均値である。

復元スループット計測部 160 は、復元スループット計測テーブル 161 における新たなレコードが計測されるたびに、復元スループット  $TH\_r$  を再計算し、更新する。復元スループット計測部 160 は、復元スループット  $TH\_r$  を更新すると、更新後の復元スループット  $TH\_r$  をチェックポイントスケジュール部 130 に通知する。

10

【0126】

次に、チェックポイント制御部 170 によるチェックポイントスループットの計測について説明する。

図 15 は、チェックポイントスループット計測テーブルの例を示す図である。

【0127】

チェックポイント制御部 170 は、チェックポイント開始時刻  $tcs$  を記録する。チェックポイント制御部 170 は、チェックポイント終了時、チェックポイント終了時刻  $tce$  を記録し、チェックポイント処理時間  $ct = tce - tcs$  を求める。また、チェックポイント制御部 170 は、チェックポイントにおけるバックアップサイズ  $cv$  をストリーム処理部 120 から取得する。チェックポイント制御部 170 は、チェックポイント開始時刻  $tcs$ 、チェックポイント終了時刻  $tce$ 、チェックポイント処理時間  $ct$  およびバックアップサイズ  $cv$  をチェックポイントスループット計測テーブル 171 に記録する。

20

【0128】

チェックポイントスループット計測テーブル 171 は、例えば記憶部 110 に格納される。チェックポイントスループット計測テーブル 171 は、チェックポイント開始時刻、チェックポイント終了時刻、チェックポイント処理時間およびバックアップサイズの項目を含む。チェックポイント開始時刻の項目には、チェックポイント開始時刻  $tcs$  が登録される。チェックポイント終了時刻の項目には、チェックポイント終了時刻  $tce$  が登録される。チェックポイント処理時間の項目には、チェックポイント処理時間  $ct$  が登録される。バックアップサイズの項目には、バックアップサイズ  $cv$  が登録される。

30

【0129】

チェックポイント制御部 170 は、チェックポイントスループット計測テーブル 171 に基づいて、例えば式 (8) により、単位時間当たりのチェックポイント処理サイズをチェックポイントスループット  $TH\_cp$  として求める。単位時間当たりのチェックポイント処理サイズは、単位時間当たりのバックアップ処理サイズとも言える。

【0130】

【数 8】

$$TH\_cp = \frac{\sum_i (ct_i - ct_{ave})(cv_i - cv_{ave})}{\sum_i (ct_i - ct_{ave})^2} \quad (8)$$

40

【0131】

ここで、 $ct_{ave}$  は  $ct_i$  の平均値である。 $cv_{ave}$  は  $cv_i$  の平均値である。

チェックポイント制御部 170 は、チェックポイントスループット計測テーブル 171 における新たなレコードが計測されるたびに、チェックポイントスループット  $TH\_cp$  を再計算し、更新する。チェックポイント制御部 170 は、チェックポイントスループット  $TH\_cp$  を更新すると、更新後のチェックポイントスループット  $TH\_cp$  および完了したチェックポイントのバックアップサイズ  $cv$  をチェックポイントスケジュール部 130 に通知する。

50

## 【 0 1 3 2 】

次に、ノード 1 0 0 の処理手順を説明する。

図 1 6 は、チェックポイント開始時刻決定例を示すフローチャートである。

( S 1 0 ) チェックポイントスケジュール部 1 3 0 は、入力レート、ストリーム処理部 1 2 0 のストリーム処理のスループット、ステート更新サイズ増加率、復元スループットおよびチェックポイントスループットの通知を受ける。入力レートは、入力レート計測部 2 2 0 からチェックポイントスケジュール部 1 3 0 に通知される。ストリーム処理のスループットは、スループット計測部 1 4 0 からチェックポイントスケジュール部 1 3 0 に通知される。ステート更新サイズ増加率は、ステート更新サイズ計測部 1 5 0 からチェックポイントスケジュール部 1 3 0 に通知される。復元スループットは、復元スループット計測部 1 6 0 からチェックポイントスケジュール部 1 3 0 に通知される。チェックポイントスループットは、チェックポイント制御部 1 7 0 からチェックポイントスケジュール部 1 3 0 に通知される。なお、チェックポイント制御部 1 7 0 は、チェックポイントスループットとともにバックアップサイズもチェックポイントスケジュール部 1 3 0 に通知する。入力レート、ストリーム処理のスループット、ステート更新サイズ増加率、復元スループットおよびチェックポイントスループットそれぞれがチェックポイントスケジュール部 1 3 0 に通知されるタイミングは同じでもよいし、異なってもよい。なお、チェックポイントスケジュール部 1 3 0 は、入力レート、ストリーム処理のスループット、ステート更新サイズ増加率、復元スループットおよびチェックポイントスループットのうちの一部の通知を受けずに、ステップ S 1 1 に進むことがある。

10

20

## 【 0 1 3 3 】

( S 1 1 ) チェックポイントスケジュール部 1 3 0 は、次のチェックポイント開始時刻を決定する。具体的には、チェックポイントスケジュール部 1 3 0 は、ステートのバックアップ処理や復旧処理の所要時間を考慮して、バッファオーバーフローが生じる前の適切なバックアップタイミングを計る。ステートのバックアップ処理の所要時間や復旧処理の所要時間は、入力レート、ストリーム処理のスループット、ステート更新サイズ、復元スループットおよびチェックポイントスループットに基づいて予測される。

## 【 0 1 3 4 】

( S 1 2 ) チェックポイントスケジュール部 1 3 0 は、ステップ S 1 1 で決定した次のチェックポイント開始時刻を、チェックポイント制御部 1 7 0 に通知する。そして、チェックポイント開始時刻決定の処理が終了する。

30

## 【 0 1 3 5 】

チェックポイントスケジュール部 1 3 0 は、上記の手順を予め定められた時間間隔などで繰り返し実行する。これにより、入力レート、ストリーム処理のスループットおよびステート更新サイズなどの変化に応じて、次のチェックポイント開始時刻を柔軟に決定できる。

## 【 0 1 3 6 】

チェックポイント制御部 1 7 0 は、チェックポイント開始時刻を受け付けると、チェックポイント開始時刻を検出するためにタイマーを開始させる。チェックポイント制御部 1 7 0 は、チェックポイント開始時刻に達したことを検出すると、ストリーム処理部 1 2 0 にチェックポイントを入力する。これにより、ステートのバックアップ処理が開始され、当該ステートのバックアップが取得される。

40

## 【 0 1 3 7 】

チェックポイントスケジュール部 1 3 0 は、決定済のチェックポイント開始時刻に達する前に、入力レート、単位時間当たりの状態データの更新量、ストリーム処理のスループット、復元スループットおよびチェックポイントスループットの少なくとも 1 つが更新されると、更新後の情報に基づきチェックポイント開始時刻を更新する。決定済のチェックポイント開始時刻に達する前に、更新後のチェックポイント開始時刻がチェックポイントスケジュール部 1 3 0 からチェックポイント制御部 1 7 0 に通知されることがある。その場合、チェックポイント制御部 1 7 0 は、タイマーを更新することで、更新後のチェック

50

ポイント開始時刻でチェックポイントが発行されるようにする。

【 0 1 3 8 】

次に、ステップ S 1 1 におけるチェックポイント開始時刻の決定例を説明する。

図 1 7 は、チェックポイント開始時刻の決定例を示す図である。

グラフ G 1 1 は、時間とバッファデータ量との関係を示す。グラフ G 1 1 の横軸は時間であり、縦軸は入力バッファ部 2 1 0 に保持されているデータサイズである。縦軸の  $L_{max}$  は、入力バッファ部 2 1 0 の上限サイズである。横軸の時間は、前回のチェックポイント終了時刻  $t_0$  を原点とし、時刻  $t_0$  からの経過時間を表す。各時刻は、時刻  $t_0$  からの経過時間によって表される。例えば、時刻  $t_x$  と言う場合、時刻  $t_0$  からの経過時間が  $t_x$  である時刻を指す。

10

【 0 1 3 9 】

チェックポイントスケジュール部 1 3 0 は、次のチェックポイント開始時刻  $t_{ncp}$  を、次の式 ( 9 ) ~ ( 1 3 ) により求める。

【 0 1 4 0 】

【数 9】

$$t_{ncp} = \frac{t_{rs} \times TH_{cp}}{TH_{cp} + D} \quad (9)$$

【 0 1 4 1 】

【数 1 0】

$$t_{rs} = t_{rc} - \frac{cv}{TH_{rs}} - \alpha \quad (10)$$

20

【 0 1 4 2 】

【数 1 1】

$$t_{rc} = \frac{t_{cpx} \times TH_p}{TH_p + R} \quad (11)$$

30

【 0 1 4 3 】

【数 1 2】

$$t_{cpx} = \frac{T_{over} \times TH_{cp}}{TH_{cp} + D} \quad (12)$$

【 0 1 4 4 】

【数 1 3】

$$T_{over} = \frac{L_{max}}{R} \quad (13)$$

40

【 0 1 4 5 】

ここで、 $TH_{cp}$  は、チェックポイント制御部 1 7 0 から通知された最新のチェックポイントスループットである。D は、ステート更新サイズ計測部 1 5 0 から通知された最新のステート更新サイズ増加率である。 $TH_{rs}$  は、復元スループット計測部 1 6 0 から通知された最新の復元スループットである。  $\alpha$  は、予め設定されたマージンである。例えば、  $\alpha$  は、システム停止検知から復旧開始までにかかる時間に応じて予め見積もられた値である。 $TH_p$  は、スループット計測部 1 4 0 から通知されたストリーム処理の最新

50

のスループットである。Rは、入力レート計測部220から通知された最新の入力レートである。

【0146】

例えば、時刻 $t_i$ で入力レート計測部220がチェックポイントスケジュール部130に入力レートRを通知したとする。グラフG11の折れ線G11aは、時刻 $t_0$ から時刻 $t_i$ までの入力バッファ部210内のデータサイズの変化を示す。グラフG11の直線G11bの傾きは、折れ線G11aのデータに基づいて計算された入力レートRである。

【0147】

時刻 $T_{over}$ は、入力バッファ部210に保持されているデータサイズが上限サイズ $L_{max}$ に到達する予測時点である。時刻 $T_{over}$ までにチェックポイント処理が完了すれば、入力バッファ部210に空きができるため、バッファオーバーフローを防ぐことができる。時間 $t_1$ は、時刻 $T_{over}$ にチェックポイント処理が完了する場合のチェックポイント処理の所要時間を示す。また、時刻 $t_{cp}$ は、その場合のチェックポイント開始時刻を示す。 $t_1 = (D \times t_{cp} / TH_{cp})$ である。 $D \times t_{cp}$ は、時刻 $t_{cp}$ におけるバックアップ対象のステートのデータサイズである。

10

【0148】

システム障害などによる復旧処理がなければ時刻 $t_{cp}$ を次のチェックポイント開始時刻としてよい。しかし、実際は復旧処理が発生することを考慮する必要がある。復旧処理は、ステートの復元処理およびストリームの再計算処理であるので、オーバーフローを防ぐことができる復旧処理の後端の開始時刻は、時刻 $t_{rs}$ となる。時間 $t_2$ は、ストリームの再計算処理の所要時間である。時間 $t_3$ は、復元処理の所要時間である。

20

【0149】

復旧処理の開始が時刻 $t_{rs}$ 以前であれば、ステート復元処理、ストリーム再計算処理、および、チェックポイント処理を、オーバーフローの予測時刻 $T_{over}$ 前に完了できる。

【0150】

時刻 $t_{rs}$ を開始時刻とするステートの復元処理の所要時間 $t_3$ は、前回のチェックポイントのバックアップサイズ $c_v$ と、復元スループット $TH_{rs}$ から計算される。ストリーム再計算処理の所要時間 $t_2$ は、復元処理完了時刻 $t_{rc}$ での入力バッファ部210内のデータサイズの見積もり量 $(= R \times t_{rc})$ と、スループット $TH_p$ から計算される。具体的には、 $t_2 = (R \times t_{rc} / TH_p)$ である。 $R \times t_{rc}$ は、時刻 $t_{rc}$ において入力バッファ部210に格納されているデータのデータサイズである。 $t_3 = c_v / TH_{rs}$ である。

30

【0151】

復旧処理が時刻 $t_{rs}$ 以降に起きないことを保証するためには、時刻 $t_{rs}$ までにチェックポイントを完了する必要がある。よって、チェックポイントスケジュール部130は、チェックポイントスループット $TH_{cp}$ とステート更新サイズ増加率Dを用いて、チェックポイント処理時間 $t_4$ を見積もり、次のチェックポイント開始時刻 $t_{ncp}$ を求める。 $t_4 = (D \times t_{ncp} / TH_{cp})$ である。 $D \times t_{ncp}$ は、時刻 $t_{ncp}$ におけるバックアップ対象のステートのデータサイズである。

40

【0152】

チェックポイント制御部170は、時刻 $t_{ncp}$ で次のチェックポイントをストリーム処理部120に入力することで、その後、仮に復旧処理が発生したとしても、入力バッファ部210がオーバーフローを起こすことを防げる。

【0153】

なお、チェックポイントスケジュール部130からチェックポイント制御部170に通知された時刻 $t_{ncp}$ が現時刻 $t_i$ よりも前である場合、チェックポイント制御部170は、直ちにチェックポイントを発行し、利用者などにアラートを通知してもよい。

【0154】

また、次のチェックポイント開始時刻は式(14)で表される時刻 $t_{ncp}'$ でもよい

50

【 0 1 5 5 】

【 数 1 4 】

$$t_{ncp}' = t_{ncp} - \beta \left( \frac{L_{max}}{R} - t_{ncp} \right) \quad (14)$$

【 0 1 5 6 】

ここで、 $\beta$  は  $0 < \beta < 1$  の実数であり、予め設定される。この場合、 $\beta$  が大きいほど次のチェックポイント開始時刻は早くなり、復旧処理による入力バッファ部 2 1 0 のバッファオーバーフローの可能性を更に低減することができる。

10

【 0 1 5 7 】

なお、チェックポイントスケジュール部 1 3 0 は、予め設定された時間間隔で、入力レート、ストリーム処理部 1 2 0 のスループット、ステート更新サイズ増加率、復元スループットおよびチェックポイントスループットを取得するようにしてもよい。次に、この場合のチェックポイント開始時刻決定の手順を説明する。

【 0 1 5 8 】

図 1 8 は、チェックポイント開始時刻決定の他の例を示すフローチャートである。

( S 2 0 ) チェックポイントスケジュール部 1 3 0 は、指定時間待機する。指定時間は予め設定される。

20

【 0 1 5 9 】

( S 2 1 ) チェックポイントスケジュール部 1 3 0 は、入力レート計測部 2 2 0、スループット計測部 1 4 0、ステート更新サイズ計測部 1 5 0、復元スループット計測部 1 6 0 およびチェックポイント制御部 1 7 0 から、それぞれ入力レート  $R$ 、スループット  $TH_{p}$ 、ステート更新サイズ増加率  $D$ 、復元スループット  $TH_{rs}$  およびチェックポイントスループット  $TH_{cp}$  を取得する。

【 0 1 6 0 】

( S 2 2 ) チェックポイントスケジュール部 1 3 0 は、入力レート  $R$ 、スループット  $TH_{p}$ 、ステート更新サイズ増加率  $D$ 、復元スループット  $TH_{rs}$  およびチェックポイントスループット  $TH_{cp}$  に基づいて、次のチェックポイント開始時刻を決定する。

30

【 0 1 6 1 】

( S 2 3 ) チェックポイントスケジュール部 1 3 0 は、ステップ S 2 2 で決定した次のチェックポイント開始時刻を、チェックポイント制御部 1 7 0 に通知する。そして、ステップ S 2 0 に処理が進む。

【 0 1 6 2 】

なお、ステップ S 2 2 における次のチェックポイント開始時刻の決定およびステップ S 2 3 における次のチェックポイント開始時刻の通知は、それぞれ前述のステップ S 1 1、S 1 2 と同様の処理となる。

【 0 1 6 3 】

図 1 8 の手順によっても、図 1 6 の手順と同様に、ノード 1 0 0 は、チェックポイント開始時刻を適切に決定できる。

40

次に、第 2 の実施の形態の情報処理システムの他の例を説明する。図 2、図 4 の説明では 1 つのメッセージサーバ 2 0 0 を示したが、メッセージサーバ 2 0 0 は複数でもよい。

【 0 1 6 4 】

具体的には、チェックポイントスケジュール部 1 3 0 は、ストリーム処理に用いられるデータが格納される複数の入力バッファ部それぞれに対してチェックポイント開始時刻の候補時刻を計算してもよい。そして、チェックポイントスケジュール部 1 3 0 は、計算した複数の候補時刻のうち、最も早い候補時刻をチェックポイント開始時刻として決定してもよい。

【 0 1 6 5 】

50

図19は、情報処理システムの他の例を示す図である。

図19に例示される情報処理システムは、メッセージサーバ200に加えて、メッセージサーバ200a, 200bを有する。メッセージサーバ200aは、入力バッファ部210aを有する。メッセージサーバ200bは、入力バッファ部210bを有する。メッセージサーバ200a, 200bは、メッセージサーバ200と同様に、それぞれが入力レート計測部および入力バッファ管理部を有するが、図19では図示を省略している。

【0166】

入力バッファ部210, 210a, 210bには、エッジ装置300, 300a, 300bから送信されたデータが格納される。ストリーム処理部120は、入力バッファ部210, 210a, 210bに格納されたデータを取得して、ストリーム処理を行う。このように、ストリーム処理部120に対して、複数の入力バッファ部が設けられてもよい。

10

【0167】

この場合、チェックポイントスケジュール部130は、メッセージサーバ200, 200a, 200bのそれぞれから、入力バッファ部210, 210a, 210bに対する入力レートを取得する。そして、チェックポイントスケジュール部130は、入力バッファ部210, 210a, 210bのそれぞれの入力レートをを用いて、上記で説明した方法により次のバックアップ処理の開始時刻(すなわち、チェックポイント開始時刻)を求め、最も早い時刻を採用する。これにより、入力バッファ部が複数あっても、各入力バッファ部におけるバッファオーバーフローの発生を防ぎつつ、バックアップ処理に伴う負荷を低減できる。

20

【0168】

ところで、第2の実施の形態で例示したチェックポイントによるバックアップ処理のコストは高い。チェックポイントによるバックアップ処理では、RAM上にあるステータデータをコピーし、圧縮して、スナップショット記憶部190にファイルやDBのレコードとして書き出すため、CPU101やIO(Input/Output)リソースを消費する。また、バックアップの保存先は、障害対策の観点からネットワーク20に接続された、ノード100とは物理的に異なる外部ストレージ装置であることもあり、バックアップ処理に伴いネットワークトラフィックが生じることもある。そのため、チェックポイントを短い時間間隔で発行すると、バックアップ処理の負荷のため、ノード100の本来のストリーム処理のパフォーマンス(スループットやレイテンシ)が悪化する可能性がある。

30

【0169】

一方、チェックポイントの時間間隔を長くすると、入力バッファ部210に保持しなければならないデータ量が大きくなり、復旧時に入力バッファ部210がオーバーフローを起こしてしまう可能性が高くなる。チェックポイント以後のデータは復旧に備え、次のチェックポイントが完了する(次のチェックポイントがパイプラインの最後段まで流れる)まで入力バッファ部210で保持しておく必要がある。なぜなら、復旧では、パイプラインの各タスクに対して復元されるステータは前回チェックポイント時のステータであるため、チェックポイント後のデータからストリーム処理をやり直す(再計算する)必要があるためである。再計算の間も、入力バッファ部210にはデータが入力され続けるので、入力バッファ部210がオーバーフローを起こす前に、再計算がデータ入力に追いつくことが求められる。チェックポイントの時間間隔が長いほど、入力バッファ部に保持すべきデータ量、すなわち、再計算するデータ量が増え、再計算がデータ入力に追いつかずに、入力バッファ部210がオーバーフローを起こす可能性が高くなる。

40

【0170】

チェックポイントのタイミングを、例えば、定期的な時間間隔として、ユーザにより手動で設定することがある。しかし、チェックポイントの時間間隔が短すぎる場合、ストリーム処理の性能低下を防ぐためには、より多くのCPUやネットワークリソースを要することになる。逆に、チェックポイントの時間間隔が長すぎる場合、入力バッファ部210のオーバーフローを防ぐために、入力バッファ部210により多くの記憶資源を要することになる。何れの場合も、システムのハードウェアのコストを上げる要因になる。更に、

50

近年、I o T (Internet of Things) デバイスやモバイルサービスの普及により、入力バッファ部 2 1 0 に入力されるデータの頻度は、実世界の状況に依存して変化している。例えば、車両データ (位置やスピードなど) のイベントメッセージの収集の頻度は、曜日や時間帯によって変化する。そのため、チェックポイントの最適なタイミングを予め求めておくことは難しい。

【 0 1 7 1 】

そこで、ノード 1 0 0 は、入力レート、ストリーム処理のスループット、復元スループットおよびチェックポイントスループットに基づいて、次のチェックポイントの時刻を動的に自動調整する。これにより、バッファオーバーフローを防ぎながら、チェックポイント処理コストを低減することができる。チェックポイント処理コストの低減により、ノード 1 0 0 によるストリーム処理の性能低下を抑えられる。また、これらの利点を、システムコストを上げずに、例えばハードウェア増強を伴わずに実現できる。

10

【 0 1 7 2 】

なお、入力バッファ部 2 1 0 をメッセージサーバ 2 0 0 に設けるものとしたが、入力バッファ部 2 1 0、入力レート計測部 2 2 0 および入力バッファ管理部 2 3 0 をノード 1 0 0 に設けてもよい。例えば、ノード 1 0 0 を複数台設け、各ノードで同一のデータストリームを保持しておき、何れか 1 つのノードを運用系、残りのノードを待機系とする。運用系のノードは、ステートのバックアップを外部ストレージに取得する。すると、運用系のノードが障害でダウンしても、待機系のノードによって、外部ストレージにバックアップされたステートを復元し、当該待機系のノードのバッファに保持されるデータストリームによりシステムの運用を継続することができる。

20

【 0 1 7 3 】

また、第 1 の実施の形態の情報処理は、処理装置 1 4 にプログラムを実行させることで実現できる。第 2 の実施の形態の情報処理は、CPU 1 0 1 にプログラムを実行させることで実現できる。プログラムはコンピュータ読み取り可能な記録媒体 2 3 に記録できる。

【 0 1 7 4 】

例えば、プログラムを記録した記録媒体 2 3 を配布することで、プログラムを流通させることができる。また、プログラムを他のコンピュータに格納しておき、ネットワーク経由でプログラムを配布してもよい。コンピュータは、例えば、記録媒体 2 3 に記録されたプログラムまたは他のコンピュータから受信したプログラムを、RAM 1 0 2 や HDD 1 0 3 などの記憶装置に格納し (インストールし)、当該記憶装置からプログラムを読み込んで実行してもよい。

30

【 0 1 7 5 】

以上の第 1 , 第 2 の実施の形態を含む実施形態に関し、更に以下の付記を開示する。

( 付記 1 ) バッファに格納された入力データに対して実行される処理に対応する状態データを記憶する第 1 の記憶装置と、

前記バッファに対する前記入力データの入力レートから前記バッファのオーバーフローが生じる第 1 の時刻を計算し、前記入力レートと単位時間当たりの前記状態データの更新量と前記第 1 の記憶装置から第 2 の記憶装置への前記状態データのバックアップ処理の第 1 のスループットと前記第 2 の記憶装置から前記第 1 の記憶装置への前記状態データの復元処理の第 2 のスループットと前記処理の第 3 のスループットとに基づいて、前記状態データの前記バックアップ処理を開始する第 2 の時刻であって、前記第 1 の時刻よりも前の前記第 2 の時刻を決定する処理装置と、

40

を有する情報処理システム。

【 0 1 7 6 】

( 付記 2 ) 前記処理装置は、前記第 2 の時刻に達する前に、前記入力レート、単位時間当たりの前記状態データの更新量、前記第 1 のスループット、前記第 2 のスループットおよび前記第 3 のスループットの少なくとも 1 つが更新されると、前記第 2 の時刻を更新する、付記 1 記載の情報処理システム。

【 0 1 7 7 】

50

(付記 3) 前記処理装置は、前記入力レートを定期的に取得する、付記 1 記載の情報処理システム。

(付記 4) 前記処理装置は、前記バッファに格納されているデータサイズが前記バッファの上限サイズに対して所定割合だけ増加するたびに、前記入力レートを取得する、付記 1 記載の情報処理システム。

【0178】

(付記 5) 前記処理装置は、前回の前記入力レートを取得した時刻から、所定のデータサイズを前回の前記入力レートで割った時間が経過すると、前記入力レートを取得する、付記 1 記載の情報処理システム。

【0179】

(付記 6) 前記処理装置は、各時刻において前記バッファに格納されているデータサイズに基づいて前記入力レートを取得し、前記データサイズが取得されていない時刻の前記データサイズを、過去に取得済の前記データサイズに基づいて補完する、付記 1 記載の情報処理システム。

【0180】

(付記 7) 前記処理装置は、前記第 3 のスループットおよび単位時間当たりの前記状態データの更新量を定期的に取り得る、付記 1 記載の情報処理システム。

(付記 8) 前記処理装置は、前記第 3 のスループットおよび単位時間当たりの前記状態データの更新量の計測を、前回の前記計測後から前記処理に入力された前記入力データの数が増加したときに行う、付記 1 記載の情報処理システム。

【0181】

(付記 9) 前記処理装置は、各時刻における前記状態データの更新により更新された部分のサイズに基づいて単位時間当たりの前記状態データの更新量を取得し、前記更新された部分のサイズが取得されていない時刻の当該サイズを、過去に取得済の当該サイズに基づいて補完する、付記 1 記載の情報処理システム。

【0182】

(付記 10) 前記処理装置は、前記第 2 の時刻が現時刻よりも前の場合、前記状態データのバックアップ処理を直ちに開始し、アラートを通知する、付記 1 記載の情報処理システム。

【0183】

(付記 11) 前記処理装置は、  
単位時間当たりの前記状態データの更新量と前記第 1 のスループットとに基づいて、前記第 1 の時刻に前記バックアップ処理を完了させる場合の前記バックアップ処理の第 1 の開始時刻を計算し、

前記入力レートと前記第 2 のスループットと前記第 3 のスループットとに基づいて、前記第 1 の開始時刻に、前記状態データのバックアップによる前記第 1 の記憶装置の前記状態データの復旧を完了させる場合の前記復元処理の第 2 の開始時刻を計算し、

単位時間当たりの前記状態データの更新量と前記第 1 のスループットとに基づいて、前記第 2 の開始時刻に前記バックアップ処理を完了させる場合の前記バックアップ処理の第 3 の開始時刻を計算し、

前記第 3 の開始時刻または前記第 3 の開始時刻よりも所定時間前の時刻を、前記第 2 の時刻とする、

付記 1 記載の情報処理システム。

【0184】

(付記 12) 前記処理装置は、前記第 2 の時刻で前記バックアップ処理を開始し、前記バックアップ処理を終了すると、前記バッファに格納された前記入力データのうち、前記第 2 の時刻よりも前に前記処理に入力済である前記入力データを前記バッファから削除する、付記 1 記載の情報処理システム。

【0185】

(付記 13) 前記処理装置は、前記処理に用いられる前記入力データが格納される複

10

20

30

40

50

数の前記バッファそれぞれに対して前記第 2 の時刻の候補時刻を計算し、計算した複数の前記候補時刻のうち、最も早い前記候補時刻を前記第 2 の時刻とする、付記 1 記載の情報処理システム。

【0186】

(付記 14) 前記処理は、前記入力データを処理する複数のタスクを含むストリーム処理である、付記 1 記載の情報処理システム。

(付記 15) コンピュータが、

入力データを格納するバッファに対する前記入力データの入力レートから前記バッファのオーバーフローが生じる第 1 の時刻を計算し、

前記入力レートと前記入力データに対して実行される処理に対応する状態データの単位時間当たりの更新量と前記状態データを記憶する第 1 の記憶装置から第 2 の記憶装置への前記状態データのバックアップ処理の第 1 のスループットと前記第 2 の記憶装置から前記第 1 の記憶装置への前記状態データの復元処理の第 2 のスループットと前記処理の第 3 のスループットとに基づいて、前記状態データのバックアップ処理を開始する第 2 の時刻であって、前記第 1 の時刻よりも前の前記第 2 の時刻を決定する、

情報処理方法。

【0187】

(付記 16) コンピュータに、

入力データを格納するバッファに対する前記入力データの入力レートから前記バッファのオーバーフローが生じる第 1 の時刻を計算し、

前記入力レートと前記入力データに対して実行される処理に対応する状態データの単位時間当たりの更新量と前記状態データを記憶する第 1 の記憶装置から第 2 の記憶装置への前記状態データのバックアップ処理の第 1 のスループットと前記第 2 の記憶装置から前記第 1 の記憶装置への前記状態データの復元処理の第 2 のスループットと前記処理の第 3 のスループットとに基づいて、前記状態データのバックアップ処理を開始する第 2 の時刻であって、前記第 1 の時刻よりも前の前記第 2 の時刻を決定する、

処理を実行させる情報処理プログラム。

【符号の説明】

【0188】

10 情報処理システム

11 バッファ

12 第 1 の記憶装置

13 第 2 の記憶装置

14 処理装置

14 a 処理実行部

14 b バックアップ時刻決定部

14 c バックアップ実行部

G1 グラフ

R 入力レート

t 1 , t 4 バックアップ所要時間

t 2 再計算所要時間

t 3 復元所要時間

10

20

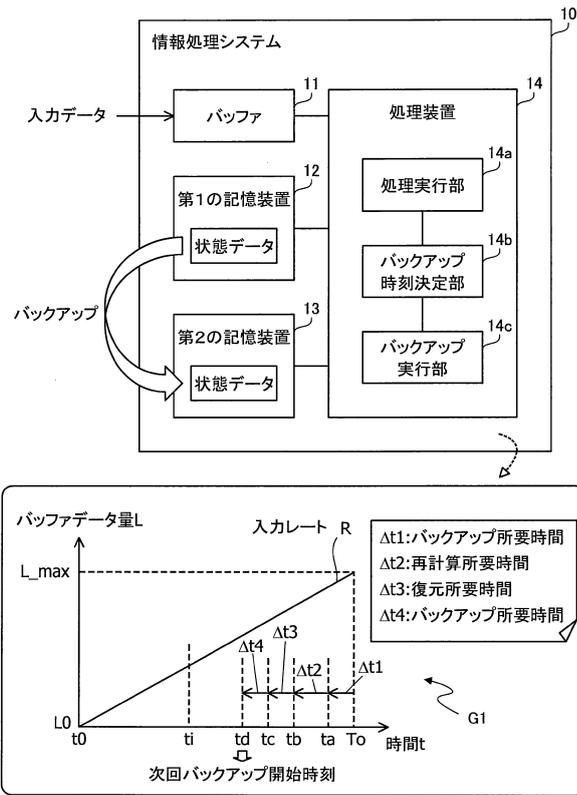
30

40

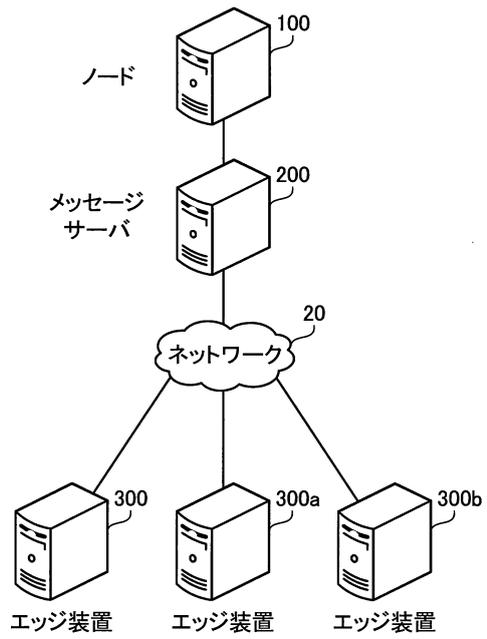
50

【図面】

【図1】



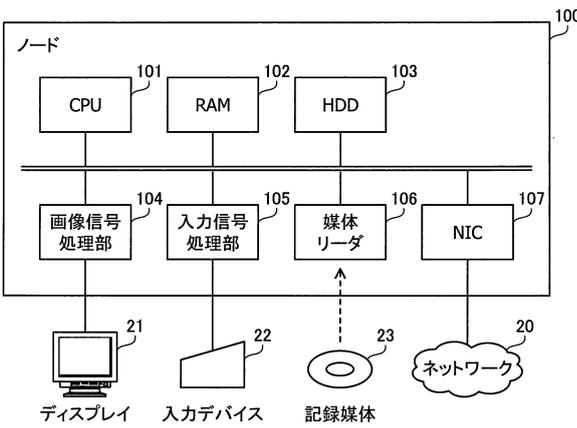
【図2】



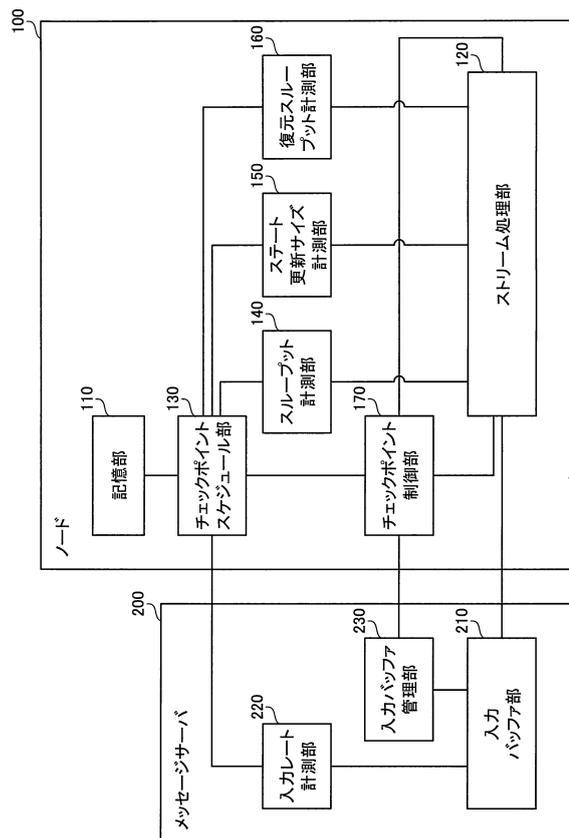
10

20

【図3】



【図4】

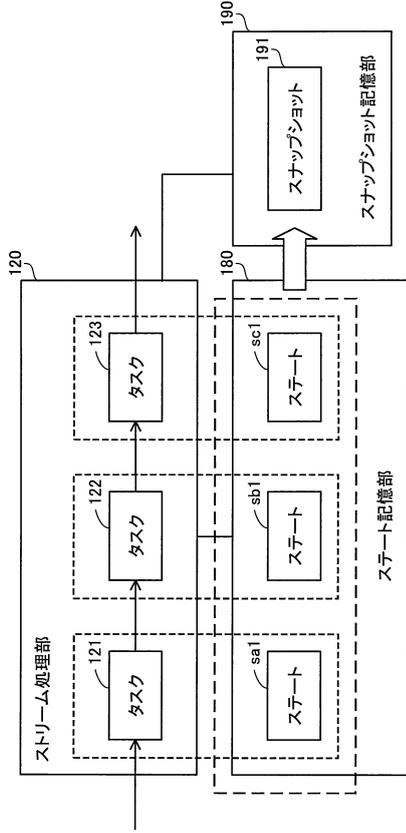


30

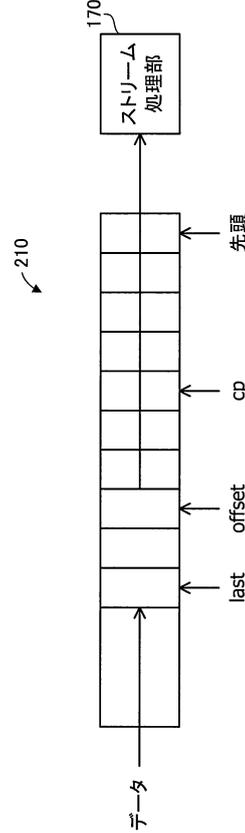
40

50

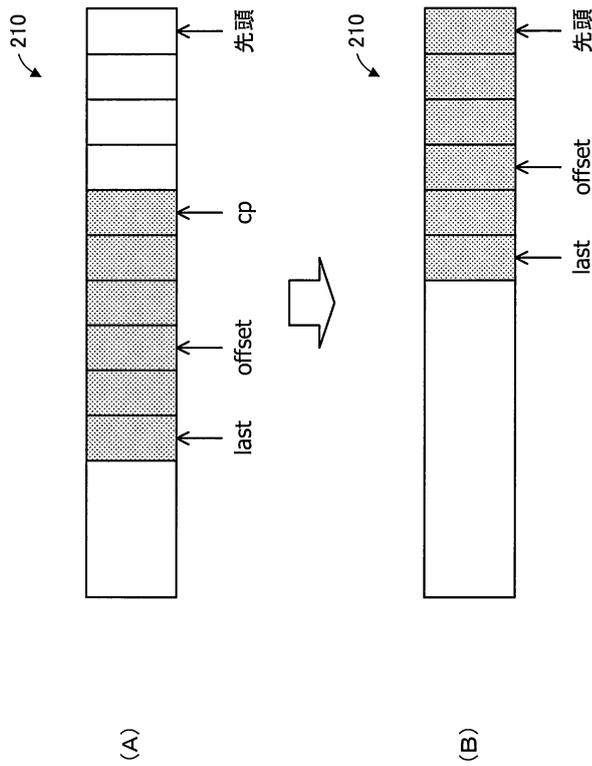
【図 5】



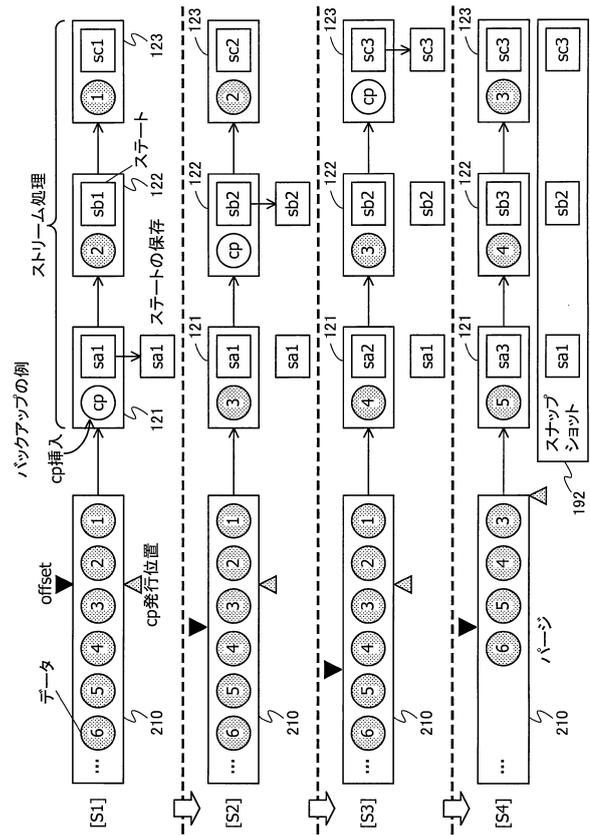
【図 6】



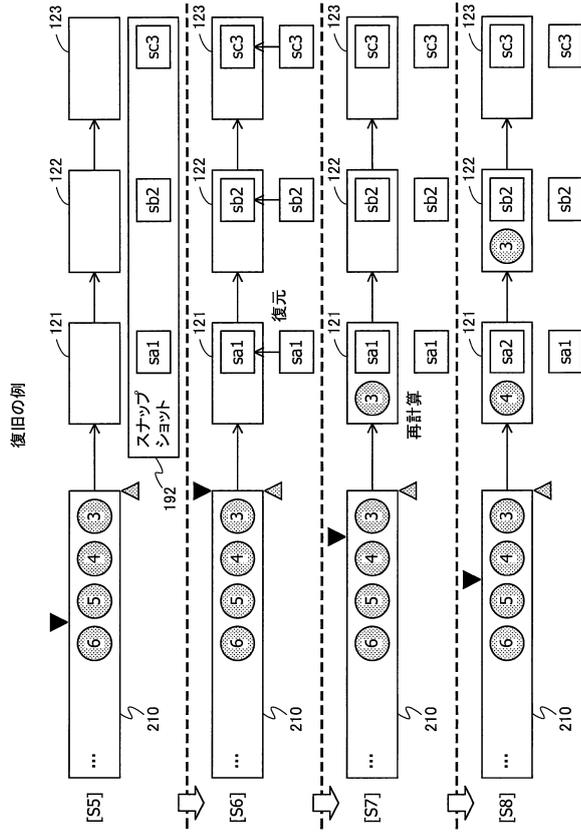
【図 7】



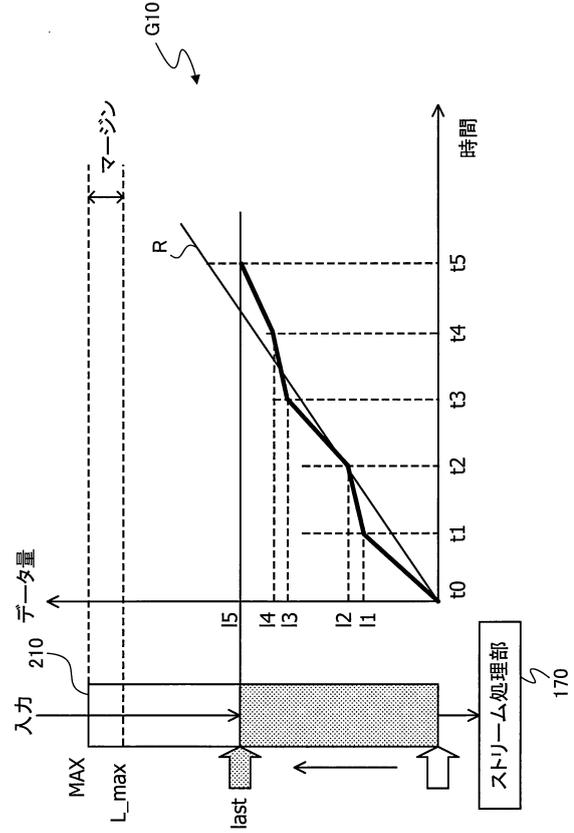
【図 8】



【図 9】



【図 10】



【図 11】

前回チェックポイントからの経過時間	入力バッファ部のデータ量
t0	l0
t1	l1
t2	l2
...	...

【図 12】

データ処理開始時刻	データ処理終了時刻	データ処理時間	データ量総和
ts1	te1	pt1	pv1
ts2	te2	pt2	pv2
...	...	...	...

10

20

30

40

50

【図 1 3】

状態更新サイズ計測テーブル	
前回チェックポイントからの経過時間	状態更新サイズ
t0	uv0
t1	uv1
t2	uv2
...	...

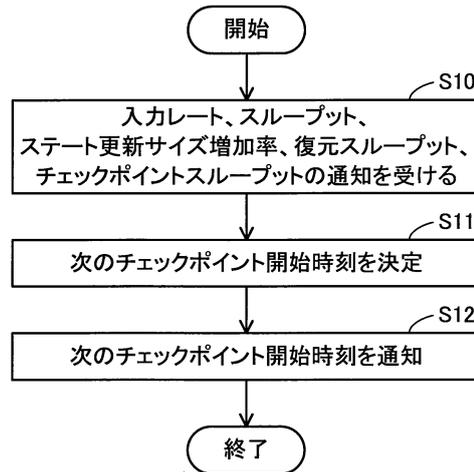
【図 1 4】

復元スループット計測テーブル	
バックアップサイズ	復元時間
bv0	rt0
bv1	rt1
bv2	rt2
...	...

【図 1 5】

チェックポイントスループット計測テーブル			
チェックポイント開始時刻	チェックポイント終了時刻	チェックポイント処理時間	バックアップサイズ
tcs1	tce1	ct1	cv1
tcs2	tce2	ct2	cv2
...	...	...	...

【図 1 6】



10

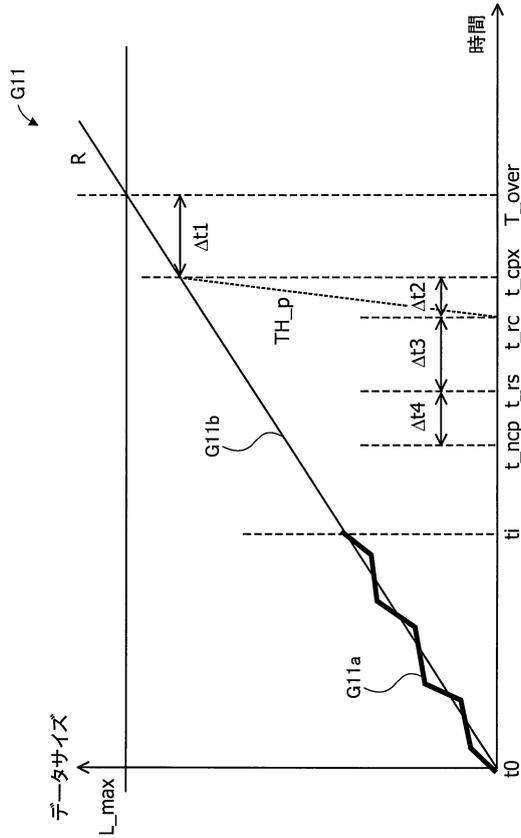
20

30

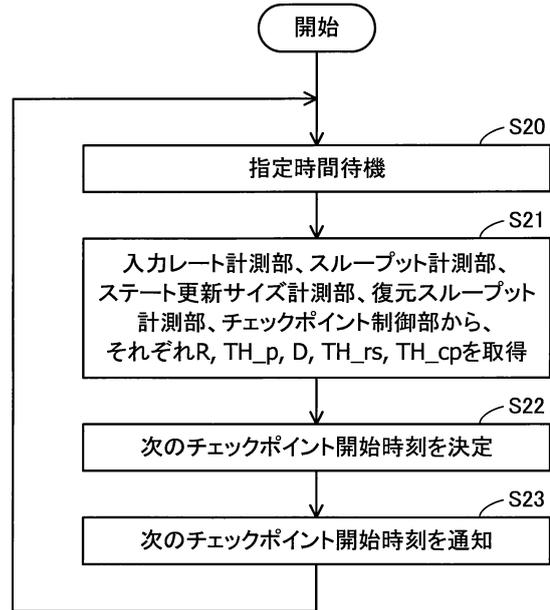
40

50

【図17】



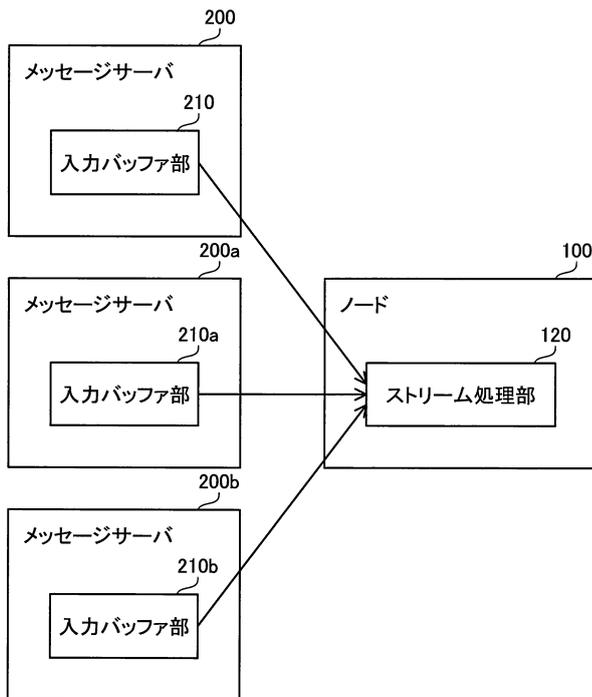
【図18】



10

20

【図19】



30

40

50

## フロントページの続き

- (72)発明者 西口 直樹  
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 山岡 久俊  
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 板倉 宏太  
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 笠松 大佑  
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 大西 隆史  
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 松本 達郎  
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 松井 一樹  
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- 審査官 田中 啓介
- (56)参考文献 特開2009-205548(JP,A)  
特開2008-146577(JP,A)  
特表2008-500627(JP,A)
- (58)調査した分野 (Int.Cl., DB名)  
G06F3/06-3/08  
G06F9/455-9/54  
G06F11/07、11/14  
G06F11/28-11/36  
G06F13/10-13/14