

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4161998号
(P4161998)

(45) 発行日 平成20年10月8日(2008.10.8)

(24) 登録日 平成20年8月1日(2008.8.1)

(51) Int.Cl.		F I			
G06F 13/00	(2006.01)	G06F 13/00	520C		
G06F 11/20	(2006.01)	G06F 11/20	310A		
		G06F 13/00	357Z		

請求項の数 18 (全 23 頁)

(21) 出願番号	特願2005-314537 (P2005-314537)	(73) 特許権者	000004237
(22) 出願日	平成17年10月28日(2005.10.28)		日本電気株式会社
(65) 公開番号	特開2006-309701 (P2006-309701A)		東京都港区芝五丁目7番1号
(43) 公開日	平成18年11月9日(2006.11.9)	(74) 代理人	100109313
審査請求日	平成17年10月28日(2005.10.28)		弁理士 机 昌彦
(31) 優先権主張番号	特願2005-92080 (P2005-92080)	(74) 代理人	100121290
(32) 優先日	平成17年3月28日(2005.3.28)		弁理士 木村 明隆
(33) 優先権主張国	日本国(JP)	(74) 代理人	100111637
			弁理士 谷澤 靖久
(出願人による申告)平成17年度、総務省委託研究「ユビキタスネットワーク制御・管理技術の研究開発」によるもの、産業活力再生特別措置法第30条の適用を受ける特許出願		(72) 発明者	佐藤 正 東京都港区芝五丁目7番1号 日本電気株式会社内
		(72) 発明者	磯山 和彦 東京都港区芝五丁目7番1号 日本電気株式会社内

最終頁に続く

(54) 【発明の名称】 負荷分散振り分けシステム、イベント処理分散制御装置並びにイベント処理分散制御プログラム

(57) 【特許請求の範囲】

【請求項1】

サーバに具備されたアプリケーションにイベント処理結果を通知する複数の処理サーバと、

前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト振り分け手段と、

前記複数の処理サーバにイベントの内容を示すコンテキストに基づく処理を示す処理ルールを設定し、前記複数の処理サーバが前記処理ルールを実行するために必要なイベントが適切に分配されるように振り分けルールを前記コンテキスト振り分け手段に記憶させるイベント処理分散制御手段と、を有し、

前記イベント処理分散制御手段は、

前記処理サーバごとに、前記処理ルールに含まれる、処理すべきイベントの条件を示すイベント条件を少なくとも記録する処理ルール振り分け記録手段と、

アプリケーションから処理ルールを受信する処理ルール受信手段と、

前記処理ルール受信手段が新規に処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の最小値に対し、あらかじめ定められた一定範囲以内の数の処理ルールを割り当てられた処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる振り分けルール決定手段と、を有する、

負荷分散振り分けシステム。

【請求項 2】

請求項 1 に記載の負荷分散振り分けシステムであって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、請求項 1 に記載の負荷分散振り分けシステム。

【請求項 3】

請求項 1 に記載の負荷分散振り分けシステムであって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、請求項 1 に記載の負荷分散振り分けシステム。

10

【請求項 4】

請求項 1 に記載の負荷分散振り分けシステムであって、
前記処理ルールごとに重み付け係数を備え、
前記振り分けルール決定手段は、前記処理ルール受信手段が新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の前記重み付け係数の和の最小値に対し、前記重み付け係数の和があらかじめ定められた一定範囲以内である処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる、
請求項 1 に記載の負荷分散振り分けシステム。

【請求項 5】

請求項 1 に記載の負荷分散振り分けシステムであって、
前記振り分けルール決定手段は、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、
請求項 1 に記載の負荷分散振り分けシステム。

20

【請求項 6】

請求項 1 に記載の負荷分散振り分けシステムであって、
前記振り分けルール決定手段は、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、
請求項 1 に記載の負荷分散振り分けシステム。

30

【請求項 7】

サーバに具備されたアプリケーションイベント処理結果を通知する複数の処理サーバと、前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト依存振り分け手段と、を有する負荷分散振り分けシステムで用いられ、

アプリケーションから受信した処理ルールを処理サーバに振り分けるイベント処理分散制御装置であって、

前記処理サーバごとに、振り分けた前記処理ルールのうち少なくとも前記処理ルールに含まれるイベント条件を記録する処理ルール振り分け記録手段を有し、

アプリケーションから新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の最小値に対し、あらかじめ定められた一定範囲以内の数の処理ルールを割り当てられた処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる、イベント処理分散制御装置。

40

【請求項 8】

請求項 7 に記載のイベント処理分散制御装置であって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、請求項 7 に記載のイベント処理分散制御装置。

【請求項 9】

50

請求項7に記載のイベント処理分散制御装置であって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、請求項7に記載のイベント処理分散制御装置。

【請求項10】

サーバに具備されたアプリケーションイベント処理結果を通知する複数の処理サーバと、前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト依存振り分け手段とを有する負荷分散振り分けシステムで用いられ、アプリケーションから受信した処理ルールを処理サーバに振り分けるイベント処理分散制御装置であって、

前記処理ルールごとに重み付け係数を備え、

前記処理サーバごとに、振り分けた前記処理ルールのうち少なくとも前記処理ルールに含まれるイベント条件を記録する処理ルール振り分け記録手段を有し、

新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の前記重み付け係数の和の最小値に対し、前記重み付け係数の和があらかじめ定められた一定範囲以内である処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる、イベント処理分散制御装置。

【請求項11】

請求項10に記載のイベント処理分散制御装置であって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、請求項10に記載のイベント処理分散制御装置。

【請求項12】

請求項10に記載のイベント処理分散制御装置であって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、請求項10に記載のイベント処理分散制御装置。

【請求項13】

サーバに具備されたアプリケーションイベント処理結果を通知する複数の処理サーバと、前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト依存振り分け手段とを有する負荷分散振り分けシステムで用いられ、アプリケーションから受信した処理ルールを処理サーバに振り分けるイベント処理分散制御プログラムであって、

前記処理サーバごとに、振り分けた前記処理ルールのうち少なくとも前記処理ルールに含まれるイベント条件を記録する処理ルール振り分け記録手段を有し、

新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の最小値に対し、あらかじめ定められた一定範囲以内の数の処理ルールを割り当てられた処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる、イベント処理分散制御プログラム。

【請求項14】

請求項13に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、請求項13に記載のイベント処理分散制御プログラム。

【請求項15】

請求項13に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、請求項13に記載のイベント処理分散制御プログラム。

【請求項16】

サーバに具備されたアプリケーションイベント処理結果を通知する複数の処理サーバと、前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト依存振り分け手段とを有する負荷分散振り分けシステムで用いられ、アプリケーションから受信した処理ルールを処理サーバに振り分けるイベント処理分散制御プログラムであって、

前記処理ルールごとに重み付け係数を備え、

前記処理サーバごとに、振り分けた前記処理ルールのうち少なくとも前記処理ルールに含まれるイベント条件を記録する処理ルール振り分け記録手段を有し、

アプリケーションから新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の前記重み付け係数の和の最小値に対し、前記重み付け係数の和があらかじめ定められた一定範囲以内である処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる、イベント処理分散制御プログラム。

10

【請求項 17】

請求項 16 に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、請求項 16 に記載のイベント処理分散制御プログラム。

【請求項 18】

請求項 16 に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる、請求項 16 に記載のイベント処理分散制御プログラム。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数配置された処理サーバの負荷分散を実現する負荷分散振り分けシステムに関する。

【背景技術】

【0002】

従来、デバイスから発生するイベントを処理するシステムでは、図 5 に示すように 1 台の処理サーバ 504 で処理を行っていた。このため、デバイス 505 a ~ f の数やイベント 511 a ~ f の量が増加するにしたがって、処理サーバ 504 が処理のボトルネックとなり、システムの処理能力を上げられないという問題があった。

30

【0003】

図 6 に示す負荷分散方式では、Web サービスのようなサーバクライアント方式を前提とし、各サーバ 606 a, b は、同一のアプリケーションプログラム 608 a, b を実装し、TCP コネクションや cookie を参照して、同じクライアント 605 a ~ f からのセッションは同じサーバ 606 a, b に転送している。

【0004】

また、特許文献 1 に開示されている発明では、パケットプロセッサからの応答に従って、パケット分散制御部がどのパケットプロセッサに処理を要求するか決定している。

40

【0005】

さらに、特許文献 2 に開示されている発明では、プロセッサ内に振り分けプログラムを配置して、制御チャンネル番号や端末番号に基づき、プロセッサ間で処理を受け渡している。

【0006】

【特許文献 1】特開 2000 - 222374 号公報

【特許文献 2】特開 2004 - 005249 号公報

【発明の開示】

50

【発明が解決しようとする課題】

【0007】

本発明は係る問題に鑑みてなされたものであり、処理サーバを複数配置し、処理サーバの前段にコンテキスト依存振り分け器103を配置することで、それぞれの処理サーバが設定された処理ルールを実行するために必要なイベントが適切に分配させるようにコンテキスト依存振り分け器103に振り分けルールを設定する負荷分散システ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】

【0008】

上記目的を達成するために、請求項1記載の負荷分散振り分けシステムは、サーバに具備されたアプリケーションにイベント処理結果を通知する複数の処理サーバと、前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト振り分け手段と、前記複数の処理サーバにイベントの内容を示すコンテキストに基づく処理を示す処理ルールを設定し、前記複数の処理サーバが前記処理ルールを実行するために必要なイベントが適切に分配されるように振り分けルールを前記コンテキスト振り分け手段に記憶させるイベント処理分散制御手段と、を有し、前記イベント処理分散制御手段は、前記処理サーバごとに、前記処理ルールに含まれる、処理すべきイベントの条件を示すイベント条件を少なくとも記録する処理ルール振り分け記録手段と、アプリケーションから処理ルールを受信する処理ルール受信手段と、前記処理ルール受信手段が新規に処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の最小値に対し、あらかじめ定められた一定範囲以内の数の処理ルールを割り当てられた処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる振り分けルール決定手段と、を有する。

10

20

【0009】

請求項2記載の発明は、請求項1記載の負荷分散振り分けシステムであって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

30

【0010】

請求項3記載の発明は、請求項1記載の負荷分散振り分けシステムであって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

【0011】

請求項4記載の発明は、請求項1記載の負荷分散振り分けシステムであって、前記処理ルールごとに重み付け係数を備え、前記振り分けルール決定手段は、前記処理ルール受信手段が新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の前記重み付け係数の和の最小値に対し、前記重み付け係数の和があらかじめ定められた一定範囲以内である処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる。

40

【0012】

請求項5記載の発明は、請求項1に記載の負荷分散振り分けシステムであって、前記振り分けルール決定手段は、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

50

【 0 0 1 3 】

請求項 6 記載の発明は、請求項 1 に記載の負荷分散振り分けシステムであって、前記振り分けルール決定手段は、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

【 0 0 1 4 】

請求項 7 に記載の発明のイベント処理分散制御装置は、サーバに具備されたアプリケーションイベント処理結果を通知する複数の処理サーバと、前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト依存振り分け手段と、を有する負荷分散振り分けシステムで用いられ、アプリケーションから受信した処理ルールを処理サーバに振り分けるイベント処理分散制御装置であって、前記処理サーバごとに、振り分けた前記処理ルールのうち少なくとも前記処理ルールに含まれるイベント条件を記録する処理ルール振り分け記録手段を有し、アプリケーションから新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の最小値に対し、あらかじめ定められた一定範囲以内の数の処理ルールを割り当てられた処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる。

10

【 0 0 1 5 】

請求項 8 に記載の発明のイベント処理分散制御装置は、請求項 7 に記載のイベント処理分散制御装置であって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

20

【 0 0 1 6 】

請求項 9 に記載の発明のイベント処理分散制御装置は、請求項 7 に記載のイベント処理分散制御装置であって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

30

【 0 0 1 7 】

請求項 10 に記載の発明のイベント処理分散制御装置は、サーバに具備されたアプリケーションイベント処理結果を通知する複数の処理サーバと、前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト依存振り分け手段とを有する負荷分散振り分けシステムで用いられ、アプリケーションから受信した処理ルールを処理サーバに振り分けるイベント処理分散制御装置であって、前記処理ルールごとに重み付け係数を備え、前記処理サーバごとに、振り分けた前記処理ルールのうち少なくとも前記処理ルールに含まれるイベント条件を記録する処理ルール振り分け記録手段を有し、新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の前記重み付け係数の和の最小値に対し、前記重み付け係数の和があらかじめ定められた一定範囲以内である処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる。

40

【 0 0 1 8 】

請求項 11 に記載の発明のイベント処理分散制御装置は、請求項 10 に記載のイベント処理分散制御装置であって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

50

【 0 0 1 9 】

請求項 1 2 に記載の発明のイベント処理分散制御装置は、請求項 1 0 に記載のイベント処理分散制御装置であって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

【 0 0 2 0 】

請求項 1 3 に記載の発明のイベント処理分散制御プログラムは、サーバに具備されたアプリケーションイベント処理結果を通知する複数の処理サーバと、前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト依存振り分け手段とを有する負荷分散振り分けシステムで用いられ、アプリケーションから受信した処理ルールを処理サーバに振り分けるイベント処理分散制御プログラムであって、前記処理サーバごとに、振り分けた前記処理ルールのうち少なくとも前記処理ルールに含まれるイベント条件を記録する処理ルール振り分け記録手段を有し、新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の最小値に対し、あらかじめ定められた一定範囲以内の数の処理ルールを割り当てられた処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる。

10

【 0 0 2 1 】

請求項 1 4 に記載の発明のイベント処理分散制御プログラムは、請求項 1 3 に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

20

【 0 0 2 2 】

請求項 1 5 に記載の発明のイベント処理分散制御プログラムは、請求項 1 3 に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

30

【 0 0 2 3 】

請求項 1 6 に記載の発明のイベント処理分散制御プログラムは、サーバに具備されたアプリケーションイベント処理結果を通知する複数の処理サーバと、前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト依存振り分け手段とを有する負荷分散振り分けシステムで用いられ、アプリケーションから受信した処理ルールを処理サーバに振り分けるイベント処理分散制御プログラムであって、前記処理ルールごとに重み付け係数を備え、前記処理サーバごとに、振り分けた前記処理ルールのうち少なくとも前記処理ルールに含まれるイベント条件を記録する処理ルール振り分け記録手段を有し、アプリケーションから新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の前記重み付け係数の和の最小値に対し、前記重み付け係数の和があらかじめ定められた一定範囲以内である処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てる、イベント処理分散制御プログラム。

40

【 0 0 2 4 】

請求項 1 7 に記載の発明のイベント処理分散制御プログラムは、請求項 1 6 に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

50

【 0 0 2 5 】

請求項 18 に記載のイベント処理分散制御プログラムは、請求項 16 に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てる。

【 0 0 2 6 】

請求項 20 に記載の発明のイベント処理分散制御プログラムは、請求項 19 に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てることを特徴とする。

10

【 0 0 2 7 】

請求項 21 に記載の発明のイベント処理分散制御プログラムは、請求項 19 に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てることを特徴とする。

【 0 0 2 8 】

請求項 22 に記載の発明のイベント処理分散制御プログラムは、複数のアプリケーションイベント処理結果を通知する複数の処理サーバと、前記複数の処理サーバへ端末で発生したイベントを振り分けるコンテキスト依存振り分け手段とを有する負荷分散振り分けシステムで用いられ、アプリケーションから受信した処理ルールを処理サーバに振り分けるイベント処理分散制御プログラムであって、前記処理ルールごとに重み付け係数を備え、前記処理サーバごとに、振り分けた前記処理ルールのうち少なくとも前記処理ルールに含まれるイベント条件を記録する処理ルール振り分け記録手段を有し、新規の処理ルールを受信すると、前記処理ルール振り分け記録手段を参照して、前記処理サーバに割り当てられた処理ルールの数の前記重み付け係数の和の最小値に対し、前記重み付け係数の和があらかじめ定められた一定範囲以内である処理サーバであって、同一のイベント条件を有する処理サーバに、前記新規の処理ルールを割り当てることを特徴とする。

20

【 0 0 2 9 】

請求項 23 に記載の発明のイベント処理分散制御プログラムは、請求項 22 に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、すでに割り当てられているイベント条件の一部が前記新規の処理ルールのイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てることを特徴とする。

30

【 0 0 3 0 】

請求項 24 に記載の発明のイベント処理分散制御プログラムは、請求項 22 に記載のイベント処理分散制御プログラムであって、同一のイベント条件を有する処理サーバがない場合、前記新規の処理ルールのイベント条件の一部がすでに割り当てられているイベント条件と一致する処理サーバに前記新規の処理ルールを割り当てることを特徴とする。

【 発明の効果 】

40

【 0 0 3 1 】

本発明によれば、処理サーバを複数配置し、その前段にコンテキスト依存振り分け手段を配置し、イベント処理分散制御部がそれぞれの処理サーバに個別の処理ルールを設定し、コンテキスト依存振り分け手段にそれぞれの処理サーバが設定された処理ルールを実行するために必要なイベントが適切に分配されるように振り分けルールを設定することにより、それぞれの処理サーバが異なったイベント処理を行い、異なったイベントを必要とする環境でも処理サーバの負荷分散を実現することができる。

【 発明を実施するための最良の形態 】

【 0 0 3 2 】

本発明の概略を説明する。 本発明の負荷分散振り分けシステムは、アプリケーション

50

がシステムに対して、RFIDリーダ、センサ等のデバイスからの発生情報（イベント）の処理を処理ルールとして要求し、システムは要求されたルールに従って、イベントの内容（コンテキスト）に基づき処理し、結果をアプリケーションに通知するシステムを前提としている。処理ルールは、イベントの内容（コンテキスト）に基づいて処理するように構成されている。

【0033】

このようなシステムにおいて、処理の負荷分散のために、処理サーバを複数配置し、イベント処理分散制御部がそれぞれの処理サーバに別個のアプリケーションのためのイベント処理ルールを設定する。それに従って、イベント処理分散制御部が振り分け装置に、各サーバが処理すべきイベントがそのサーバに分配されるようにコンテキストに基づいた振り分けルールを設定することにより、処理サーバの負荷分散を実現する。

10

【0034】

さらに、コンテキストに基づいてイベントを適切な処理サーバに振り分けるコンテキスト依存振り分け器103の前段に、イベントのコンテキストに基づかずイベントをランダムに振り分けるコンテキスト非依存振り分け器を配置し、その後段に同一の振り分けルールを持ったコンテキスト依存振り分け器103を複数配置する。

【0035】

デバイスからのイベントはコンテキスト非依存振り分け器により、ランダムに複数のコンテキスト依存振り分け器103に分配され、同一の振り分けルールを持ったコンテキスト依存振り分け器103によりコンテキストに基づいて適切な処理サーバに転送される。これによりコンテキスト依存振り分け器103が一台の時と同様に負荷分散のためのイベント振り分けができると同時に、コンテキスト依存振り分け器103の負荷分散も可能となる。

20

【0036】

次に、図面を参照して、本発明の実施形態を説明する。

【0037】

負荷分散振り分けシステムは、イベント処理分散制御部101、コンテキスト依存振り分け器103、複数の処理サーバ104a～fから構成され、複数のRFIDリーダ、センサ等のデバイス105a～fがコンテキスト振り分け器103に接続され、複数のアプリケーション106a～hが処理サーバ104a～fに接続される。

30

【0038】

処理サーバ104a～fの数、デバイス105a～fの数、アプリケーション106a～hの数は同一でなくてもよい。例えば、処理サーバ104aのように1つの処理サーバが複数のアプリケーション106a, bに対応してもよい。また、アプリケーション106bのように、システムに複数のイベント処理を要求する場合、複数の処理サーバ104a, bに対応してもよい。

【0039】

コンテキスト依存振り分け器103の中には、イベントを振り分ける方針を記した振り分けルール107が配置されている。また、処理サーバ104a～fにはイベントの処理方法を記した処理ルール108a～fが配置されている。それらの情報はイベント処理分散制御部101から供給される（矢印109, 110a～f）。

40

【0040】**[第1の実施形態]**

次に、第1の実施形態における負荷分散振り分けシステムの処理動作について、図1を参照して説明する。

【0041】

まず、イベント処理分散制御部101は、処理サーバ104a～fに、アプリケーション106a～fが要求するイベント処理の処理ルール108a～fを設定する（矢印110a～f）。

【0042】

50

アプリケーション106a~hが要求するイベント処理の処理ルール108a~fを設定する場合に、例えば、RFIDシステムにおいて、アプリケーション106aが、ある製品を製造するために必要な全ての部材のRFIDタグを全て検出したら通知することを、処理サーバ104aの処理ルール108aに「全ての部材のRFIDタグを全て検出したらアプリケーションに通知」というルールを設定する(矢印110a)。

【0043】

次に、イベント処理分散制御部101は、設定された処理ルール108a~fにより処理サーバが処理する必要のあるイベントがそのサーバに適切に振り分けられるように、コンテキスト依存振り分け器103に振り分けルール107を設定する(矢印109)。上記に示した例の場合、「その製品に関する部材のRFIDタグの検出イベントは処理サーバ104aに転送する。」といったルールとなる。

10

【0044】

デバイス105a~fで発生したイベント111a~fは、全てコンテキスト依存振り分け器103で受信される。デバイス105a~fで発生したイベントを受信したコンテキスト依存振り分け器103は、そのイベントの内容(コンテキスト)を参照し、振り分けルール107に従って、それらを適切な処理サーバ104a~fに転送する(矢印113a~f)。

【0045】

コンテキスト依存振り分け器103によって振り分けられたイベント113a、113fを受信した処理サーバは、処理ルール107に従ってアプリケーション106a~hに転送する(114a~i)。

20

【0046】

[第2の実施形態]

次に、図2を参照して、第2の実施形態を説明する。

【0047】

第2の実施形態では、コンテキスト依存振り分け器103203a~cを複数配置し、その前段にコンテキストを参照せずにイベントを複数のコンテキスト依存振り分け器103203a~cにランダムに振り分ける(矢印212a~c)コンテキスト非依存振り分け器202を配置する。イベント処理分散制御部201から各コンテキスト依存振り分け器103203a~cに設定される振り分けルールは207a~cは全て同一で、その内容、手順は第1の実施形態と同様である。

30

【0048】

処理サーバを複数配置し、処理サーバの負荷を分散させるためのコンテキスト依存振り分け器103であるが、コンテキスト依存振り分け器103自体は、振り分けのためにイベントのコンテキストを参照するため処理負荷が重くなるという問題があった。しかし、図2に示すように、コンテキストを参照しないでイベントをランダムに振り分けるといった処理の軽いコンテキスト非依存振り分け器を前段に配置し、その後段に複数のコンテキスト依存振り分け器103を複数配置することにより、コンテキスト依存振り分け器103の負荷分散を実現することができる。

【0049】

40

[第3の実施形態]

第3の実施形態では、第1の実施形態におけるコンテキスト依存振り分け器103内の振り分けルール107を、デバイス305a~f内に配置している(307a~f)。これにより、デバイス305a~fがイベントを検出すると、自律的にコンテキスト依存のイベント振り分け313a~fが可能となり、コンテキスト依存振り分け器103を配置せずに済み、コンテキスト依存振り分けの負荷をデバイスに分散させることができる。

【0050】

[第4の実施形態]

第4の実施形態では、第2の実施形態におけるコンテキスト非依存振り分け器、コンテ

50

キスト依存振り分け器 103 に障害回避機能を付加した例である。

【0051】

コンテキスト非依存振り分け器は402、402'と二重化されており、デバイス405a~fは両方のコンテキスト非依存振り分け器402、402'にイベント411a, a'~f, f'を通知する。もしくはデバイス405a~405fは1イベントパケットのみを送信し、そのパケットを両方のコンテキスト非依存振り分け器402、402'が受信してもよい。

【0052】

両方のコンテキスト非依存振り分け器402、402'はお互いに障害確認415を行い、通常はコンテキスト非依存振り分け器402のみがイベント振り分け412a'~c'を行い、もう一方は行わない。しかし、障害確認415によりコンテキスト非依存振り分け器402が障害になったことを検知すると、コンテキスト非依存振り分け器402'がイベント振り分け412a'~c'を開始する。これにより、コンテキスト非依存振り分け器の障害回避が可能となる。

10

【0053】

また、コンテキスト依存振り分け器103403a~cは、コンテキスト非依存振り分け器402, 402'にコンテキスト依存振り分け器103自身の生存通知416a, a'~c, c'を行っており、コンテキスト非依存振り分け器402、402'がイベント振り分け412a~412c、412a'~c'を行う際、生存しているコンテキスト依存振り分け器103のみにイベント振り分けを行い、生存していない(障害中の)コンテキスト依存振り分け器103にはイベント振り分けを行わないことにより、コンテキスト依存振り分け器103403a~cの障害回避が可能となる。

20

【0054】

[第5の実施形態]

さて、第1の実施形態においては、ひとつの処理ルールはひとつの処理サーバで処理される。一方、アプリケーションの数が増えると、ひとつの処理サーバで処理すべき処理ルールの数が増加することになる。この場合、処理ルールが特定の処理サーバに集中して割り当てられた場合には、その処理サーバの処理負荷が増大してしまうという問題が発生する。

【0055】

これを解決する方法として、ひとつの処理ルールを複数の処理サーバに割り当てる方法がある。しかし、この方法は、処理ルールが過去の処理の履歴に依存しないステートレスな処理である場合には有効であるが、処理ルールが過去の処理の履歴に依存するステートフルな処理である場合には、複数の処理サーバがアクセス可能な共有メモリを設け、同一処理ルールを処理する処理サーバ間で共有メモリを参照、更新しながら処理ルールを適用する必要が発生する。このような方法では、共有メモリとの通信や処理サーバ間の排他制御が必要となり、それがボトルネックとなって全体の処理能力を落とす恐れがある。したがって、ひとつの処理ルールはひとつの処理サーバで処理することが、特に大規模なシステムでは有効である。

30

【0056】

図7にイベント処理分散制御部101の構成を示す。

イベント処理分散制御部101は、アプリケーション106a~hから処理ルールを受信する処理ルール受信部200、処理ルールをどの処理サーバに割り当てるかを決定する処理ルール振り分け部201、処理ルールを処理サーバに送信する処理ルール送信部202、処理ルールの処理サーバから振り分けルールを生成する振り分けルール生成部203、振り分けルールをコンテキスト依存振り分け器103に送信する振り分けルール送信部204、処理サーバごとにどの処理ルールを振り分けたかを記録する処理ルール振り分け記録部205を有する。

40

【0057】

まず、この処理ルールの一例を図8に示す。この処理ルールは、ルールの名称、処理す

50

べきイベントの条件（イベント条件）、イベントに対する処理の記述を有する。

行0001は、処理ルールの名称を示す。行0002は、この処理ルールを適用するイベントの条件を示す。ここではイベントにtag_kindという、RFIDタグの種別を示すフィールドがあり、そのフィールドの値が4である場合に、この処理ルールを適用するという意味を持つ。たとえば、イベントが、event(tag_kind,TagID)であらわせるとき、tag_kind=4であるようなすべてのTagIDに関するイベントに対して処理を行うことを示す。

0003以下の行が処理の内容を示すものである。0004行は処理ルールを開始する際の初期設定時の処理を示す。ここでは、0005行目でeventlogというイベントの格納する配列記憶領域を確保し、0006行目で、イベント格納数であるevent_countという整数変数を0クリアすることを示す。

10

【0058】

0008行～0018行が、イベントを受信したときの処理である。0009行で、受信したイベントをeventlogに保存し、0010行でevent_countをインクリメントする。0011行でもしevent_countが10となったら、0012行でeventlogに記録されている10個のイベントを、ルール処理を設定したアプリケーションに送信するとともに、0013行でevent_countを0にし、0014行でタイマーをクリアする。

【0059】

もし、0011行でevent_countが10未満の場合は、0017行で、タイマー10秒をセットし、タイマーが満了したら、0019行のtimeout処理を行うことを指示する。0014行のタイマーのクリアは、この処理をキャンセルするものである。

20

【0060】

0019行のtimeoutの処理は、eventlogにひとつでもイベントが未送信状態で残っているときに、最後のイベントを受信してから10秒たったら起動される処理であって、0020行で、eventlogに記録されているイベントを、ルール処理を設定したアプリケーションに送信するとともに、0021行でevent_countを0にし、0022行でタイマーをクリアする。

【0061】

つまり、この処理ルールは、tag_kindが4であるイベントを10個受信するか、最後のイベントを受信してから10秒たつと、記録されているイベントをルール処理を設定したアプリケーションに送信するというものであり、処理サーバからアプリケーションへのイベント送信に必要な処理を軽減するための処理ルールである。

30

【0062】

また、この処理ルールは、event_count、eventlogの記録部を必要とし、その状態によって、イベントに対する処理内容が変わるため、ステートフルな処理である。

【0063】

次に、図9に、処理ルール振り分け記録部205に記録する情報の一例を示す。処理ルール振り分け記録部206には、少なくとも、処理サーバごとに設定した処理ルールの数と、処理ルールのイベント条件とを記録する必要があるが、ここでは、処理サーバごとに、設定処理ルール数、処理ルール名とイベントの条件を記録するものとする。ここではEP1～EP6が処理サーバ108a～fに対応する識別子であり、すでに18個のルールが設定されており、イベント条件は、tag_kindのみで決定できるものとする。

40

【0064】

イベント処理分散制御部は、アプリケーションから処理ルールを処理ルール受信部によって受信する。処理ルール受信部200は、受信した処理ルールを処理ルール振り分け部201に渡す。

【0065】

図10は、処理ルール振り分け部201がその処理ルールをどの処理サーバに振り分けるかを決定する手順を示すフローチャートである。

【0066】

まず、処理ルール振り分け部201は、各処理サーバに割り当てられたルール数の最小値を求める。図9の例においては、最小値はEP1およびEP3の“2”である。

50

【 0 0 6 7 】

処理ルール振り分け部 2 0 1 は、ルール数の最小値との差があらかじめ定められた一定値以下である処理サーバの候補を選択する（ステップ 1）。

この一定値が“ 0 ”であれば、EP 1、EP 2 が選択される。また 1 であれば、EP1、EP2、EP 3、EP5 が選択される。ここでは、一定値を 1 として EP1、EP2、EP3、EP5 が選択されたものと仮定する。ここで、処理サーバの候補がひとつになった場合は、その処理サーバを選択する。

【 0 0 6 8 】

つぎに、処理ルール振り分け部 2 0 1 は、処理ルールのイベント条件と、選択済みの EP 1、EP2、EP3、EP5 のイベント条件を比較する。（ステップ 2）

ここでもし、一致するイベント条件をもつ処理サーバが 1 つだけあれば、その処理サーバを最終的な振り分け先として決定する。たとえば、tag_kind = 4 がイベント条件である場合には、EP1、EP2、EP3、EP5 のうちそのイベント条件を有する処理サーバは、EP 2 だけひとつであるので、EP 2 を振り分け先として決定する。（ステップ 3 -A）

一致するイベント条件を持つ処理サーバが複数あった場合は、その複数の処理サーバからイベント数が最も小さい処理サーバを選択し、その処理サーバを振り分け先として決定する。たとえば、tag_kind = 5 をイベント条件とした場合、EP2 と EP 5 がそのイベント条件を持っているが、この場合はイベント数が小さい、EP 2 を選択するものとする。（ステップ 3 -B）

一致するイベント条件を持つ処理サーバがひとつもない場合にも、複数の処理サーバからイベント数が最も小さい処理サーバを選択し、その処理サーバを振り分け先として決定する。（ステップ 3 -C）

なお、前記ステップ 3 -A、3 -B、3 -C で、複数の処理サーバが候補として残る場合がある。このときは、若番/老番で決定する、ランダムに決定するなど、別のルールで決定すればよい。

【 0 0 6 9 】

なお、ステップ 1 において、ルール数の最小値との差があらかじめ定められた一定値以下である処理サーバの候補を選択するとしているが、必ずしも差である必要はなく、たとえば、ルール数の最小値の 1 . 5 倍以内など、比率で選択してもよい。

【 0 0 7 0 】

また、新たな処理ルールのイベント条件が tag_kind = 7 である場合、全く同じイベント条件はないが、rule12 の tag_kind = 5 or tag_kind = 7 の条件に tag_kind = 7 の条件が包含されていることから、EP 5 を条件に合致するものとしてもよい。

また逆に、新たな処理ルールのイベント条件が、tag_kind = 3 or tag_kind = 8 のように複数の条件の or であった場合、もし、すべてのイベント条件を持つ処理サーバがあればそれを優先的に選択する。すべてのイベント条件を満たす処理サーバがない場合は、部分的に一致するイベント条件を満たす処理サーバを選択する。たとえば、tag_kind = 3 が部分的に一致することから、EP 3 を選択する。

【 0 0 7 1 】

なお、上記説明では、イベント条件として、タグの種別を示す tag_kind がある値を持つ場合について説明したが、イベント条件はこれに限るわけではなく、あるイベントに対してそれをイベントそのもので選別できる条件であれば、どのようなものであってもよい。さらに、上記説明では、各処理サーバに割り当てられた処理ルール数に基づくものとしているが、それぞれの処理ルールに、イベント条件に合致するイベントの数や処理の負荷の度合いに重み付けを行った負荷度を定義し、負荷度の合計を処理ルール数に代えて用いてもよい。

【 0 0 7 2 】

処理ルールを処理すべき処理サーバを決定すると、処理ルール振り分け部 2 0 1 は、処理ルール振り分け記録部 2 0 5 にその処理ルールの振り分け結果を登録して更新する。次にあらたな処理ルールが発生すると、更新された処理ルール振り分け記録部 2 0 5 の内容

10

20

30

40

50

を元に、処理サーバへの振り分けを決定する。

さらに、処理ルール振り分け部 201 は、処理ルール送信部 202 を介して処理ルールを振り分けることに決定した処理サーバに送信する。

【0073】

また、処理ルール振り分け部 201 は、その振り分け結果を振り分けルール生成部 203 に通知する。振り分けルール生成部 203 は、その処理ルールのイベント条件と処理サーバをペアとして、振り分けルールを生成し、振り分けルール送信部 204 を介して、コンテキスト依存振り分け器 103 に送信する。

【0074】

なお、コンテキスト依存振り分け器 103 は、複数の振り分けルールを適用してあるイベントをどの処理サーバに送るかを決定するが、あるイベントについて複数振り分けルールがあるひとつの処理サーバを送信先とする場合、振り分けルールごとにイベントを送信するのではなく、一回だけ送信し、処理サーバ側で複数の処理ルールを適用することが望ましい。

10

【0075】

さらには、複数の処理ルールについて、イベント条件と処理サーバが同一の場合、振り分けルールをコンテキスト依存振り分け器 103 に送信しないように構成してもよい。

【0076】

以上説明したように、本実施例に示す処理ルールの振り分けを行うことによって、負荷が一部の処理サーバに集中することがなくなると同時に、同一または一部共通するイベント条件を持った処理ルールを特定のサーバに集中させることができるので、あるイベントについて、コンテキスト依存振り分け器 103 が送信する相手となる処理サーバの数を小さくできるようになる。また、全処理サーバが受信するイベント数の総数を減少させることができるので、ひとつの処理サーバでも受信するイベント数の総数が減少し、全体の処理能力を向上させることができるようになる。

20

【0077】

[第6の実施形態]

別の実施形態における本発明の処理ルール振り分けの適用例を示す。

図11に処理ルール振り分けの流れを示す。アプリケーション106a~106hから処理ルールが設定されるたびに図11の処理が実施される。図11のステップ1の許容処理ルール数差はあらかじめ決めておく値である。いずれかのステップにおいて、処理サーバが1つに絞られた場合には、その処理サーバに処理ルールを割り当てる。イベント種別の識別方法として、例えば、イベントに属性を持たせ、その属性を用いて分類することが可能である。

30

各ステップを実施する順番は固定ではなく入れ替えることも可能である。

【0078】

(イベント、処理ルールの定義)

イベントと処理ルールの例を用いて、上記処理ルール振り分け方法を説明する。例では、デバイスをRFIDリーダーとし、アプリケーションをRFIDリーダーが通知するRFIDの情報を利用するアプリケーションプログラムとする。イベントはRFID(TagID)とそのRFIDを読み取ったリーダーのID(ReaderID)から構成されるとする。イベントの例を図12に示す。図12に示すイベントはTagIDが10とReaderIDが20のイベントである。

40

【0079】

アプリケーションから設定される処理ルールの例を図13に示す。

【0080】

Rule_A1からRule_A6までが順次設定されるとする。各処理ルールのカンマで区切られた左半分が条件を表わし、右半分が動作を表わす。処理ルールの条件部分の2つのイベントが記述されているが、これは、この2つのイベントがデバイスから発生した場合に、条件に合致することを意味する。たとえば、Rule_A1の、Event(TagID=10, ReaderID=20) & Event(TagID=11, ReaderID=20) という条件は、TagID=10, ReaderID=20のイベントと、Tag

50

ID=11, ReaderID=20のイベントの両方が発生したときに処理ルールの動作部分を起動することを意味する。この場合、TagID=10, ReaderID=20のイベントと、TagID=11, ReaderID=20のイベントの両方を処理する必要があるので、この処理ルールを設定押された処理サーバには、TagID=10, ReaderID=20のイベントと、TagID=11, ReaderID=20のイベント両方は振り分けられなければならない。

【 0 0 8 1 】

また、処理ルールの動作部分の " notify to Application-A " はApplication-Aに条件に合致したことを通知することを意味する。ここでApplication-Aはアプリケーション 1 0 6 a、Application-Bはアプリケーション 1 0 6 bに対応し、以下同様である。

【 0 0 8 2 】

(処理ルール振り分け例1)

処理サーバが3台で、許容処理ルール数差が2であるとする、図 1 3 に示す処理ルールはイベント処理分散制御部によって、図 1 4 に示すように処理サーバに振り分けられる。

以下、その流れを説明する。3台の処理サーバをそれぞれEP1、EP2、EP3と呼ぶ。また、処理ルールの条件チェックに必要なイベントを " 処理ルールチェックに必要なイベント " と呼び、処理サーバに設定され処理ルールチェックに必要なイベントの集合を " 処理サーバに必要なイベント " と呼ぶ。

【 0 0 8 3 】

Rule_A1がアプリケーション 1 0 6 aから設定されると、初期状態で各処理サーバに設定されている処理ルールはないので、ステップ1では、全処理サーバが選択される。

【 0 0 8 4 】

次にステップ2に進む。処理ルールチェックに必要なイベントはEvent(TagID=10, ReaderID=20)とEvent(TagID=11, ReaderID=20)である。また、各処理サーバには処理ルールが設定されていないため、必要となるイベントはない。よって、共通に必要なイベント数は各処理サーバで等しくなるため、全ての処理サーバが選択され、ステップ3に進む。

【 0 0 8 5 】

ステップ3では、各処理サーバに処理ルールが設定されていないため、必要とするイベント数は各処理サーバにおいて0で、よって、全ての処理サーバが選択される。最終的に処理ルールを振り分ける処理サーバをランダムに選択する。EP2が選択されたとして次に進む。

【 0 0 8 6 】

Rule_A2がアプリケーション 1 0 6 bから設定されると、ステップ1では、全処理サーバに設定され処理ルールが許容処理ルール数差2以内であるため、全処理サーバが選択され、ステップ2に進む。処理ルールチェックに必要なイベントはEvent(TagID=10, ReaderID=20)とEvent(TagID=11, ReaderID=20)である。EP2に必要なイベントはEvent(TagID=10, ReaderID=20)とEvent(TagID=11, ReaderID=20)であり、他の処理サーバに必要なイベントはない。処理ルールチェックとEP2で共通に必要なイベントは2つであるため、EP2にRule_A2が設定される。

【 0 0 8 7 】

Rule_A3がアプリケーション 1 0 6 cから設定されると、ステップ1では、全処理サーバに設定され処理ルールが許容処理ルール数差2以内であるため、全処理サーバが選択され、ステップ2に進む。処理ルールチェックと各処理サーバで共通に必要なイベントはないため、ステップ2では、全ての処理サーバが選択され、ステップ3へ進む。EP2は2つのイベントを必要とし、他の処理サーバは1つもイベントを必要としないため、EP1とEP3が選択される。最終的にEP1とEP3のいずれかがランダムに選択され、選択された処理サーバにRule_A3が設定される。EP1が選択されたとして次に進む。

【 0 0 8 8 】

Rule_A4がアプリケーション 1 0 6 cから設定されると、ステップ1では、全処理サーバに設定され処理ルールが許容処理ルール数差2以内であるため、全処理サーバが選択され、ステップ2に進む。処理ルールチェックとEP2で共通に必要なイベントはEvent(TagID=11

10

20

30

40

50

、ReaderID=20)であり、処理ルールチェック他の各処理サーバで共通に必要なイベントはないため、Rule_A4はEP2に設定される。

【 0 0 8 9 】

Rule_A5がアプリケーション 1 0 6 a から設定されると、ステップ1では、EP2とEP3に設定され処理ルール数が許容処理ルール数差2より大きいいため、EP 1 とEP3が選択され、ステップ2へ進む。処理ルールチェックと各処理サーバで共通に必要なイベントはないため、全ての処理サーバが選択され、ステップ3へ進む。EP 1 が必要とする処理ルールが2つであり、EP3が必要とする処理ルールが0であるため、EP3にRule_A5が設定される。

【 0 0 9 0 】

Rule_A6がアプリケーション 1 0 6 b から設定されると、ステップ1では、全処理サーバに設定され処理ルールが許容処理ルール数差2以内であるため、全ての処理サーバが選択され、ステップ2に進む。処理ルールチェックとEP 1 で共通に必要なイベントは0で、処理ルールチェックと他の各処理サーバで共通に必要なイベントは1つである。よって、EP2とEP3が選択され、ステップ3へ進む。EP2が必要とするイベントが3つであり、EP3が必要とするイベントは2つであるため、EP3にRule_A6が設定される。

【 0 0 9 1 】

図 1 4 のように処理ルールが振り分けられる場合、コンテキスト依存振り分け器 1 0 3 に設定されるイベント振り分け処理ルールは図 1 5 のようになる。図 1 5 の表はイベントの条件とその条件に合致するイベントの転送先処理サーバを表わす。例えば、Event(TagID=10, ReaderID=20)はEP2に転送され、Event(TagID=13, ReaderID=25)はEP2とEP3に転送される。

【 0 0 9 2 】

(処理ルール振り分け例2)

上記例では、処理ルールの条件に記述されるイベントの全属性を見て、処理ルールの振り分けを行って処理ルールの条件に記述されるイベントの一部の属性を見て、処理ルールを振り分けることも可能である。以下、TagIDを参照する属性として振り分けの流れを説明する。TagIDを参照する属性として振り分けると図 1 3 に示す処理ルールは図 1 6 のように振り分けられる。以下、属性の値としてアスタリスクが記述されている場合、任意の値とする。

【 0 0 9 3 】

Rule_A1がアプリケーション 1 0 6 a から設定されると、初期状態で各処理サーバに設定されている処理ルールはないので、ステップ1では、全処理サーバが選択される。次にステップ2に進む。処理ルールチェックに必要なイベントはEvent(TagID=10, ReaderID=*)とEvent(TagID=11, ReaderID=*)である。また、各処理サーバには処理ルールが設定されていないため、必要となるイベントはない。よって、共通に必要なイベント数は各処理サーバで等しくなるため、全ての処理サーバが選択され、ステップ3に進む。各処理サーバに処理ルールが設定されていないため、必要とするイベント数は各処理サーバで0で、よって、全ての処理サーバが選択される。最終的に処理ルールを振り分ける処理サーバをランダムに選択する。EP2が選択されたとして次に進む。

【 0 0 9 4 】

Rule_A2がアプリケーション 1 0 6 b から設定されると、ステップ1では、全処理サーバに設定され処理ルールが許容処理ルール数差2以内であるため、全処理サーバが選択され、ステップ2に進む。処理ルールチェックに必要なイベントはEvent(TagID=10, ReaderID=*)とEvent(TagID=11, ReaderID=*)である。EP2で必要となるイベントはEvent(TagID=10, ReaderID=*)とEvent(TagID=11, ReaderID=*)であり、他の処理サーバで必要となるイベントはない。処理ルールチェックで必要なイベントとEP2が必要とするイベントで共通なイベントが2つであるため、EP2にRule_A2が設定される。

【 0 0 9 5 】

Rule_A3がアプリケーション 1 0 6 c から設定されると、ステップ1では、全処理サーバに設定され処理ルールが許容処理ルール数差2以内であるため、全処理サーバが選択され

10

20

30

40

50

、ステップ2に進む。処理ルールチェックで必要なイベントはEvent(TagID=11, ReaderID=*)とEvent(TagID=12, ReaderID=*)であり、処理ルールチェックで必要なイベントとEP2が必要とするイベントで共通なイベントが1つある。よって、EP2にRule_A3が設定される。

Rule_A4がアプリケーション106cから設定されると、ステップ1では、EP2に設定され処理ルール数と他の処理サーバに設定され処理ルール数との差が許容処理ルール数差2より大きいため、EP1とEP3が選択され、ステップ2へ進む。処理ルールチェックと各処理サーバで共通に必要なイベントがないため、全処理サーバが選択されステップ3へ進む。各処理サーバが必要とするイベントはないため、全処理サーバが選択される。最終的にEP1かEP3かがランダムに選択される。EP1が選択されたとして進める。

【0096】

10

Rule_A5がアプリケーション106bから設定されると、ステップ1では、EP2に設定され処理ルール数と他のEP3に設定され処理ルール数との差が許容処理ルール数差2より大きいため、EP1とEP3が選択され、ステップ2へ進む。処理ルールチェックと各処理サーバで共通に必要なイベントはないため、全処理サーバが選択され、ステップ3へ進む。EP1に必要なイベントは2つであり、EP3が必要とするイベントは0であるためEP3にRule_A5が設定される。

【0097】

Rule_A6がアプリケーション106aから設定されると、ステップ1では、全処理サーバに設定され処理ルールが許容処理ルール数差2以内であるため、全ての処理サーバが選択され、ステップ2に進む。処理ルールチェックとEP2で共通に必要なイベントは0であり、処理ルールチェックと他の処理サーバで共通に必要なイベントは1つである。よって、EP1とEP3が選択され、ステップ3へ進む。EP1とEP3が必要とするイベントはそれぞれ2つなので、最終的にランダムに選択され、選択された処理サーバにRule_A6が設定される。EP3が選択されたとすると、処理ルールの振り分け状況は図16のようになる。

20

【0098】

図16のように処理ルールが振り分けられる場合、コンテキスト依存振り分け器103に設定されるイベント振り分け処理ルールは図17のようになる。表の見方は図15と同様である。

【0099】

このような構成により、デバイスからのイベント発生量が増え、イベントデータの参照・更新が頻繁になっても、従来技術であるイベントデータを共有する場合に発生する排他制御によるボトルネックが発生しない。

30

本ルール振り分け方法を用いることによりコンテキスト依存振り分け器103におけるイベントコピーを最小限にでき、システムのリソースを効率的に使用できる。

【0100】

ルール振り分けの例として、例1と例2を示したが、例1では、コンテキスト依存振り分け器103でより厳密にイベントが絞られるため、各処理サーバは最小限のイベントのみを処理すればよい。例2では、コンテキスト依存振り分け器103が振り分け時に1属性のみを参照すればよいいため、全属性を見る場合と比べて高速に処理できるという効果がある。

40

【図面の簡単な説明】

【0101】

【図1】第1の実施形態における負荷分散振り分けシステムの構成を示した図である。

【図2】第2の実施形態における負荷分散振り分けシステムの構成を示した図である。

【図3】第3の実施形態における負荷分散振り分けシステムの構成を示した図である。

【図4】第4の実施形態における負荷分散振り分けシステムの構成を示した図である。

【図5】従来の負荷分散システムの構成を示した図である。

【図6】従来の負荷分散システムの構成を示した図である。

【図7】イベント処理分散制御部の構成を示した図である。

50

【図 8】 処理ルールの例を示した図である。

【図 9】 処理ルール振り分け記録の例を示した図である。

【図 10】 処理ルール振り分けの手順のフローチャートを示した図である。

【図 11】 ルール振り分けの手順のフローチャートを示した図である。

【図 12】 イベント例を示した図である。

【図 13】 クライアントから設定される処理ルールの例を示した図である。

【図 14】 処理分散制御サーバのルール振り分け例 1 を示した図である。

【図 15】 イベント振り分けルールの例 1 を示した図である。

【図 16】 処理分散制御サーバのルール振り分け例 2 を示した図である。

【図 17】 イベント振り分けルールの例 2 を示した図である。

10

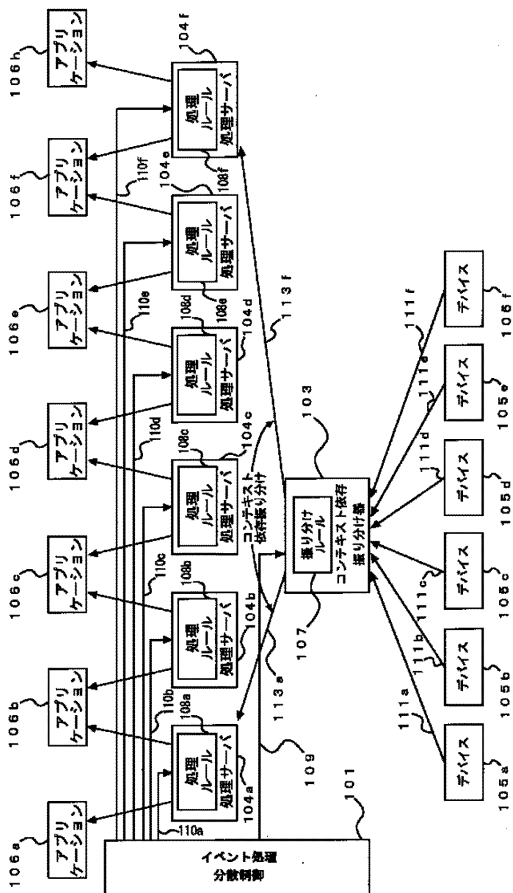
【符号の説明】

【 0 1 0 2 】

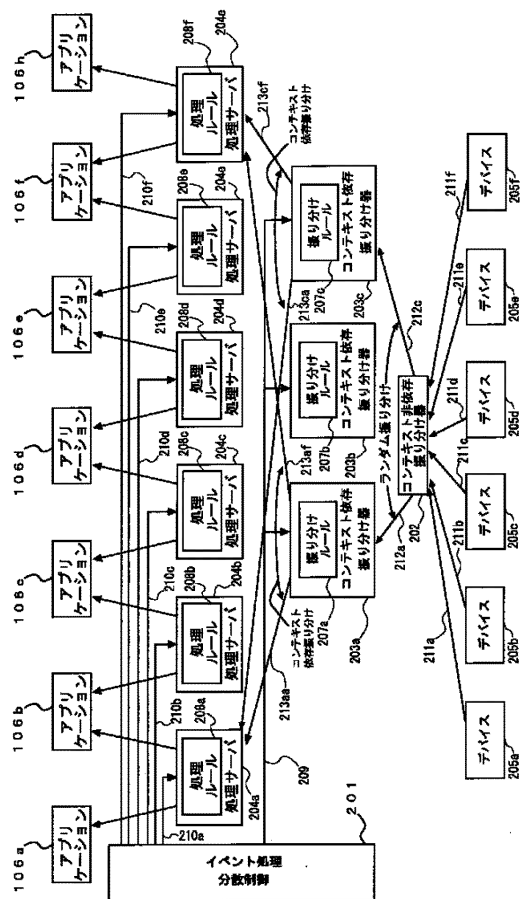
- 1 0 1 イベント処理分散制御部
- 1 0 3 コンテキスト依存振り分け器
- 1 0 4 a ~ 1 0 4 h 処理サーバ
- 1 0 5 a ~ 1 0 5 f デバイス
- 1 0 6 a ~ 1 0 6 h アプリケーション
- 1 0 7 振り分けルール
- 1 0 8 a ~ 1 0 8 f 処理ルール
- 2 0 0 処理ルール受信部
- 2 0 1 処理ルール振り分け部
- 2 0 2 処理ルール送信部
- 2 0 3 振り分けルール生成部
- 2 0 4 振り分けルール送信部
- 2 0 5 処理ルール振り分け記録部

20

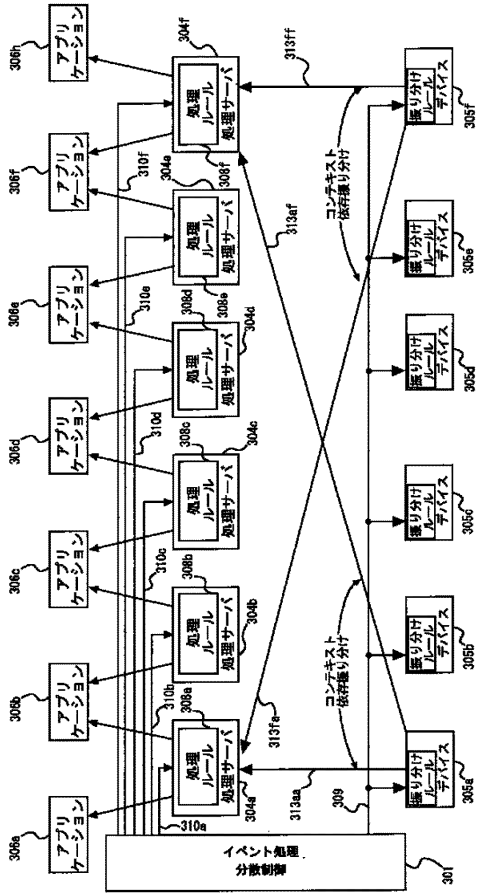
【図 1】



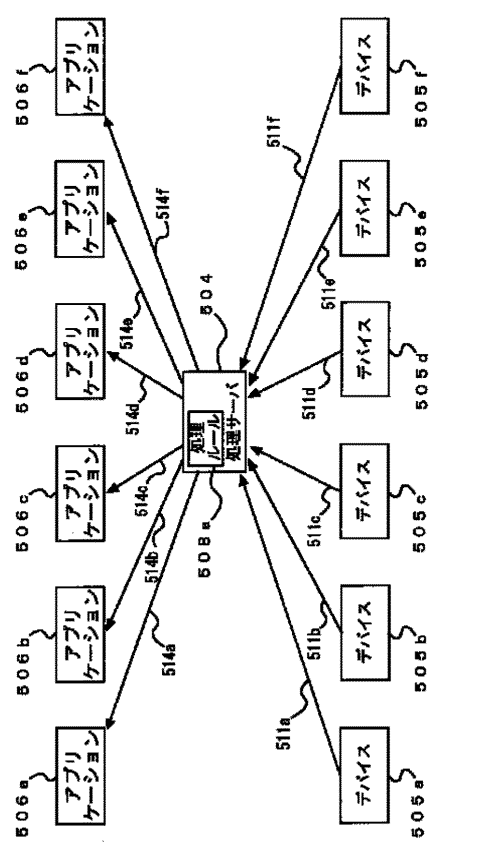
【図 2】



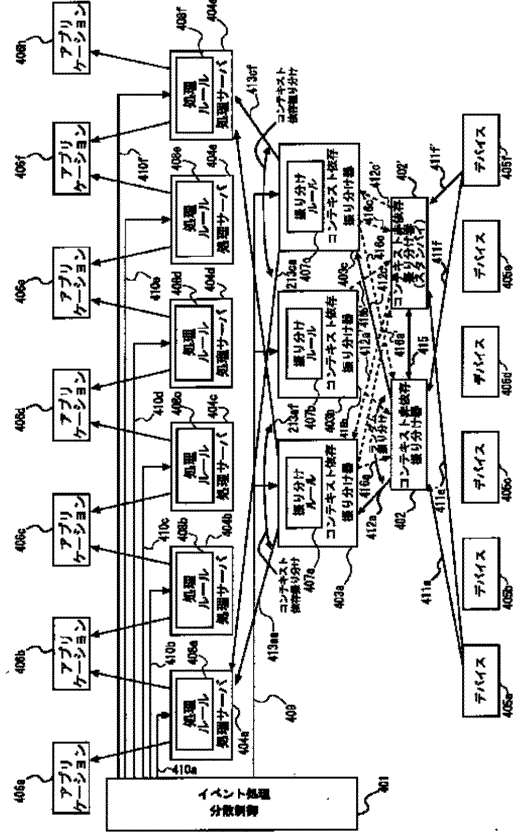
【図 3】



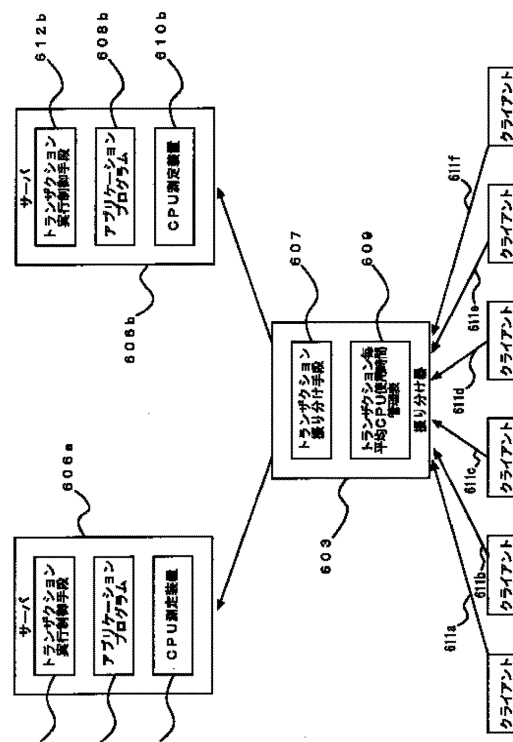
【図 5】



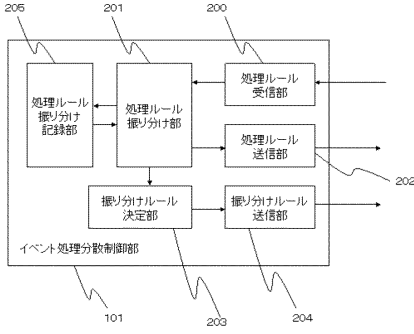
【図 4】



【図 6】



【図7】



【図8】

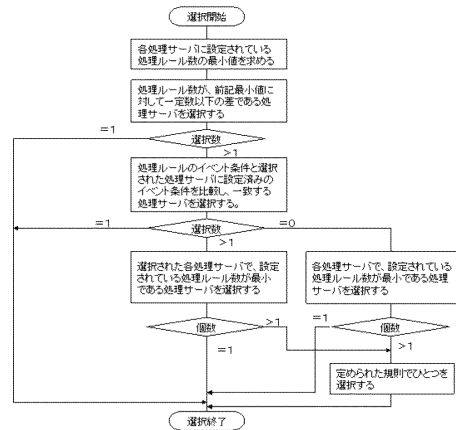
```

0001 Rule_A_name = "Rule_A-n"
0002 Event_condition = (tag_kind = 4)
0003 Action = {
0004     initial {
0005         dim eventlog[10] ,event_count
0006         event_count = 0
0007     }
0008     receive (event){
0009         eventlog[event_count] = event
0010         event_count = event_count + 1
0011         if event_count >= 10 {
0012             send_eventlog(eventlog,10)
0013             event_count = 0
0014             clear_timer()
0015         }
0016         else{
0017             set_timer(10, timeout) /#sec*/
0018         }
0019     }
0020     timeout{
0021         send_eventlog(eventlog,event_count)
0022         event_count = 0
0023         clear_timer()
0024     }
0025 }
    
```

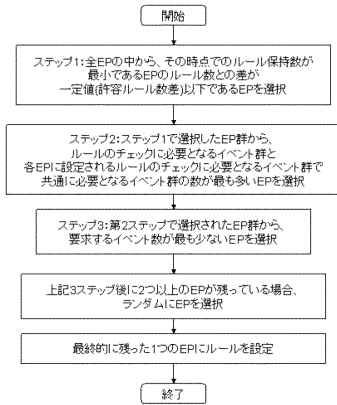
【図9】

処理サーバID	設定処理ルール数	ルール名称1	イベント条件1	ルール名称2	イベント条件2	ルール名称3	イベント条件3	ルール名称4	イベント条件4
EP1	2	rule1	tag_kind = 1	rule2	tag_kind = 3				
EP2	2	rule3	tag_kind = 2	rule4	tag_kind = 5				
EP3	3	rule5	tag_kind = 3	rule6	tag_kind = 3	rule7	tag_kind = 3		
EP4	4	rule8	tag_kind = 4	rule9	tag_kind = 4	rule10	tag_kind = 4	rule11	tag_kind = 4
EP5	3	rule12	tag_kind = 5 or tag_kind = 7	rule13	tag_kind = 5	rule14	tag_kind = 5		
EP6	4	rule15	tag_kind = 6	rule16	tag_kind = 6	rule17	tag_kind = 6	rule18	tag_kind = 6

【図10】



【 図 1 1 】



【 図 1 2 】

Event(TagID=10, ReaderID=20)

【 図 1 3 】

	処理サーバ1	処理サーバ2	処理サーバ3
割り当てられるルール	Rule_A3	Rule_A1 Rule_A2 Rule_A4	Rule_A5 Rule_A6

【 図 1 4 】

```

Rule_A1( Event(TagID=10, ReaderID=20) & Event(TagID=11, ReaderID=20) , notify to Application-A )
Rule_A2( Event(TagID=10, ReaderID=20) & Event(TagID=11, ReaderID=20) , notify to Application-B )
Rule_A3( Event(TagID=11, ReaderID=21) & Event(TagID=12, ReaderID=22) , notify to Application-C )
Rule_A4( Event(TagID=11, ReaderID=20) & Event(TagID=13, ReaderID=25) , notify to Application-C )
Rule_A5( Event(TagID=14, ReaderID=20) & Event(TagID=15, ReaderID=21) , notify to Application-A )
Rule_A6( Event(TagID=13, ReaderID=25) & Event(TagID=14, ReaderID=20) , notify to Application-B )
  
```

【 図 1 5 】

転送対象のイベント	転送先の処理サーバ
Event(TagID=10, ReaderID=20)	EP2
Event(TagID=11, ReaderID=20)	EP2
Event(TagID=11, ReaderID=21)	EP1
Event(TagID=12, ReaderID=22)	EP1
Event(TagID=13, ReaderID=25)	EP2, EP3
Event(TagID=14, ReaderID=20)	EP3
Event(TagID=15, ReaderID=21)	EP3

【 図 1 6 】

	処理サーバ1	処理サーバ2	処理サーバ3
割り当てられる ルール	Rule_A4	Rule_A1 Rule_A2 Rule_A3	Rule_A5 Rule_A6

【 図 1 7 】

転送対象のイベント	転送先の処理サーバ
Event(TagID=10, ReaderID=*)	EP2
Event(TagID=11, ReaderID=*)	EP1, EP2
Event(TagID=12, ReaderID=*)	EP2
Event(TagID=13, ReaderID=*)	EP1, EP3
Event(TagID=14, ReaderID=*)	EP3
Event(TagID=15, ReaderID=*)	EP3

フロントページの続き

審査官 田内 幸治

- (56)参考文献 特開2002-335268(JP,A)
特開平11-250020(JP,A)
特開2001-117897(JP,A)
ブロードバンド時代のWebシステム構築 サーバー負荷分散装置 アプリ情報を識別して分散
携帯サイト向けの機能強化が進む,日経インターネットテクノロジー 第53号 Nikkei Int
ernet Technology,日本,日経BP社 Nikkei Business Publications,Inc.,2001年11
月22日,P.22-P.27
森山 徹,解説 機能強化が進むWWWサーバー負荷分散装置,日経オープンシステム 第81
号 NIKKEI OPEN SYSTEMS,日本,日経BP社 Nikkei Business Publications,Inc.,1999
年12月15日,P.128-P.131

(58)調査した分野(Int.Cl.,DB名)

G06F 9/46
G06F 9/48
G06F 9/50 - 9/52
G06F 9/54
G06F 11/20
G06F 13/00
G06F 15/00