

平成29年11月30日判決言渡

平成28年(行ケ)第10078号 審決取消請求事件

口頭弁論終結日 平成29年11月2日

判 決

原 告 ブル エス アー エス

同 訴 訟 代 理 人 弁 理 士 村 越 智 史  
今 村 光 広

被 告 特 許 庁 長 官  
同 指 定 代 理 人 石 井 茂 和  
高 木 進  
野 崎 大 進  
板 谷 玲 子

主 文

- 1 特許庁が不服2015-96号事件について平成27年11月17日  
にした審決を取り消す。
- 2 訴訟費用は被告の負担とする。

事 実 及 び 理 由

第1 請求

主文同旨

第2 事案の概要

本件は、特許出願拒絶査定に対する不服審判請求を不成立とした審決の取消訴訟である。争点は、補正要件違反(新規事項の追加, 目的外補正, 独立特許要件違反)

の有無，進歩性の有無（相違点の判断）である。

## 1 特許庁における手続の経緯

原告は，名称を「命令スレッドを組み合わせた実行の管理システムおよび管理方法」とする発明につき，平成22年（2010年）9月15日を国際出願日として特許出願（請求項の数10）をし（パリ条約による優先権主張 平成21年〔2009年〕9月25日〔本願優先日〕・フランス，国際公開番号WO2011/036377。甲6），平成26年3月31日，手続補正をした（請求項の数10。甲9）が，平成26年8月26日付けで拒絶査定を受けたので，平成27年1月5日，拒絶査定不服審判請求（不服2015-96号。甲11）及び手続補正をした（本件補正，請求項の数10。甲12）。

特許庁は，平成27年11月17日，「本件審判の請求は，成り立たない。」との審決をし，その謄本は，同年12月1日，原告に送達された。

## 2 本件発明の要旨

(1) 平成26年3月31日付けで補正された本願の特許請求の範囲の請求項1～10に係る発明（以下，請求項の番号を用いて「本願発明1」～「本願発明10」といい，これらを総称して「本願発明」という。）は，以下のとおりのものである（甲9）。

### 【請求項1】

命令スレッドの実行を管理するコンピュータシステム（18）であって，各中央処理装置が，前記各中央処理装置に属する複数の仮想プロセッサ（24C，24S，26C，26S）上で，複数の命令スレッドを組み合わせて実行するように構成される少なくとも1つの中央処理装置（24，26）と，前記仮想プロセッサに命令スレッドの実行を分散するように構成されるマネージャ（38）と，を有するコンピュータシステム（18）において，実行する前記命令スレッドを複数の所定のタイプに応じて分類する分類手段（36）を有し，前記命令スレッドの実行を分散するように構成される前記マネージャ（38）を，

前記命令スレッドが分類された前記複数の所定のタイプのうちの1つに基づき、実行する前記命令スレッドを前記仮想プロセッサ（24C, 24S, 26C, 26S）に方向付けるように設計するコンピュータシステムであって、  
前記各中央処理装置の仮想プロセッサ（24C, 24S, 26C, 26S）は、前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する、ことを特徴とするコンピュータシステム。

#### 【請求項2】

前記中央処理装置（24, 26）の各仮想プロセッサ（24C, 24S, 26C, 26S）を前記所定のタイプの1つに関連付ける手段（C, S）を備える、請求項1に記載のコンピュータシステムにおいて、前記命令スレッドの実行を分散するように構成される前記マネージャ（38）を、実行する前記命令スレッドをそれぞれ該スレッドと同じタイプの仮想プロセッサ（24C, 24S, 26C, 26S）に方向付けるように設計するコンピュータシステムであって、  
前記各中央処理装置の仮想プロセッサ（24C, 24S, 26C, 26S）は、前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する、ことを特徴とするコンピュータシステム。

#### 【請求項3】

前記中央処理装置（24, 26）は2つの仮想プロセッサ（24C, 24S, 26C, 26S）を有し、1つは命令スレッドの第1のタイプ（C）に関連付け、もう1つは命令スレッドの第2のタイプ（S）に関連付け、実行するコンピュータプロセスはそれぞれ前記第1のタイプか前記第2のタイプである、請求項1または2に記載のコンピュータシステム。

#### 【請求項4】

- － 「計算」と呼ぶ命令スレッドの前記第1のタイプ（C）は、結果を供給す

るように前記コンピュータシステムが実行するようになっているプログラムに対して、該プログラムの実行に参加して結果を直接生成する命令スレッドに関係し、

－ 「サービス」と呼ぶプロセスの前記第2のタイプ（S）は、該プログラムの実行に参加して「計算」タイプの前記命令スレッドに付属サービスを供給する命令スレッドに関係する、請求項3に記載のコンピュータシステム。

#### 【請求項5】

少なくとも2つの中央処理装置（24, 26）を有し、それぞれが複数の仮想プロセッサ（24C, 24S, 26C, 26S）を備え、仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する、請求項1から4のいずれか一項に記載のコンピュータシステムにおいて、前記命令スレッドの実行を分散するように構成される前記マネージャ（38）を、中央処理装置にある仮想プロセッサとは独立して、実行する命令スレッドそれぞれに対して中央処理装置を選択するように設計するコンピュータシステムであって、

前記各中央処理装置の仮想プロセッサ（24C, 24S, 26C, 26S）は、前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する、ことを特徴とするコンピュータシステム。

#### 【請求項6】

コンピュータシステム（18）の少なくとも1つの中央処理装置（24, 26）に属する複数の仮想プロセッサ（24C, 24S, 26C, 26S）上で複数の命令スレッドを組み合わせて実行するのを管理する方法であって、前記仮想プロセッサに前記命令スレッドの実行を分散するステップ（104）を含む方法において、複数の所定のタイプに応じて実行する前記命令スレッドを分類する予備ステップ（100）を含み、前記命令スレッドの実行を分散するステップ（104）で、実行する命令スレッドそれぞれをタイプに応じて仮想プロセッサに方向付ける、方法であって、

前記各中央処理装置の仮想プロセッサ（24C，24S，26C，26S）は，前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する，ことを特徴とする方法。

**【請求項7】**

前記分類ステップ（100）で，命令スレッドを管理するアプリケーションプログラミングインターフェースの関数でタイプを識別するパラメータに基づいて，前記命令スレッドをそれぞれタイプで分類する，請求項6に記載の方法。

**【請求項8】**

前記分類ステップ（100）で，命令スレッドの実行を含むプログラムの実行コマンドでタイプを識別するパラメータに基づいて，前記命令スレッドをそれぞれこのタイプで分類する，請求項6に記載の方法。

**【請求項9】**

スーパーコンピュータタイプのサーバクラスタ（10）の複数の処理ノード上で複数の命令スレッドを実行することの管理に対する請求項6から8のいずれか一項に記載の方法の適用であって，各処理ノード（18）が，少なくとも1つの中央処理装置に属する複数の仮想プロセッサ（24C，24S，26C，26S）上で複数の命令スレッドを組み合わせて実行する前記少なくとも1つの中央処理装置（24，26）を有する適用であって，  
前記各中央処理装置の仮想プロセッサ（24C，24S，26C，26S）は，前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する，ことを特徴とする適用。

**【請求項10】**

通信ネットワークからダウンロード可能なコンピュータプログラムおよび／またはコンピュータで再生可能な媒体に記録したコンピュータプログラムおよび／またはプロセッサのなかでもとりわけコンピュータのオペレーティングシステム（34）で実行可能なコンピュータプログラムであって，前記プログラムをコンピュータ上

で実行した場合に、請求項6から8のいずれか一項に記載の複数の命令スレッドを組み合わせた実行の管理方法のステップを実行するプログラムコードの命令を含むコンピュータプログラムであって、

各中央処理装置の仮想プロセッサ（24C，24S，26C，26S）は、前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する、ことを特徴とするコンピュータシステム。

(2) 本件補正によって補正された後の本願の特許請求の範囲の請求項1～10に係る発明（以下、請求項の番号を用いて「本願補正発明1」～「本願補正発明10」といい、これらを総称して「本願補正発明」という。）は、以下のとおりのものである（甲12。下線は、補正箇所を示す。）

#### 【請求項1】

命令スレッドの実行を管理するコンピュータシステム（18）であって、各中央処理装置が、前記各中央処理装置に属する複数の仮想プロセッサ（24C，24S，26C，26S）上で、複数の命令スレッドを組み合わせて実行するように構成される、少なくとも1つの中央処理装置（24，26）と、前記仮想プロセッサに命令スレッドの実行を分散するように構成されるマネージャ（38）と、を有するコンピュータシステム（18）において、  
前記命令スレッドには、少なくとも2つの異なる所定のタイプ（C，S）が存在し、  
－「計算」と呼ぶ命令スレッドの前記第1のタイプ（C）は、結果を供給するように前記コンピュータシステムが実行するようになっているプログラムに対して、該プログラムの実行に参加して結果を直接生成する命令スレッドに関係し、  
－「サービス」と呼ぶプロセスの前記第2のタイプ（S）は、該プログラムの実行に参加して前記「計算」と呼ぶタイプ（C）の前記命令スレッドに付属サービス（前記結果の保存など）を供給する命令スレッドに関係し、  
前記コンピュータシステム（18）は、実行する前記命令スレッドを前記所定のタ

タイプ (C, S) に応じて分類する分類手段 (36) を有し、  
前記命令スレッドの実行を分散するように構成される前記マネージャ (38) は、  
前記命令スレッドが分類された前記所定のタイプ (C, S) に基づき、実行する前  
記命令スレッドを前記仮想プロセッサ (24C, 24S, 26C, 26S) に方向  
付けるように設計され、  
前記各中央処理装置の仮想プロセッサ (24C, 24S, 26C, 26S) は、前  
記中央処理装置に属する内部資源を共有することにより、同時マルチスレッディング  
を実施する複数の論理区画に相当する、ことを特徴とするコンピュータシステム。

#### 【請求項2】

前記中央処理装置 (24, 26) の各仮想プロセッサ (24C, 24S, 26C, 26S) を前記所定のタイプ (C, S) の1つに関連付ける手段を備える、請求項1に記載のコンピュータシステムにおいて、前記命令スレッドの実行を分散するように構成される前記マネージャ (38) を、実行する前記命令スレッドをそれぞれ該スレッドと同じタイプの仮想プロセッサ (24C, 24S, 26C, 26S) に方向付けるように設計するコンピュータシステムであって、  
前記各中央処理装置の仮想プロセッサ (24C, 24S, 26C, 26S) は、前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する、ことを特徴とするコンピュータシステム。

#### 【請求項3】

前記中央処理装置 (24, 26) は2つの仮想プロセッサ (24C, 24S, 26C, 26S) を有し、1つは命令スレッドの第1のタイプ (C) に関連付け、もう1つは命令スレッドの第2のタイプ (S) に関連付け、実行するコンピュータプロセスはそれぞれ前記第1のタイプ (C) か前記第2のタイプ (S) である、請求項1または2に記載のコンピュータシステム。

#### 【請求項4】

- － 「計算」と呼ぶ命令スレッドの前記第1のタイプ (C) は、結果を供給するように前記コンピュータシステムが実行するようになっているプログラムに対して、該プログラムの実行に参加して結果を直接生成する命令スレッドに関係し、
- － 「サービス」と呼ぶプロセスの前記第2のタイプ (S) は、該プログラムの実行に参加して「計算」と呼ぶタイプ (C) の前記命令スレッドに付属サービス (前記結果の保存など) を供給する命令スレッドに関係する、請求項3に記載のコンピュータシステム。

#### 【請求項5】

少なくとも2つの中央処理装置 (24, 26) を有し、それぞれが複数の仮想プロセッサ (24C, 24S, 26C, 26S) を備え、仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する、請求項1から4のいずれか一項に記載のコンピュータシステムにおいて、前記命令スレッドの実行を分散するように構成される前記マネージャ (38) を、中央処理装置にある仮想プロセッサとは独立して、実行する命令スレッドそれぞれに対して中央処理装置を選択するように設計するコンピュータシステムであって、

前記各中央処理装置の仮想プロセッサ (24C, 24S, 26C, 26S) は、前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する、ことを特徴とするコンピュータシステム。

#### 【請求項6】

コンピュータシステム (18) の少なくとも1つの中央処理装置 (24, 26) に属する複数の仮想プロセッサ (24C, 24S, 26C, 26S) 上で複数の命令スレッドを組み合わせて実行するのを管理する方法であって、前記仮想プロセッサに前記命令スレッドの実行を分散するステップ (104) を含む方法において、複数の所定のタイプ (S, C) に応じて実行する前記命令スレッドを分類する予備ステップ (100) を含み、前記命令スレッドの実行を分散するステップ (104)



で、実行する命令スレッドそれぞれをタイプ (S, C) に応じて仮想プロセッサに方向付ける、方法であって、

前記各中央処理装置の仮想プロセッサ（24C, 24S, 26C, 26S）は、前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する、ことを特徴とする方法。

**【請求項7】**

前記分類ステップ（100）で、命令スレッドを管理するアプリケーションプログラミングインターフェースの関数でタイプ (S, C) を識別するパラメータに基づいて、前記命令スレッドをそれぞれタイプ (S, C) で分類する、請求項6に記載の方法。

**【請求項8】**

前記分類ステップ（100）で、命令スレッドの実行を含むプログラムの実行コマンドでタイプ (S, C) を識別するパラメータに基づいて、前記命令スレッドをそれぞれこのタイプ (S, C) で分類する、請求項6に記載の方法。

**【請求項9】**

スーパーコンピュータタイプのサーバクラスタ（10）の複数の処理ノード上で複数の命令スレッドを実行することの管理に対する請求項6から8のいずれか一項に記載の方法の適用であって、各処理ノード（18）が、少なくとも1つの中央処理装置に属する複数の仮想プロセッサ（24C, 24S, 26C, 26S）上で複数の命令スレッドを組み合わせて実行する前記少なくとも1つの中央処理装置（24, 26）を有する適用であって、

前記各中央処理装置の仮想プロセッサ（24C, 24S, 26C, 26S）は、前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する、ことを特徴とする適用。

**【請求項10】**

通信ネットワークからダウンロード可能なコンピュータプログラムおよび／また

はコンピュータで再生可能な媒体に記録したコンピュータプログラムおよび／またはプロセッサのなかでもとりわけコンピュータのオペレーティングシステム(34)で実行可能なコンピュータプログラムであって、前記プログラムをコンピュータ上で実行した場合に、請求項6から8のいずれか一項に記載の複数の命令スレッドを組み合わせた実行の管理方法のステップを実行するプログラムコードの命令を含むコンピュータプログラムであって、各中央処理装置の仮想プロセッサ(24C, 24S, 26C, 26S)は、前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する、ことを特徴とするコンピュータシステム。

### 3 審決の理由の要点

#### (1) 補正の適否

##### ア 新規事項

独立請求項である本件補正後の請求項6と、それを引用する本件補正後の請求項7及び8の「タイプ(S, C)」という記載は、平成24年4月17日付けで提出された明細書、請求の範囲の日本語による翻訳文及び国際出願の願書に添付された図面(以下、これらを併せて「当初明細書等」という。甲6)に記載されたものではなく、本件補正は、当初明細書等の記載の範囲内でされたものではない。

##### イ 目的要件

仮に、本件補正が、当初明細書等の記載の範囲内でされたものであるとしても、本件補正前の請求項6～8に対する本件補正における補正事項は、特許請求の範囲の限定的減縮、明瞭でない記載の釈明、誤記の訂正のいずれにも該当しない。

そして、本件補正において、請求項の削除は行われていない。

##### ウ 独立特許要件

仮に、本件補正が、当初明細書等の記載の範囲内でされたものであって、目的要件を満たすものであったとしても、本願補正発明6と甲1(特開2004-220

608号公報)に記載された発明(以下、「甲1発明」という。)との一致点及び相違点は、以下のとおりであり、相違点1～4に係る事項は、甲2(特開2004-348462号公報)に記載された発明及び当該技術分野における周知の技術事項(甲3～5)に基づいて当業者が適宜なし得る事項であるから、本願補正発明6は、これらの発明及び周知の技術事項に基づいて当業者が容易に発明することができた。

したがって、本願補正発明6は、特許出願の際独立して特許を受けることができない。

#### 【一致点】

コンピュータシステムの少なくとも1つの中央処理装置上で複数の命令スレッドを組み合わせることを実行するのを管理する方法であって、

資源に命令スレッドの実行を分散するステップを含む方法において、複数の所定のタイプ(S, C)に応じて実行する前記命令スレッドを分類する予備ステップを含み、前記命令スレッドの実行を分散するステップで、実行する命令スレッドそれぞれをタイプ(S, C)に応じて資源に方向付ける、方法。

#### 【相違点1】

“中央処理装置上で複数の命令スレッドを組み合わせる”に関して、

本件補正発明6においては、“中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせる”ものであるのに対して、

引用発明においては、「仮想プロセッサ上で複数の命令を実行する」点については、特に言及されていない点。

#### 【相違点2】

“資源”に関して、

本件補正発明6においては、“資源”が、「仮想プロセッサ」であるのに対して、引用発明における“資源”である「リソース」に、「仮想プロセッサ」が含まれることが明言されていない点。

#### 【相違点3】

“実行する命令スレッドそれぞれをタイプ（S，C）に応じて資源に方向付ける”  
ことに関して、

本件補正発明6においては「仮想プロセッサに方向付ける」ものであるのに対して、

引用発明においては、「リソースの共通セットに割り当てられる」ものである点。

#### 【相違点4】

本件補正発明6においては、「各中央処理装置の仮想プロセッサ（24C，24S，26C，26S）は、前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する」ものであるのに対して、

引用発明においては、「仮想プロセッサ」についての言及がない点。

#### (2) 進歩性

本願発明6と甲1発明との一致点及び相違点は、以下のとおりであり、相違点a～dに係る事項については、上記(1)の相違点1～4に係る事項において検討したとおりであるから、本願発明6は、甲1発明、甲2に記載された発明及び当該技術分野における周知の技術事項（甲3～5）に基づいて当業者が容易に発明をすることができた。

#### 【一致点】

コンピュータシステムの少なくとも1つの中央処理装置上で複数の命令スレッドを組み合わせて実行するのを管理する方法であって、

資源に命令スレッドの実行を分散するステップを含む方法において、複数の所定のタイプに応じて実行する前記命令スレッドを分類する予備ステップを含み、前記命令スレッドの実行を分散するステップで、実行する命令スレッドそれぞれをタイプに応じて資源に方向付ける、方法。

#### 【相違点a】

“中央処理装置上で複数の命令スレッドを組み合わせて実行する”に関して、

本願発明 6 においては，“中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する”ものであるのに対して，

引用発明においては，「仮想プロセッサ上で複数の命令を実行する」点については，特に言及されていない点。

**【相違点 b】**

“資源”に関して，

本願発明 6 においては，“資源”が，“仮想プロセッサ”であるのに対して，

引用発明における“資源”である「リソース」に，“仮想プロセッサ”が含まれることが明言されていない点。

**【相違点 c】**

“実行する命令スレッドそれぞれをタイプに応じて資源に方向付ける”ことに関して，

本願発明 6 においては「仮想プロセッサに方向付ける」ものであるのに対して，

引用発明においては，“リソースの共通セットに割り当てられる”ものである点。

**【相違点 d】**

本願発明 6 においては，“各中央処理装置の仮想プロセッサ（24C，24S，26C，26S）は，前記中央処理装置に属する内部資源を共有することにより同時マルチスレッディングを実施する機能を有する複数の論理区画に相当する”ものであるのに対して，

引用発明においては，“仮想プロセッサ”についての言及がない点。

**第 3 原告主張の審決取消事由**

**1 補正の適法性についての判断の誤り**

**(1) 新規事項追加禁止要件違反についての判断の誤り**

次のとおり，本件補正によってされた請求項 6 における「タイプ」を「タイプ(S，C)」に変更する補正事項は，当初明細書等に明示的に記載されているか，少なくとも当初明細書等の記載から自明なものであり，新規事項追加禁止要件を充たしてい

るから、請求項6を引用する請求項7及び8についても、上記要件を満たしている。

ア(ア) a 請求項6において、「タイプ(S, C)」は、①「複数の所定のタイプ(S, C)に応じて実行する前記命令スレッドを分類する」及び②「実行する命令スレッドそれぞれをタイプ(S, C)に応じて仮想プロセッサに方向付ける」という発明特定事項において用いられている。

当初明細書等の【0039】の記載を考慮すると、①「複数の所定のタイプ(S, C)に応じて実行する前記命令スレッドを分類する」という発明特定事項は、命令スレッドのタイプ(例えば「計算」と「サービス」と呼ばれる二つのタイプ)に応じて、実行する命令スレッドを分類する、という意味であることが容易に理解できる。

当初明細書等の【0047】の記載を考慮すると、②「実行する命令スレッドそれぞれをタイプ(S, C)に応じて仮想プロセッサに方向付ける」という発明特定事項は、実行する命令スレッドを、当該命令スレッドのタイプ(例えば「計算」と「サービス」と呼ばれる二つのタイプ)に応じて仮想プロセッサに方向付ける、という意味であることが理解できる。

以上の理解に基づくと、これらの発明特定事項において用いられている「タイプ(S, C)」は、「命令スレッドのタイプ(S, C)」を意味することも理解できる。

b 当初明細書等において、「S」及び「C」が仮想プロセッサのタイプを表すために用いられていることはあるにしても、命令スレッドの第1のタイプ(計算タイプ)を表すために「C」、命令スレッドの第2のタイプ(サービスタイプ)を表すために「S」が用いられている(甲6【請求項3】、【請求項4】、【図1】)。

【図 1】

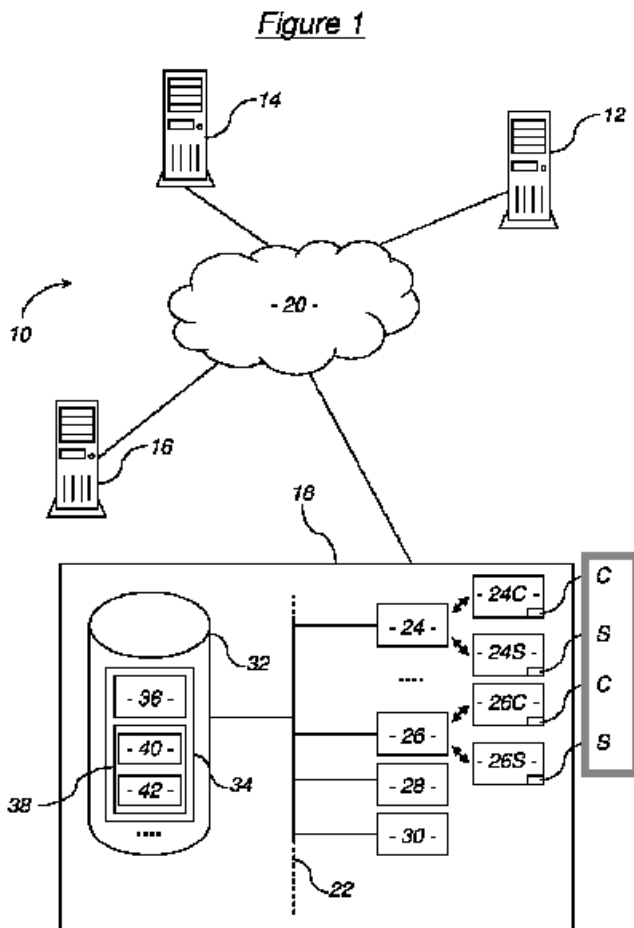


図 1 においては、仮想プロセッサ 24 C, 24 S, 26 C, 26 S を表すブロックの各々の右下隅に小さなブロックが描かれている。そして、仮想プロセッサ 24 C 及び仮想プロセッサ 26 C の右下隅のブロックにはそれぞれ参照符号「C」が付され、仮想プロセッサ 24 S 及び仮想プロセッサ 26 S の右下隅のブロックにはそれぞれ参照符号「S」が付されている。仮想プロセッサを示す符号としては、既に 24 C, 24 S, 26 C, 26 S が用いられているから、上記の小さい方のブロックの各々に付されている参照符号は仮想プロセッサのタイプを示すものではない。

他方、請求項 3 には、「前記中央処理装置 (24, 26) は 2 つの仮想プロセッサ

(24C, 24S, 26C, 26S)を有し、一つは命令スレッドの第1のタイプ(C)に関連付け、もう一つは命令スレッドの第2のタイプ(S)に関連付け」と記載されている。この請求項3の記載に沿って図1を理解すると、図1の仮想プロセッサ24C, 26Cを表すブロックの各々の右下隅にあり「C」の符号が付されている小ブロックは、命令スレッドの第1のタイプ(C)を表していると容易に理解することができる。符号「S」についても同様である。

(イ) 括弧書きされている「(S, C)」は、請求項の「タイプ」という用語と発明の詳細な説明や図面に記載された具体的な構成との対応関係を明確にするために、「タイプ」の後の括弧内に発明の詳細な説明や図面で使用されている参照符号「S」及び「C」を付加したものである。

当初明細書等の【0046】、【0047】及び【図1】において、計算タイプを表す符号として「C」が、サービスタイプを表す符号として「S」が用いられている。

したがって、「タイプ」に「(S, C)」を付加することにより、「タイプ(S, C)」が発明の詳細な説明や図面におけるサービスタイプ及び計算タイプに対応することが特定されている。

参照符号の付加により当初明細書等の記載から導き出すことができないような技術事項が導入されることはない。

イ(ア) 本件補正による①「複数の所定のタイプ (S, C) に応じて実行する前記命令スレッドを分類する」という補正事項は、当初明細書等の記載から自明なものである。

当初明細書等の請求項4及び【0039】の記載に鑑みると、当業者は、当初明細書等には、①「命令スレッドの第1のタイプ(C)及び命令スレッドの第2のタイプ(S)という二つの命令スレッドのタイプに応じて、実行する命令スレッドを分類する」ことが記載されていることを理解することができる。

(イ) 本件補正による②「実行する命令スレッドそれぞれをタイプ (S, C)



に応じて仮想プロセッサに方向付ける」という補正事項は、当初明細書等の記載から自明なものである。

当業者は、発明の詳細な説明において命令スレッドのタイプについて「C」や「S」の符号を付さずに説明がされていても、命令スレッドのタイプとしては第1のタイプ（C）及び第2のタイプ（S）があることを当然に理解できる。

そうすると、当業者は、当初明細書等の【0047】の記載から、当初明細書等には、②「実行する命令スレッドをそれぞれタイプに応じて1つの仮想プロセッサに方向付けし、さらに正確にはタイプが同じ仮想プロセッサにも方向付けするように設計する。」ことが記載されていると理解することができる。

ウ(ア) 被告の主張によると、本件補正後の「タイプ（S，C）」という用語の意義は、本件補正前の「タイプ」という用語の意義と全く同じということになる。

請求項には、「特許出願人が特許を受けようとする発明を特定するために必要と認める事項のすべて」（特許法36条5項）が記載されているのであるから、請求項に記載された事項は全て発明を特定するために必要な事項であって、このような発明を特定するための事項を無視して発明の要旨認定を行うことは許されない。

(イ) 特許法施行規則様式29条の2の[備考]14のロは、発明の要旨認定の準則とされる規定ではない。特許法36条5項によると、請求項に記載された符号も当該請求項に係る発明を特定するために必要な事項であり、このような符号の記載も考慮した上で発明の要旨を認定すべきである。

乙1は、請求項において図面に対応する符号が用いられたからといって、当該請求項に係る発明を当該符号が用いられた図面に対応する実施例に限定して解釈されることはないということを説明しているにすぎず、当該符号を考慮せずに要旨認定を行うことを説明しているのではない。

## (2) 目的要件違反についての判断の誤り

次のとおり、本件補正における①「複数の所定のタイプに応じて実行する前記命令スレッドを分類する」及び②「実行する命令スレッドそれぞれをタイプに応じて仮想

プロセッサに方向付ける」の「タイプ」を「タイプ（S，C）」と変更する補正は、特許請求の範囲の限定的減縮に当たり、本願補正は、特許請求の範囲の限定的減縮を目的としたものであるから、目的要件を満たしている。

ア 「タイプ」を「タイプ（S，C）」と変更する本願補正は、元々記載されていた命令スレッドの「タイプ」に符号「（S，C）」を付加して、請求項の「タイプ」という用語と発明の詳細な説明や図面に記載された具体的な構成との対応関係をより明確にしたものである。

本願補正前の「タイプ」は、命令スレッドのタイプという意味を有していたが、これを「タイプ（S，C）」と補正することにより、請求項6で用いられている「タイプ」が発明の詳細な説明や図面に記載された「サービスタイプ」や「計算タイプ」と対応することがさらに特定された。

イ(ア) 当初明細書等の【0039】には、命令スレッドの複数の所定のタイプとして、第1のタイプ（第1のタイプ（C））及び第2のタイプ（第2のタイプ（S））が挙げられているが、「複数の所定のタイプに応じて、また特に、」と記載されていることから、この第1のタイプ（C）及び第2のタイプ（S）は例示であることがわかる。

したがって、命令スレッドの「複数の所定のタイプ」には、第1のタイプ（C）及び第2のタイプ（S）以外のタイプも含まれることが想定されている。

(イ) 本件補正前の①「複数の所定のタイプに応じて実行する前記命令スレッドを分類する」という発明特定事項は、

A1：第1のタイプ（C）や第2のタイプ（S）を含む複数の所定のタイプに応じて実行する命令スレッドを分類するという態様、及び

A2：第1のタイプ（C）や第2のタイプ（S）を含まない複数の所定のタイプに応じて実行する命令スレッドを分類するという態様、  
のいずれの態様も包含している。

本件補正後の請求項6においては、「複数の所定のタイプ」が当初明細書等の第1

のタイプ（C）及び第2のタイプ（S）に対応するものであることが特定された。

したがって、本件補正により、前記態様A 2が特許請求の範囲から除外されたことになる。

(ウ) 本件補正前の②「実行する命令スレッドそれぞれをタイプに応じて仮想プロセッサに方向付ける」という発明特定事項は、

B 1：実行する命令スレッドを第1のタイプ（C）や第2のタイプ（S）を含むタイプに応じて仮想プロセッサに方向付けるという態様、及び

B 2：第1のタイプ（C）や第2のタイプ（S）を含まないタイプに応じて仮想プロセッサに方向付けるという態様、  
のいずれの態様も包含している。

本件補正後の請求項6においては、「タイプ（S，C）」が当初明細書等の第1のタイプ（C）及び第2のタイプ（S）に対応する概念であることが特定された。

したがって、本件補正により、前記態様B 2が特許請求の範囲から除外されたことになる。

ウ 本件補正の前後で、請求項6～8に係る発明の産業上の利用分野及び解決課題がそれぞれ同一であることも明らかである。

エ 仮に、前記内容の本件補正により、補正後の内容が補正前の内容から実質的に変更されていないとしても、当該補正事項は、特許請求の範囲の限定的減縮に該当する。

すなわち、一般に、審判請求時の補正において特許請求の範囲の限定的減縮を目的とする補正が許容されているのは、このような補正は、審査・審理の対象を大幅に変更するものではなく従前の審査結果を利用して効率的に審査・審理を行うことができるためである。

上記の補正事項に係る補正は、審査・審理の対象を大幅に変更するものではないから、従前の審査結果を利用することが可能であり、特許法によって許容された補正といえる。

(3) 独立特許要件違反についての判断の誤り

ア 甲1発明において相違点3に係る発明特定事項を採用することは容易になし得たことではないこと

(ア) 審決の甲2に記載された技術事項に関する認定は誤っていること

a 甲2に記載されているのは、模擬対象の模擬処理を、模擬イベント交換頻度及び模擬データ転送量に応じて、論理プロセッサに割り付ける分散シミュレーションシステムであり、甲2には、この技術について、次の点が開示されている。

(a) 甲2に記載された技術は、訓練用シミュレーション、計画・予測・解析用シミュレーションなど、様々な用途のシミュレーションを実行する分散シミュレーションシステムに関するものである。

(b) 従来から、分散シミュレーションシステムを用いて、地面や路面などの模擬空間における地形や地勢を模擬することが行われていた。

(c) 地表面、車両、航空機などの模擬対象に応じてシミュレーションプログラムを分割して複数の論理プロセッサで分散処理する場合に、分割されたプログラムを実行する論理プロセッサ間で頻繁にデータをやり取りする必要があるため、論理プロセッサ間で大きなサイズのデータをやり取りする必要がある場合や、データ交換頻度が非常に高い場合などでは、システム全体として当該シミュレーションを高速に処理することが困難になる。

地表面のような模擬環境と車両や航空機などの移動する模擬物体とでは、互いに関連しあうデータを常にやり取りする必要がある。このため、これらに対応するプログラムを実行する論理プロセッサ間では、データ交換頻度が非常に高くなり、やり取りすべきデータ量も膨大なものとなる。

模擬対象とされた地表面が非常に広範囲であったり、模擬対象とすべき車両や航空機などの物体が非常に多数であると、これらに対応した特定のプログラムの負荷が高くなってしまうため、システム全体として処理速度が落ちてしまう。

(d) 甲2に記載された技術は、上記課題を解決するためのものである。具体的には、甲2記載の分散シミュレーションシステムは、模擬対象定義から互いに関連し合う挙動が模擬される模擬対象の組合せを求め、これらの組合せに関連する領域ごとにその模擬処理を論理プロセッサに分担する。模擬環境や模擬物体などの模擬対象について模擬イベント交換頻度と模擬データ転送量が高くなると予想される組合せを同一の論理プロセッサに割り当てる。

(e) 甲2に記載された技術は、論理プロセッサ間における模擬イベント交換頻度と模擬データ転送量を削減することができ、高速で効率的な処理を実現するという効果を奏するためのものである。

b 甲2の【0033】～【0035】を考慮しても、甲2には、上記(ア)の技術を一般化して、種類の異なる複数の処理を論理プロセッサに割り付ける技術は、記載も示唆もされていない。

また、甲2には、模擬イベント交換頻度及び模擬データ転送量に応じた論理プロセッサへの割当ては記載されているが、これを一般化した、処理の種類に応じたプロセッサへの割当てについては、記載も示唆もされていない。

(イ) 甲2に記載された技術を甲1発明には容易に適用できないこと

a 技術分野が異なること

(a) 甲1発明は、マルチスレッド・コンピュータにおけるプログラムの最小の実行単位であるスレッドのコンピュータリソースへの割当てに関する発明であるのに対し、甲2に記載された技術は、訓練用シミュレーション等の各種シミュレーションを実行する分散シミュレーションシステムにおける模擬処理の論理プロセッサへの割当てに関するものであるから、甲1発明と甲2に記載された技術とは技術分野が異なる。

したがって、甲1発明に対して、甲1発明とは異なる分散シミュレーションシステムでの模擬処理の分担に関する技術を適用する動機付けが存在せず、そのような適用を行うことが容易であったとはいえない。

(b) 確かに、甲1発明及び甲2に記載された技術は、いずれも、命令を複数のスレッドに分割して実行するマルチスレッディングに関するものである点では共通する。

しかし、マルチスレッディングには、(A)複数の物理プロセッサにスレッドを同時に実行させるマルチプロセッサ技術と、(B)単一の物理プロセッサによって認識される複数の仮想プロセッサにスレッドを同時に実行させるマルチスレッド同時処理(SMT)技術があり(甲4【0004】)、これらはマルチスレッディングにおける別種のアプローチである。この両技術は、その定義上、マルチプロセッサ技術においてはスレッド処理の効率性が複数の物理プロセッサの物理的な関係に影響されるのに対し、SMT技術ではそのようなことがない点で異なっており、それに伴って対処すべき技術的課題も異なる。

甲1発明は、複数の物理プロセッサ間相互のデータ転送に起因する通信コストの問題を解決するために、命令スレッドをリソースの共通セットに割り当てる、すなわち、同一型の命令スレッドをコンピュータの同一物理サブシステムのリソースに割り当てるという構成を採用し、複数の物理サブシステム間における交差トラフィックを減少させており、前者のマルチプロセッサ技術に関するものであるのに対し、甲2に記載された技術は、後者のマルチスレッド同時処理(SMT)技術に関するものである。

「同一タイプのスレッドが、リソースの共通セットに割り当てられる」という構成は、複数の物理プロセッサ間相互のデータ転送に起因する通信コストの問題を解消するというマルチプロセッサ技術に固有の技術課題を解決するために甲1発明において採用されたマルチプロセッサ技術と密接に関連する構成であるから、当業者は、このような構成をSMT技術の構成で置き換えることには動機付けられない。すなわち、SMT技術においては、複数の仮想プロセッサは単一の物理プロセッサにより実現されることが前提であって、複数の物理プロセッサ間の通信コストが課題とならないから、甲1発明において複数の物理プロセッサ間の通信コストに関する

る技術課題を解決する前に不可欠な構成を，SMT技術に関連する構成で置換することに動機付けられるとは考えられない。

b 解決すべき課題が異なること

(a) 甲1発明の課題は，主に，処理対象のスレッドにプロセッサ等の物理的リソースを効率的に割り当てて，複数のプロセッサ間におけるデータ転送に関する通信コストを解決すること，及び，その際に著しいカスタム化を行わないようにすること，という物理プロセッサ間で処理を分散して行うことを前提としたものであるのに対し，甲2に記載された技術に関する課題は，模擬対象に応じて分割したシミュレーションプログラムを複数の論理プロセッサで分散処理するに当たり，主に，論理プロセッサ間での模擬イベント交換頻度及び模擬データ転送量を削減し，システム全体としての処理速度を高速に保ち，かつ効率的な模擬処理を実行することができる分散シミュレーションシステムを得ることであるから，物理プロセッサ間での通信コストの問題とは無関係である。

したがって，甲1発明と甲2に記載された技術とは解決すべき課題が異なっており，甲2に記載された技術事項を甲1発明に適用する動機付けが存在しない。

(b) 本件補正発明においては，仮想プロセッサへの命令スレッドの割当てに関して，ロードバランシングアルゴリズムの実行のために仮想プロセッサの特性を考慮する必要があるという課題（当初明細書等【0007】）や，プログラムがスレッドに対するスケジューラの挙動を明示する際に，コンピュータシステムのトポロジーの知識が必要になり，また，ソフトウェア層との競合が生じるおそれのある配置作業を行わなければならないという課題（当初明細書等【0010】）に着目し，その解決のために相違点3に係る構成，すなわち，「実行する命令スレッドそれぞれをタイプ（S，C）に応じて仮想プロセッサに方向付ける」という構成を採用している。このような課題は，甲1～5のいずれにも記載されていない。

当業者は，このような仮想プロセッサへ命令スレッドを割り当てる際の課題についての認識がなければ，甲1発明においてスレッドをそのタイプに応じて仮想プロ

セッサに割り当てるという構成を採用することには動機付けられない。

また、甲1発明において、実行する命令スレッドをタイプに応じた特定の物理サブシステムに割り当てるとは、通信コストの問題を解消するという甲1発明の技術課題を解決するために必要不可欠な構成である。この構成に代えて、スレッドをそのタイプに応じて仮想プロセッサに割り当てるとは、甲1発明の課題が解決できなくなるため、このような変更をすることはできない。

c 甲2に記載された技術を甲1発明に適用することが可能であるとしても容易に想到し得ないこと

甲2に記載された技術事項は、模擬対象の模擬処理を、模擬イベント交換頻度及び模擬データ転送量に応じて、論理プロセッサに割り付けることである。命令スレッドを論理プロセッサに割り付けることは、甲2には記載されていない。

甲2の模擬処理がスレッド単位で実行されるとしても、甲2には、その模擬処理を実行するスレッドを論理プロセッサに割り当てることは記載も示唆もされていない。むしろ、甲2の【0017】、【0033】、【0035】、【0037】、【0039】、【0040】、及び【0050】には、模擬対象の組合せを同一の論理プロセッサに割り当てることが記載されているのであって、甲2に記載された技術では、論理プロセッサへの割当ては、組み合わせられた模擬対象の模擬処理を単位として行われている。

そうすると、仮に甲2に記載された技術を甲1発明に適用することが可能であるとしても、組み合わせられた模擬対象の模擬処理を単位として論理プロセッサへの割当てがなされるにすぎず、その模擬処理に含まれるスレッドをそのタイプに応じて仮想プロセッサに割り付けることは想到し得ない。

また、甲2には、模擬イベント交換頻度及び模擬データ転送量に応じた論理プロセッサへの割当ては記載されているが、これを一般化した、処理の種類に応じたプロセッサへの割当てについても記載も示唆もされていない。

d 甲1発明においてSMT技術を採用することができないとしても相



違点3に係る構成をなるように変更する動機付けがないこと

甲1発明においてSMT技術を採用することができたとしても、甲1発明における「複数の物理サブシステム」に備えられた「マイクロプロセッサ」の各々において複数のスレッドが並列に実行されるだけであり、甲1発明の「同一タイプのスレッドが、リソースの共通セットに割り当てられる」という構成を相違点3に係る構成となるように変更する動機付けは存在しない。

e 甲3には相違点3に係る構成は開示されていないこと

甲3には、「リアルタイムビデオアプリケーションにおいて、ビデオを取り込む一つのスレッド、ビットストリームを符号化する一つのスレッド、およびビットストリームを送り出す他のスレッドが存在する」ことは記載されているが、これらのスレッドがどのように仮想プロセッサに割り当てられるかについては全く開示がない。図4Aを参照すると、二つのスレッドが並列に処理されていることは理解できるが、この二つのスレッドがどのようにして仮想プロセッサに割り当てられているかは不明である。

甲3は、図6を参照しつつ【0054】～【0059】において、ビデオの復号化プロセスを説明しているが、この説明においてもスレッドをどのように仮想プロセッサに割り当てるとは一切説明されていない。

したがって、甲3には、実行する命令スレッドをそのタイプに応じて仮想プロセッサに方向付けるという相違点3に係る構成は開示されていない。

イ 甲1発明に甲2に記載された技術を適用して相違点1に係る発明特定事項を採用することは容易になし得たことではないこと

(ア) 前記ア(ア)のとおり、甲2には、模擬イベント交換頻度及び模擬データ転送量に応じた論理プロセッサへの割当ては記載されているが、これを一般化した、処理の種類に応じたプロセッサへの割当てについては、記載も示唆もされていない。

(イ) 甲1発明と甲2に記載された技術とでは、技術分野が大きく異なる。

甲1発明と甲2に記載された技術とでは、リソースへの割当てが行われる処理の単位が異なるだけでなく、甲1発明においては物理的なリソースへの割当てがされるのに対して、甲2の技術では論理プロセッサという論理的なリソースへの割当てがされる点も異なる。

(ウ) 前記ア(イ) b (a) のとおり、甲1発明の課題と甲2に記載された技術に関する課題とは異なる。

(エ) したがって、甲2に記載された技術事項を甲1発明に適用する動機付けが存在しない。

## 2 進歩性に関する判断の誤り

仮に本件補正が却下されたとしても、本件補正発明6と甲1発明の相違点cは、相違点3と、相違点aは、相違点1と実質的に同じであるため、前記の相違点1及び3に関する主張は、すべて相違点c及びaにも当てはまり、甲1発明において、相違点a及びcに係る構成を採用することは当業者にとって容易ではなかった。

したがって、本願補正発明6は、進歩性を有する。

## 第4 被告の主張

### 1 取消事由1について

#### (1) 新規事項追加禁止要件違反について

ア 次のとおり、請求項6～8において、「タイプ(S, C)」は、スレッドのタイプを識別する情報として用いられている一方、当初明細書等では、仮想プロセッサのタイプを識別する情報(パラメータC及びS)として用いられており、一致しないから、補正後の請求項6～8は、当初明細書等に記載されたものではない。

当初明細書等の【0046】の記載から、本願明細書の発明の詳細な説明において、「C」及び「S」とは、仮想プロセッサのタイプを識別するためのパラメータ「C」及び「S」であることが読み取れる。

当初明細書等の【0039】は、命令スレッドのタイプとして、『計算』と呼ぶ命令スレッドの第1のタイプ』および『サービス』と呼ぶプロセスの第2のタイプ』

(ここで、「プロセス」は「命令スレッド」の誤記であると推察される。)を例示しており、「タイプ(S, C)」との表記を、命令スレッドのタイプとして用いていない。

当初明細書等の【0047】には、「仮想プロセッサ24C, 24S, 26Cおよび26S」という記載があり、ここでも、「C」又は「S」は、仮想プロセッサのタイプを示している。さらに、同段落の「仮想プロセッサ24Cまたは26Cには計算タイプの命令スレッドしか方向付けできず、仮想プロセッサ24Sまたは26Sにはサービスタイプの命令スレッドしか方向付けできない。」という記載は、仮想プロセッサのタイプとして「C」又は「S」を用いており、命令スレッドのタイプとしては、「計算タイプ」又は「サービスタイプ」を用いており、両者は明確に使い分けられている。

当初明細書等の【図1】の符号「C」及び「S」は、プロセッサを計算タイプ又はサービスタイプに関連付けるためのパラメータC及びSであり、スレッドのタイプそのものを示す符号ではない。

イ 一般に、参照符号は、明細書や図面等に開示された技術的構成との対応関係を例示するものであり、請求項に記載された発明特定事項を技術的に限定するものではない。

特許法施行規則24条の4は「願書に添付すべき特許請求の範囲は、様式第二十九の二により作成しなければならない。」と規定しており、特許法施行規則において、様式第29の2(第24条の4関係)として、特許請求の範囲の様式が規定されている。特許法施行規則様式29の2の[備考]14のロに規定された「請求項の記載の内容を理解するため必要があるときは、当該願書に添付した図面において使用した符号を括弧をして用いた符号が特許請求の範囲の記載に付加されることによって、請求項記載の発明が具体的構成に限定されるものではない(乙1)。

請求項6の「少なくとも1つの中央処理装置(24, 26)」という記載は、発明特定事項としては「少なくとも1つの中央処理装置」としつつも、「中央処理装置(2

4, 26)」と複数の中央処理装置を例示しており、一般的な意味の参照符号であるといえる。そして、請求項6の「複数の仮想プロセッサ(24C, 24S, 26C, 26S)」という記載も、発明特定事項としては「複数の仮想プロセッサ」としつつも、「仮想プロセッサ(24C, 24S, 26C, 26S)」と4つの仮想プロセッサを例示しており、一般的な意味の参照符号であるといえる。

括弧書きされている(S, C)も参照符号であるとする、「タイプ(S, C)」は、発明特定事項としては「タイプ」であって、「(S, C)」は単なる例示にすぎないものとなる。

## (2) 目的要件違反について

ア 前記(1)のとおり、S又はCは、仮想プロセッサのタイプを表す情報であるから、「タイプ」を「タイプ(S, C)」と変更する補正は、本願の明細書等の誤った解釈に基づいて誤記を導入するものであって、特許請求の範囲を減縮するものとはいえない。

命令スレッドのタイプを特定するのであれば、「タイプ(S, C)」ではなく、「タイプ(計算タイプ, サービスタイプ)」とすべきであった。

イ 本件補正後の請求項6における「複数の所定のタイプ」という用語は、「命令スレッド」の「タイプ」を意味するものとして明確であり、「複数の所定のタイプ」に付加された「(S, C)」も、本願の願書に添付した図面において使用した符号を括弧をして用いたものとして解釈し得る。

「命令スレッド」の「タイプ」として、例えば、データの入出力等を行う「タイプ」の「命令スレッド」であるとか、浮動小数点演算を行う「タイプ」の「命令スレッド」であるとか、制御フローを担当する「タイプ」の「命令スレッド」などがさまざまな「タイプ」の「命令スレッド」が含まれ得ることは、当事者にとって自明である。

本願発明のマルチスレッディングコンピュータシステムの技術分野において、「命令スレッド」の「タイプ」として、「S」及び「C」以外のタイプはおよそ用いられ

るものでないことが当業者の一般常識になっているなどの特段の事情があるとはいえないから、明細書の発明の詳細な説明で技術的に裏付けられているのが「S」及び「C」のタイプを使用するものだけであるとか、実施例が「S」及び「C」のタイプを使用するものだけであることのみから、特許請求の範囲に記載された「複数の所定のタイプ」を「少なくとも『S』及び『C』のタイプを含むもの」と限定して解することはできないというべきである。

ウ 仮に、「複数の所定のタイプ(S, C)」という表記において、「(S, C)」が参照符号であり、命令スレッドのタイプ例示に過ぎず、命令スレッドのタイプを技術的に限定するものでなければ、「(S, C)」という記載を除いて発明特定事項を把握可能であり、特許請求の範囲の限定的減縮に該当する余地はあるといえる。

しかし、上記アのとおり、「タイプ」を「タイプ(S, C)」と変更する補正は、本願明細書の誤った解釈に基づいて誤記を導入するものであるから、「(S, C)」は命令スレッドのタイプの例示とはいえない。

### (3) 独立特許要件違反について

ア 次のとおり、相違点3に係る構成は、甲1発明と甲2に記載された技術又は甲3～5により認定可能な周知技術であるSMT技術を適用することにより当業者が容易に想到し得たものである。

(ア) 命令を複数のスレッドに分割して実行するマルチスレッディング(マルチスレッド・コンピューティング)の技術分野において、仮想プロセッサ(論理プロセッサ)は、各々アーキテクチャステート(アーキテクチャ状態)を備えることにより、一つの物理プロセッサに複数の仮想プロセッサ(論理プロセッサ)が認識されており、キャッシュや実行ユニット、分岐予測器、制御論理等の物理プロセッサが有するリソース群を共有しており、複数のスレッドが複数の仮想プロセッサに割り当てられて同時に実行されるSMT(同時マルチスレッディング、マルチスレッド同時処理)技術は、本願優先日前に周知であったといえる(甲3～5)。

(イ) 引用発明と上記周知技術(SMT技術)は、命令を複数のスレッドに

分割して実行するマルチスレッディング（マルチスレッド・コンピューティング）の技術分野に属しており，当該技術分野では，スレッドを実行する主体として仮想プロセッサに割り当てること，及び，各仮想プロセッサは，キャッシュや実行ユニット，分岐予測器，制御論理等の物理プロセッサが有するリソース群を共有していることは，本願優先日前に周知の技術であったといえる。

(ウ) 上記周知技術（SMT技術）の「仮想プロセッサ」が共有する「物理プロセッサが有するリソース群」は，引用発明の「リソースの共通セット」に対応する。

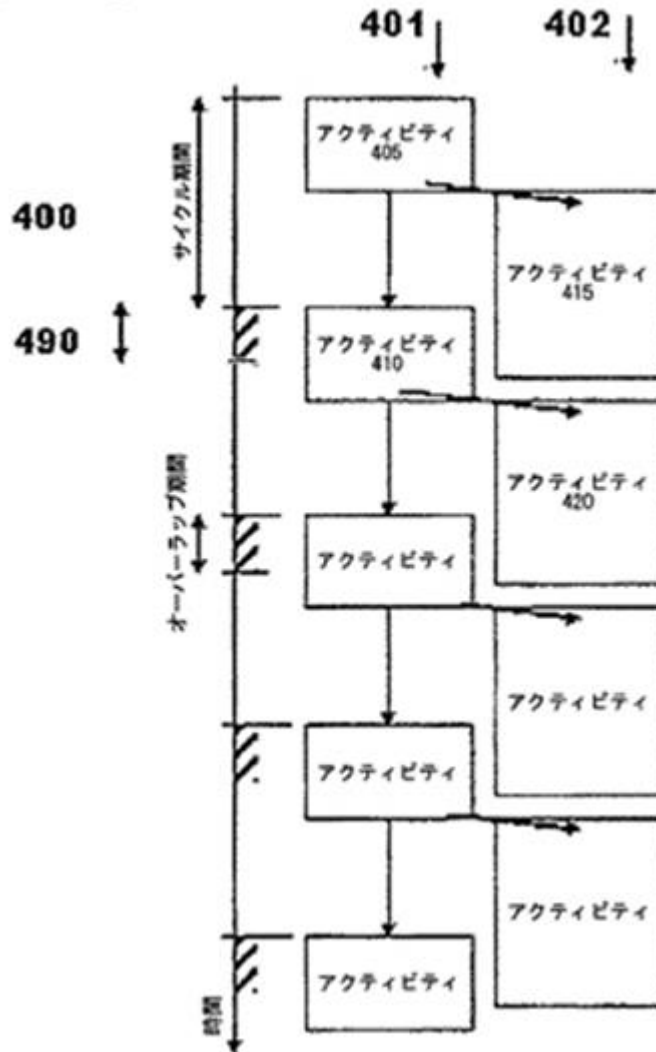
(エ) したがって，引用発明において，実行する命令スレッドをタイプに応じて割り当てる（方向付ける）際に，リソースの共通セットに割り当てる構成に代えて，上記周知技術（SMT技術）を適用して，命令スレッドを物理プロセッサが有するリソース群からなる仮想プロセッサに割り当てる（方向付ける）よう構成することは，当業者が容易に想到し得たものである。

イ 仮に，相違点3に係る「実行する命令スレッドそれぞれをタイプ（S，C）に応じて資源に方向付ける」という構成を「実行する命令スレッドそれぞれを（少なくとも計算タイプおよびサービスタイプなどの）タイプに応じて資源に方向付ける」と限定するものと解釈した場合であっても，相違点3に係る構成は，甲1発明と甲3に記載された技術事項から，当業者が容易に想到し得たものである。

(フ) 甲3（【0039】～【0041】，【図4A】）には，「リアルタイムビデオアプリケーション」において，三つのスレッド（「ビデオを取り込む1つのスレッド，ビットストリームを符号化する1つのスレッド，およびビットストリームを送り出す他のスレッド」）が例示されており，同時マルチスレッディングの一例とされている。そして，【図4A】のアクティビティ405，アクティビティ410などを有するスレッド401は，「ビデオを取り込む1つのスレッド」であり，【図4A】のアクティビティ415，アクティビティ420などを有するスレッド402は，「ビットストリームを符号化する1つのスレッド」である。これらのスレッド40

1 とスレッド 402 は、異なる 2 つの論理プロセッサに割り当てられて実行されることにより、オーバーラップ期間の間、同時に実行されている。

【図 4 A】



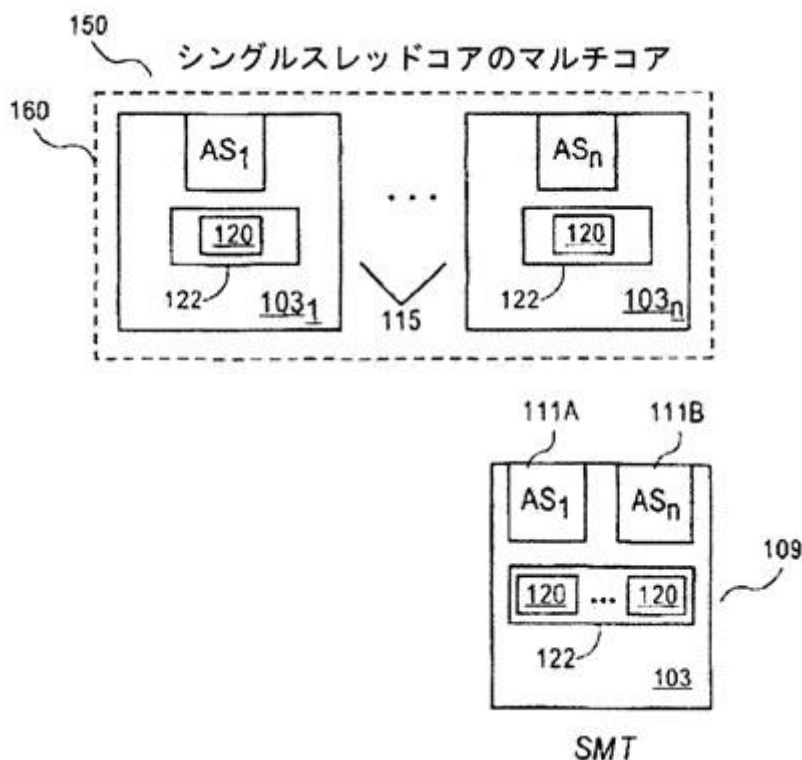
(イ) ここで、「ビットストリームを符号化する 1 つのスレッド」は、本願明細書の「計算タイプの命令スレッド」に相当し、「ビデオを取り込む 1 つのスレッド」は本願明細書の「サービスタイプの命令スレッド」に相当するといえる。

1 つの物理プロセッサ内に認識される複数の仮想プロセッサは、バッファやキャッシュ等のリソースを共有しており、「ビデオを取り込む 1 つのスレッド」を一つの仮想プロセッサに割り当て、「ビデオを取り込む 1 つのスレッド」がバッファ等に取り

り込んだビデオフレームデータを、「ビットストリームを符号化する1つのスレッド」を別の仮想プロセッサに割り当て、当該別の仮想プロセッサでデコード処理を実行するように構成することは、サービスタイプの命令スレッドと計算タイプの命令スレッドを別々の仮想プロセッサに割り当て実行させていることにほかならない。

ウ 甲4には、マルチスレッディングを実行するシステム構成として、(A) 複数の物理プロセッサにスレッドを同時に実行させるマルチプロセッサ技術 (図1Cの150) や、(B) 単一の物理プロセッサによって認識される複数の仮想プロセッサにスレッドを同時に実行させるマルチスレッド同時処理 (SMT) 技術 (図1Cの109) だけでなく、(C) 複数の物理プロセッサ (コア) 毎に複数の仮想プロセッサを備えたコンピュータシステムにスレッドを同時に実行させるマルチスレッド同時処理 (SMT) 技術 (SMTコアのマルチコア (図1Cの170)) などが例示されている (下図において  $AS_1$  および  $AS_n$  が仮想プロセッサに、103が物理プロセッサに対応)。

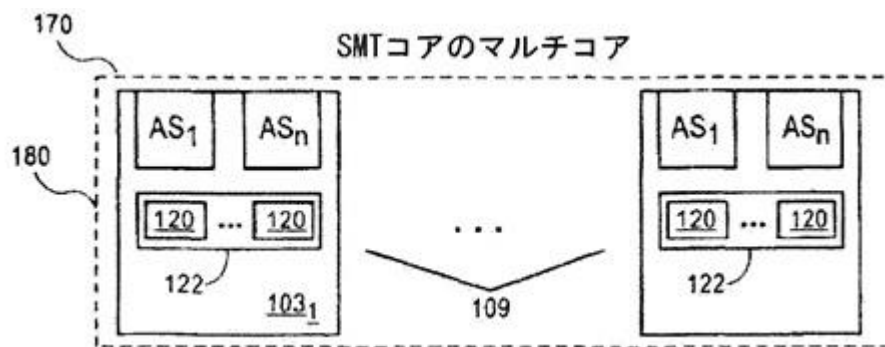
【図1C】



(A) 複数の物理プロセッサにスレッドを同時に実行させるマルチプロセッサシステム例

(B) 単一の物理プロセッサが複数の仮想プロセッサを備えるシステム例

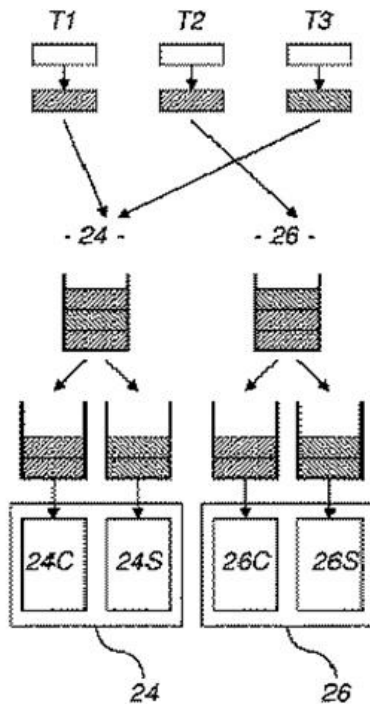




(C) 複数の物理プロセッサ毎に複数の仮想プロセッサを備えたシステム例

甲4には、マルチプロセッサ構成であって、かつ、SMT技術でもある(C)のシステム構成も開示されており、上記(C)のシステム構成も、マルチプロセッサ構成である以上、スレッド処理の効率性が複数の物理プロセッサの物理的な関係に影響されるといえる。

本願の明細書及び図面に記載された実施例においても、上記(C)のマルチプロセッサ構成であって、かつ、SMT技術でもあるシステム構成が実施例として記載されており、本願の特許請求の範囲の請求項6においても、「コンピュータシステム(18)の少なくとも1つの中央処理装置(24, 26)に属する複数の仮想プロセッサ(24C, 24S, 26C, 26S)上で複数の命令スレッドを組み合わせるを実行するのを管理する方法であって」と記載されており、上記(B)及び(C)のシステム構成を包含するものとなっている。



本願の図2（一部抜粋）

複数の物理プロセッサ（24，26）がそれぞれ、複数の仮想プロセッサ（24C，24S，26C，26S）を備えるシステム構成（C）となっている。

原告の主張は、本願の特許請求の範囲の請求項6に包含され、本願の実施例でもある上記（C）のシステム構成を考慮したものとなっていない。

上記（A）のマルチプロセッサ構成システムを、上記（C）の複数の物理プロセッサ（コア）毎に複数の仮想プロセッサを備えたSMTシステム構成にすることには、電力消費量増加につながるクロック数の上昇を抑えながら、並列実行可能なスレッド数を増加させて、性能の向上を計るという動機付けが存在しているといえる。

エ 乙2は、SMT技術の周知技術を示す甲3及び甲4の出願人でもある主要なプロセッサ・メーカーであるインテル・コーポレーションに所属する著者が、マルチスレッディング技術に対応したマルチコアCPU（プロセッサ）の技術開発動向を記載したものであり、物理プロセッサ（コア）の数を（2から4に）増加させることにより性能の向上を計ってきたデスクトップPC用CPU（Core2 DuoからCore2 Quad）が2008年11月発表のCore i7では、物理プロセッサ（コア）の数は増加させず各物理プロセッサ（コア）に2つずつの論理ステートを備えることで2つずつの仮想プロセッサを設ける構成を採用したことが見て取れる。

上記インテルCPUの技術開発動向の経緯に照らしても、前記(イ)の(A)のマルチプロセッサ構成システムに対応するCore2 Quadプロセッサ(最大4スレッドを同時実行可能)を、同(C)の複数の物理プロセッサ(コア)毎に複数の仮想プロセッサを備えたSMTシステム構成に対応し、より多くのスレッド(最大8スレッドを同時実行可能)を並列実行可能な次世代プロセッサであるCore i7に置換することには、同時実行可能スレッド数を増加させるという動機付けが存在しているといえる。

オ 甲1発明は、個別の物理サブシステム内においてスレッドが使用するリソースの局所化を通じて、効率的なリソース割り付けが、よりトランスペアレントな態様で実装され、著しいカスタム化を伴わないことを解決すべき課題としている。

甲1発明が課題を解決するための技術思想は、次のようなものである。

スレッドに関連付けされる特定の「型」を基礎として、それらのスレッドが動的にコンピュータ・リソースに割り当てられるように、スレッド型に、コンピュータ内の同一物理サブシステム内に常駐するリソースが割り付けられ、その結果、それらの特定スレッド型の、新しく作り出されたスレッド及び／又は再アクティブ化されたスレッドが、それぞれのスレッド型に割り付けられたリソースに動的に割り当てられる。

すなわち、甲1発明においては、スレッドに関連付けされる特定の「型」に着目し、特定スレッド型には、同一のリソース(セット)を割り付けること(局所化)により、キャッシュなどのリソースに格納されたデータを特定スレッド型が繰り返し利用する際に、キャッシュに格納済みのデータを再利用可能として、他の物理サブシステムのメモリ等からキャッシュ等へのデータ通信等のコストを削減しているといえる。

マルチスレッド処理においては、プロセッサ単位(物理プロセッサ又は仮想プロセッサ)で割付けが行われ、甲1発明において、物理サブシステム内のプロセッサが物理プロセッサのみから構成されるシステム構成(A)であろうと、仮想プロセ

ッサを含んだシステム構成(C)であろうと、スレッドに関連付けされる特定の「型」に着目し、特定スレッド型には、同一のリソース（セット）を割り付けること（局所化）により、キャッシュなどのリソースに格納されたデータを特定スレッド型が繰り返し利用する際に、キャッシュに格納済みのデータを再利用可能として、他の物理サブシステムのメモリ等からキャッシュ等へのデータ通信等のコストを削減することが可能であるといえる。

## 2 取消事由2について

相違点cは、相違点3と実質的に同じであるため、本願補正発明の相違点3の容易想到性判断で述べた議論は、すべて相違点cにも当てはまる。

したがって、甲1発明に甲3～5に示された周知技術を適用することで、当業者が本願発明を容易に想到し得たものである。

相違点aについての原告の主張は、争う。

## 第5 当裁判所の判断

### 1 本願発明及び本願補正発明について

(1) 本願発明は、前記第2の2(1)、本願補正発明は、同(2)のとおりであるところ、本願発明に係る特許出願の出願時の特許請求の範囲は、次のとおりである(甲6)。

#### 【請求項1】

命令スレッドの実行を管理するコンピュータシステム(18)であって、同じ中央処理装置に属する複数の仮想プロセッサ(24C, 24S, 26C, 26S)上で複数の命令スレッドを組み合わせて実行する少なくとも1つの中央処理装置(24, 26)と、前記仮想プロセッサに命令スレッドの実行を分散する少なくとも1つのマネージャ(38)とを有するコンピュータシステムにおいて、実行する前記命令スレッドを複数の所定のタイプに応じて分類する分類手段(36)を有することと、前記命令スレッドの実行を分散する前記マネージャ(38)を、実行する前記命令スレッドをそれぞれタイプに応じて前記仮想プロセッサ(24C, 24S,

26C, 26S) に方向付けるように設計することを特徴とするコンピュータシステム。

**【請求項2】**

前記同じ中央処理装置(24, 26)の各仮想プロセッサ(24C, 24S, 26C, 26S)を前記所定のタイプの1つに関連付ける手段(C, S)を備える、請求項1に記載のコンピュータシステムにおいて、前記命令スレッドの実行を分散する前記マネージャ(38)を、実行する前記命令スレッドをそれぞれ該スレッドと同じタイプの仮想プロセッサ(24C, 24S, 26C, 26S)に方向付けるように設計するコンピュータシステム。

**【請求項3】**

前記中央処理装置(24, 26)は2つの仮想プロセッサ(24C, 24S, 26C, 26S)を有し、1つは命令スレッドの第1のタイプ(C)に関連付け、もう1つは命令スレッドの第2のタイプ(S)に関連付け、実行するコンピュータプロセスはそれぞれ前記第1のタイプか前記第2のタイプである、請求項1または2に記載のコンピュータシステム。

**【請求項4】**

— 「計算」と呼ぶ命令スレッドの前記第1のタイプ(C)は、結果を供給するように前記コンピュータシステムが実行するようになっているプログラムに対して、該プログラムの実行に参加して結果を直接生成する命令スレッドに関係し、

— 「サービス」と呼ぶプロセスの前記第2のタイプ(S)は、該プログラムの実行に参加して「計算」タイプの前記命令スレッドに付属サービスを供給する命令スレッドに関係する、

請求項3に記載のコンピュータシステム。

**【請求項5】**

少なくとも2つの中央処理装置(24, 26)を有し、それぞれが複数の仮想プロセッサ(24C, 24S, 26C, 26S)を備え、仮想プロセッサ上で複数の

命令スレッドを組み合わせて実行する，請求項 1 から 4 のいずれか一項に記載のコンピュータシステムにおいて，前記命令スレッドの実行を分散する前記マネージャ（38）を，中央処理装置にある仮想プロセッサとは独立して，実行する命令スレッドそれぞれに対して中央処理装置を選択するように設計するコンピュータシステム。

**【請求項 6】**

コンピュータシステム（18）の同じ中央処理装置（24，26）に属する複数の仮想プロセッサ（24C，24S，26C，26S）上で複数の命令スレッドを組み合わせて実行するのを管理する方法であって，前記仮想プロセッサに前記命令スレッドの実行を分散するステップ（104）を含む方法において，複数の所定のタイプに応じて実行する前記命令スレッドを分類する予備ステップ（100）を含むことと，前記命令スレッドの実行を分散するステップ（104）で，実行する命令スレッドそれぞれをタイプに応じて仮想プロセッサに方向付けることとを特徴とする方法。

**【請求項 7】**

前記分類ステップ（100）で，命令スレッドを管理するアプリケーションプログラミングインターフェースの関数でタイプを識別するパラメータに基づいて，前記命令スレッドをそれぞれタイプで分類する，請求項 6 に記載の方法。

**【請求項 8】**

前記分類ステップ（100）で，命令スレッドの実行を含むプログラムの実行コマンドでタイプを識別するパラメータに基づいて，前記命令スレッドをそれぞれこのタイプで分類する，請求項 6 に記載の方法。

**【請求項 9】**

スーパーコンピュータタイプのサーバクラスタ（10）の複数の処理ノード上で複数の命令スレッドを実行することの管理に対する請求項 6 から 8 のいずれか一項に記載の方法の適用であって，各処理ノード（18）が，同じ中央処理装置に属す

る複数の仮想プロセッサ（24C，24S，26C，26S）上で複数の命令スレッドを組み合わせて実行する少なくとも1つの中央処理装置（24，26）を有する適用。

**【請求項10】**

通信ネットワークからダウンロード可能なコンピュータプログラムおよび／またはコンピュータで再生可能な媒体に記録したコンピュータプログラムおよび／またはプロセッサのなかでもとりわけコンピュータのオペレーティングシステム(34)で実行可能なコンピュータプログラムであって、前記プログラムをコンピュータ上で実行した場合に、請求項6から8のいずれか一項に記載の複数の命令スレッドを組み合わせた実行の管理方法のステップを実行するプログラムコードの命令を含むことを特徴とするコンピュータプログラム。

(2) 本願の出願当初の明細書及び図面は、平成26年3月31日の手続補正及び本件補正によって、補正されていない(甲6，9，12)。

本願の出願当初の明細書及び図面(当初明細書等。甲6)には、次の記載がある。

**【技術分野】**

**【0001】**

本発明は、命令スレッドの実行を管理するコンピュータシステムであって、同じ中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する少なくとも1つの中央処理装置と、命令スレッドの実行を仮想プロセッサに分散する少なくとも1つのマネージャとを有するコンピュータシステムに関する。本発明はまた、その方法、サーバクラスタへの本方法の適用、および本方法を実行するための命令を含むコンピュータプログラムにも係る。

**【背景技術】**

**【0002】**

以下の説明文において、「コンピュータシステム」とは、コンピュータプログラムの命令を実行することができ、そのために少なくとも1つの中央処理装置でプロセ

ッサの計算コアまたはCPU(英語の「Central Processing Unit」)とも呼ばれるものを有する少なくとも1つのプロセッサを備えるあらゆるシステムを指す。プロセッサには、単一の計算コアのみ、または複数の計算コアを備えることができることがわかるであろう。本発明によるコンピュータシステムは、複雑性が増す順に、1つまたは複数の計算コアを有するマイクロプロセッサ、1つまたは複数のマイクロプロセッサを有する単純なマイクロコンピュータ、または複数のマイクロコンピュータを、データ転送ネットワークを介して相互接続したさらに複雑なシステムで構成することができる。本発明を実装することができるコンピュータシステムの複雑性は、原則として目的とする適用例によって異なる。

#### 【0003】

「コンピュータプロセス」とは、より一般には「プロセス」と呼ばれ、プロセッサの中央処理装置(すなわち計算コア)によって、場合によってはその他の資源を用いて、特別に割り当てたメモリ空間で実行する命令の集合を指す。コンピュータプログラム自体は命令を構造化した集合で構成されるため、コンピュータプロセスをコンピュータプログラムまたはコンピュータプログラムの一部分を実行するインスタンスと考えることができる。同じ1つのプログラムを(同じ1つのプロセッサまたはさまざまなプロセッサ上で平行かつ連続的に)複数回実行することができるため、このプログラムは複数のコンピュータプロセスを生成することができる。

#### 【0004】

コンピュータプロセスは、必ずしも単一の線形的な命令系列を備えている必要はなく、非同期に複数の系列を要求することができる。そのため、より一般に「スレッド」と呼ばれる「命令スレッド」とは、コンピュータプロセスの実行に参加するこのような線形的な命令系列を指す。その結果、命令実行の観点から、コンピュータプロセスとは常に命令スレッドまたは命令スレッドの集合であると考えてよい。

#### 【0005】

「同じ中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組



み合わせて実行する中央処理装置」とは、命令スレッド実行の並列処理を、中央処理装置の内部資源（とりわけ実行の「パイプライン」、レジスタおよびキャッシュメモリ）のいくつかを共有することによって増進させるメカニズムを備える中央処理装置のことである。このような中央処理装置には、資源を同時に共有することができる命令スレッドと同じだけ仮想プロセッサがある。ユーザから見れば、単一の中央処理装置ではなく、複数、さらに正確にはこの中央処理装置にある仮想プロセッサの数だけ中央処理装置があるかのように見える。

#### 【0006】

このメカニズムは一般に、「ハイパースレッディング」または「同時マルチスレッディング」と呼ばれる。このメカニズムを実装している中央処理装置の目的は、命令スレッドの共有メモリ空間からくるデータを待つことで生じる非アクティブな時間を活用することである。具体的には、命令スレッドがこのような待機状態にあるとき、この命令スレッドを実行する中央処理装置は自動的に別の命令スレッドの実行に進むため、複数の異なる仮想プロセッサがあるかのような印象を与える。

#### 【0007】

オペレーティングシステムLinux（登録商標）は、現在このようなメカニズムをサポートしており、ハイパースレッディングメカニズムを搭載したマイクロプロセッサ上でこのオペレーティングシステムを実行すると、このメカニズムがアクティブのときにはこのマイクロプロセッサに属するN個の仮想プロセッサがあるようにユーザには見える。またこのオペレーティングシステムは、有利なように命令スレッドを実行する際にロードバランシングアルゴリズムも実行するため、この命令スレッドには仮想プロセッサの特性を考慮する必要がある。特に、このオペレーティングシステムは、同じ中央処理装置の別の仮想プロセッサに命令スレッドを切り替えて仮想プロセッサをアンロードすることはない。この制約により、ロードバランシングアルゴリズムが非常に重くなる。

#### 【0008】

具体的には、この制約は、例えば携帯電話のアプリケーションに内蔵するマイクロプロセッサなどに用いるリアルタイム計算およびオンボード処理のアプリケーションでとりわけ問題が起こる。

#### 【0009】

具体的な第2の例によれば、サーバクラスタに組織した複数の処理ノードを有するHPC（英語の「High Performance Computing」）タイプのスーパーコンピュータの環境では、ユーザは中央処理装置上でのアプリケーションの配置を細かくコントロールすることを望み、例えばそれぞれの通信メカニズムに有利に働く共有キャッシュを利用することもある。

#### 【0010】

これらの命令スレッドを管理するアプリケーションプログラミングインターフェースのシステムを呼び出すことによって、命令スレッドを細かく配置できる可能性があり、それによってユーザ（すなわちプログラマ）は、命令スレッドに対するオペレーティングシステムタスクのスケジューラの挙動を明示することができるが、これには関連するアプリケーションに対して、コンピュータシステムのトポロジーの知識およびソフトウェア層との競合が生じるおそれのある配置作業が必要になる。

#### 【0011】

仮想プロセッサの特性を考慮することは、実際には複雑である。それによって実際に問題が起こり、HPCスーパーコンピュータのアプリケーションで自動的な管理ができなくなることさえある。

#### 【発明の概要】

#### 【発明が解決しようとする課題】

#### 【0012】

そのため、前述の問題および制約を少なくとも部分的に緩和することができる、命令スレッドを組み合わせた実行の管理システムを備えることが望まれる。

#### 【課題を解決するための手段】

### 【0013】

よって本発明は、命令スレッドの実行を管理するコンピュータシステムであって、同じ中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する少なくとも1つの中央処理装置と、命令スレッドの実行を仮想プロセッサに分散する少なくとも1つのマネージャとを有するコンピュータシステムにおいて、実行する命令スレッドを複数の所定のタイプに応じて分類する分類手段を有することと、命令スレッドの実行を分散するマネージャを、実行する命令スレッドをそれぞれタイプに応じて仮想プロセッサに方向付けるように設計することを特徴とするコンピュータシステムを目的とする。

### 【0014】

したがって、命令スレッドをタイプごとに分類し、これをタイプに応じて命令スレッドの実行を分散するマネージャを用いて体系的にそれぞれの仮想プロセッサに方向付けることによって、計算資源を保護する。その結果、中央処理装置を最適な形で使用することになり、関連するアプリケーションの実行が加速する。さらに、ハイパースレッディングメカニズムによりユーザ（プログラマ）が原因となることもあるエラーのリスクが抑えられる。

### 【0015】

選択的に、本発明によるコンピュータシステムはさらに、この同じ中央処理装置の各仮想プロセッサをこれらの所定のタイプの1つに関連付ける手段を備えることができ、命令スレッドの実行を分散するマネージャを、実行する命令スレッドをそれぞれこのスレッドと同じタイプの仮想プロセッサに方向付けるように設計することができる。

### 【0016】

したがって、命令スレッドを分散するマネージャは、場合によっては命令スレッドに対して中央処理装置を選択するのに関わるだけでよく、仮想プロセッサの正確な選択は命令スレッドのタイプによって自動的に決まる。その結果、ハイパースレ

ッディングメカニズムはユーザから隠すことができる。よって、ユーザがこのメカニズムを理解していないためにエラーが生じるリスクは一切なくなる。さらに、命令スレッドは自動的に正しい仮想プロセッサに方向付けられるため、パフォーマンスは一層改善される。

**【0017】**

同じく選択的に、中央処理装置は2つの仮想プロセッサを有し、1つは命令スレッドの第1のタイプに関連付け、もう1つは命令スレッドの第2のタイプに関連付け、実行するコンピュータプロセスはそれぞれ第1のタイプか第2のタイプとする。

**【0018】**

同じく選択的に、

— 「計算」と呼ぶ命令スレッドの第1のタイプは、結果を供給するようにコンピュータシステムが実行するようになっているプログラムに対して、このプログラムの実行に参加して結果を直接生成する命令スレッドに関係し、

— 「サービス」と呼ぶプロセスの第2のタイプは、このプログラムの実行に参加して「計算」タイプの命令スレッドに付属サービスを供給する命令スレッドに関係する。

**【0019】**

同じく選択的に、本発明によるコンピュータシステムは、少なくとも2つの中央処理装置を備えることができ、それぞれがこの同じ中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行し、命令スレッドの実行を分散するマネージャを、中央処理装置が備える仮想プロセッサとは独立して、実行する命令スレッドそれぞれに対して中央処理装置を選択するように設計することができる。

**【0020】**

本発明はまた、コンピュータシステムの同じ中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行するのを管理する方法であっ

て、仮想プロセッサに命令スレッドの実行を分散するステップを含む方法において、複数の所定のタイプに応じて実行する命令スレッドを分類する予備ステップを含むことと、命令スレッドの実行を分散するステップで、実行する命令スレッドそれぞれをタイプに応じて仮想プロセッサに方向付けることとを特徴とする方法も目的とする。

**【0021】**

選択的に、分類ステップで、命令スレッドを管理するアプリケーションプログラミングインターフェースの関数でタイプを識別するパラメータに基づいて、命令スレッドをそれぞれタイプで分類する。

**【0022】**

同じく選択的に、分類ステップで、命令スレッドの実行を含むプログラムの実行コマンドでタイプを識別するパラメータに基づいて、この命令スレッドをそれぞれこのタイプで分類する。

**【0023】**

本発明はまた、スーパーコンピュータタイプのサーバクラスタの複数の処理ノード上で複数の命令スレッドを実行することの管理に対する上記のように定義した方法の適用であって、各処理ノードが、同じ中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する少なくとも1つの中央処理装置を有する適用も目的とする。

**【0024】**

最後に、本発明はまた、通信ネットワークからダウンロード可能なコンピュータプログラムおよび／またはコンピュータで再生可能な媒体に記録したコンピュータプログラムおよび／またはプロセッサのなかでもとりわけコンピュータのオペレーティングシステムで実行可能なコンピュータプログラムであって、前記プログラムをコンピュータ上で実行した場合に、上記のように定義した複数の命令スレッドを組み合わせた実行の管理方法のステップを実行するプログラムコードの命令を含む

ことを特徴とするコンピュータプログラムも目的とする。

**【0025】**

本発明は、添付の図を参照しながら例のみを目的として挙げた以下の説明文を読めばよりよく理解できるであろう。」

**「【発明を実施するための形態】**

**【0027】**

本発明はどのようなアプリケーションからも独立して、同じ中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する少なくとも1つの中央処理装置と、仮想プロセッサに命令スレッドの実行を分散する少なくとも1つのマネージャとを備えるあらゆるコンピュータシステムに実装することができ、以下に一実施形態を詳細に説明する。

**【0028】**

最も単純な事例では、本発明は、ハイパースレッディングメカニズムを搭載したマイクロプロセッサを備えれば実装することができる。リアルタイム計算および／またはオンボードのアプリケーションが本発明から利益を引き出すことができる。

**【0029】**

ネットワーク接続したコンピューティングデバイスまたはHPCスーパーコンピュータの最も複雑なアプリケーションでは、少なくとも1つのマイクロプロセッサを有する各装置が本発明を実装できることが有利である。

**【0030】**

図1に示す実施形態はHPCコンピュータの適用例であり、以下に本発明の一実施形態を詳細に説明する。実際にこれは本発明の特に重要な産業上の適用例である。ただし、前述したとおり、本発明はこのようなHPCコンピュータの設備に限定されるものではないことは明らかである。

**【0031】**

図1に示すコンピューティングデバイス12、14、16、18の設備10は、

例えば帯域がきわめて広いデータ伝送ネットワーク 20 を少なくとも 1 つ介して相互接続したサーバクラスターで組織した H P C コンピュータを非常に簡略化した不完全な例である。この設備 10 は、例えばとりわけ計算ノードを形成する符号 12, 14, 16 および 18 で示したコンピューティングデバイスと、サービスノード、管理ノードまたは格納ノードを形成するコンピューティングデバイス（図示せず）とを有する。計算ノード 12, 14, 16 および 18 の全体構造は同じであってもよいが、コンピューティングデバイス 18 の構造のみを以下に詳述する。

#### 【0032】

例えばマイクロコンピュータタイプの計算サーバであるコンピューティングデバイス 18 は、

— 第 1 の中央処理装置 24 および第 2 の中央処理装置 26 が接続する通信バスであって、コンピュータプログラム、さらに詳細には命令スレッドにより生じるコンピュータプロセスを実行するための通信バス 22 と、

— ROM 28（英語の「R e a d O n l y M e m o r y」）タイプの読み出し専用メモリ、および処理データおよび／または起動プログラム（例えば B I O S タイプのアプリケーション）を格納し、周辺機器を管理するための R A M 30（英語の「R a n d o m A c c e s s M e m o r y」）タイプの書き込み可能なメモリと、

— 少なくとも 1 つのハードディスク 32 とを有する。

#### 【0033】

このコンピューティングデバイス 18 は、場合によってはさらに（この図には示していないが）従来のように、以下の素子：スクリーン、キーボード、マウス、ネットワーク 20 との通信インターフェース、および C D - R O M または D V D のリーダー／レコーダなどの交換可能な媒体へのデータのリーダー／レコーダを 1 つまたは複数有する。

#### 【0034】

ハードディスク32は、コンピューティングデバイス18の起動時にBIOSアプリケーションをロードしたLinuxシステムなどのオペレーティングシステム34をファイルの形態で格納する。同じ中央処理装置の複数の仮想プロセッサ上で命令スレッドを組み合わせて実行するメカニズムが、このオペレーティングシステムに対してアクティブになる。例えばこのメカニズムは、中央処理装置24および26にそれぞれ2つの仮想プロセッサがあるようなものである。つまり、仮想プロセッサ24Cおよび24Sは第1の中央処理装置24に、仮想プロセッサ26Cおよび26Sは第2の中央処理装置26に対するものである。」

#### 「【0037】

より一般に、前述の構造は、コンピューティングデバイスのあらゆるタイプのネットワークに適しているためHPCタイプである必要はなく、この構造では各コンピューティングデバイスのオペレーティングシステムは、仮想プロセッサ上で命令スレッドを組み合わせて実行するメカニズムを利用する。

#### 【0038】

つまり、先ほど構造的に説明したようなコンピューティングデバイス18は、本発明を実装することができる装置の非限定的な一例にすぎないことがわかるであろう。コンピューティングデバイスの構想には多種多様なアーキテクチャがあり、同じ中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する中央処理装置を最低1つ備える装置はすべて適切である。

#### 【0039】

本発明の一実施形態によれば、コンピューティングデバイス18は、複数の所定のタイプに応じて、また特に、

— 「計算」と呼ぶ命令スレッドの第1のタイプは、結果を供給するようにコンピューティングデバイス18が実行するようになっているプログラムに対して、このプログラムの実行に参加して結果を直接生成する命令スレッドに関係し、



ー 「サービス」と呼ぶプロセスの第2のタイプは、このプログラムの実行に参加して「計算」タイプの命令スレッドに付属サービスを供給する命令スレッドに係る

という2つのタイプの命令スレッドに応じて、実行する命令スレッドを分類する分類手段36を有する。

#### 【0040】

サービスの命令スレッドは、例えば結果の保存、メモリの管理、入出力の管理、データの通信、モニタリング、ファイルへのアクセスなどに係る。

#### 【0041】

アプリケーションに応じて、分類手段36はさまざまな形態を取ることができる。

#### 【0042】

第1の適用例では、実行するプログラムが複数の命令スレッドの実行を要求し、そのうちのいくつかの命令スレッドが計算に参加し、残りの命令スレッドが付属サービスに参加する。この場合、このプログラムは、命令スレッドを管理するアプリケーションプログラミングインターフェースの関数である、例えばアプリケーションプログラミングインターフェースPOSIXの関数 `sched_setscheduler()` を、計算タイプ (HTCALCUL) かサービスタイプ (HTSERVICE) かを識別するパラメータに基づいて呼び出すように設計する。したがって、オペレーティングシステム34によって実装する命令スレッドの分類手段36は、このパラメータを自動的に認識して仮想プロセッサに方向付ける命令スレッドのタイプを識別することからなる。

#### 【0043】

第2の適用例では、複数のプログラムを実行する必要がある、そのうちのいくつかは計算プログラム、残りはサービスプログラムである。この場合、ユーザは、計算かサービスかのタイプをパラメータで識別するコマンドを用いてプログラムの実行を開始するようにする。例えばこれは、コマンド `ht_sched-[CALC`

ULまたはSERVICE] <プログラム名>である。このようにオペレーティングシステム34で実装した命令スレッドの分類手段36は、このパラメータを自動的に認識してプログラムが要求する命令スレッドのタイプを識別することからなる。

#### 【0044】

第3の適用例では、並列計算プログラムに広く使用されるMPI（英語の「Message Passing Interface」）ライブラリが、計算モードで開始される主要プログラムの枠組みでのサービスに向けた固有の命令スレッドを生成する（第2の適用例を参照）。この場合、このライブラリは、サービスタイプ（HT\_SERVICE）を識別するパラメータに基づいて関数setscheduler（）を呼び出すことができる。したがって、オペレーティングシステム34によって実装する命令スレッドの分類手段36は、このパラメータを自動的に認識してこのMPIライブラリが生成する命令スレッドのタイプを識別することからなる。

#### 【0045】

命令スレッドのタイプが前述したように先験的に明示されない第4の適用例では、オペレーティングシステム34の分類手段36を、自動的に、その上簡潔に、とりわけこれらの命令スレッドの起動元であるアプリケーションを自動的に認識することによってタイプを決定するように設計することができる。

#### 【0046】

その上、コンピューティングデバイス18は、同じ中央処理装置の各仮想プロセッサを前述した2つのタイプの一方に関連付ける手段を有する。図1に示すように、この関連付け手段は、例えば計算タイプかサービスタイプかをそれぞれ識別するパラメータCおよびSである。この関連付け手段は、例えば該当する中央処理装置24、26、または変形例ではオペレーティングシステム34によって明示的または暗黙的に管理される。したがって、図1に概略的に示した例では、第1の中央処理装置24の仮想プロセッサ24Cは計算タイプであり、第1の中央処理装置24の仮想プロセッサ24Sはサービスタイプであり、第2の中央処理装置26の仮想プ

ロセッサ 2 6 C は計算タイプであり，第 2 の中央処理装置 2 6 の仮想プロセッサ 2 6 S はサービスタイプである。

**【0047】**

オペレーティングシステム 3 4 は，仮想プロセッサ 2 4 C，2 4 S，2 6 C および 2 6 S に命令スレッドの実行を分散するマネージャ 3 8 を有する。さらに正確には，この分散マネージャ 3 8 は，実行する命令スレッドをそれぞれタイプに応じて 1 つの仮想プロセッサに方向付けし，さらに正確にはタイプが同じ仮想プロセッサにも方向付けするように設計する。したがって，仮想プロセッサ 2 4 C または 2 6 C には計算タイプの命令スレッドしか方向付けできず，仮想プロセッサ 2 4 S または 2 6 S にはサービスタイプの命令スレッドしか方向付けできない。

**【0048】**

具体的には，分散マネージャ 3 8 は，

- － 実行する命令スレッドそれぞれに対して，中央処理装置にある仮想プロセッサ 2 4 C，2 4 S または 2 6 C，2 6 S とは独立して中央処理装置 2 4 または 2 6 を選択する第 1 のソフトウェア手段 4 0 と，
  - － 選択した中央処理装置の仮想プロセッサのうち，実行する命令スレッドと同じタイプのものを自動的に割り当てる第 2 のソフトウェア手段 4 2 と
- を有する。

**【0049】**

次に，図 1 のコンピューティングデバイス 1 8 が実装する複数のコンピュータプロセスを組み合わせた実行の管理方法を図 2 を参照して詳述する。

**【0050】**

第 1 の分類ステップ 1 0 0 では，実行待機状態にある命令スレッド T 1，T 2 および T 3 を計算タイプ（右斜線）かサービスタイプ（左斜線）であるかによって分類する。このタイプは，コマンドのパラメータまたはアプリケーションプログラミングインターフェースの関数のパラメータで先験的に定義し，前述したようにオペ

レーティングシステム 3 4 のソフトウェア手段 3 6 によって自動的に認識することができる。このタイプは、オペレーティングシステム 3 4 のソフトウェア手段 3 6 によって直接、自動的に定義することもできる。

**【 0 0 5 1 】**

次のステップ 1 0 2 では、オペレーティングシステム 3 4 の分散マネージャ 3 8 は、ソフトウェア手段 4 0 を用いて各命令スレッドに対して中央処理装置 2 4 または 2 6 を選択し、選択した中央処理装置の待ち行列にこの命令スレッドを配置する。

**【 0 0 5 2 】**

その後、各命令スレッドをソフトウェア手段 4 2 が実行する仮想プロセッサに割り当てるステップ 1 0 4 では、中央処理装置 2 4 または 2 6 の待ち行列に配置した各命令スレッドが、実際にはこの命令スレッドと同じタイプのこの中央処理装置 2 4 または 2 6 の仮想プロセッサの待ち行列に自動的に配置される。

**【 0 0 5 3 】**

最後に、実行ステップ 1 0 6 では、前のステップで割り当てられた仮想プロセッサによって各命令スレッドを実行する。

**【 0 0 5 4 】**

前述したようなコンピューティングデバイス 1 8 によって、複数の仮想プロセッサ上での命令スレッドの実行管理が容易になることは明らかである。

**【 0 0 5 5 】**

先行技術では、仮想プロセッサ上で命令スレッドを組み合わせて実行するメカニズムを実装することができるコンピューティングデバイスにロードしたオペレーティングシステムは、一般の計算媒体と同じようにあらゆる仮想プロセッサを備えているため、命令スレッドの分散が一層複雑になっている。

**【 0 0 5 6 】**

本発明によれば、オペレーティングシステムによる命令スレッドの分散が簡略化し、とりわけ H P C スーパーコンピュータまたはリアルタイム計算のアプリケーション

ョンを実装するプロセッサなど計算が強く求められる環境で、命令スレッドを組み合わせるメカニズムから多くの利益が引き出される。さらに正確には、前述した実施形態では、同じ1つの中央処理装置にある2つの仮想プロセッサを、相互に交換可能な2つの計算媒体ではなく、計算に対する1つの主要な実行媒体および付属サービスに対する1つの二次的な実行媒体として示した。したがって、N個の中央処理装置がある一般化したコンピューティングデバイス18では、命令スレッドを組み合わせるメカニズムを実装したとしても、実際にはユーザ（およびオペレーティングシステム）にはN個の中央処理装置しか見えず、このメカニズムを実装していないかのように見える。その結果、命令スレッドの分散で起こる管理上の制約を受けることなくこのメカニズムから利益が引き出される。

#### 【0057】

選択した中央処理装置の仮想プロセッサに命令スレッドを自動的に割り当てることによって、命令スレッドはその役割および制約に応じて仮想プロセッサに分配され、ユーザまたはオペレーティングシステムがこの分配に介入することはない。

#### 【0058】

命令スレッドが分配されると、同じ1つの中央処理装置にある2つの仮想プロセッサ上で実行メカニズムを平行に実装することの利点が、以下の点にあることがわかる。

— 中央処理装置は2つの異なる待ち行列を管理するため、1つの命令スレッドから別の命令スレッドへの移行を管理する必要がなくなり、この同時の実行はハードウェアが管理する。

— 特に、仮想プロセッサが実行する命令スレッドが1つしか残っていないとき、中央処理装置は仮想プロセッサに割り込むことなく仮想プロセッサにコードを実行させることができる。

— 1つのプログラムから別のプログラムへの移行は、命令ごとに行われる。

#### 【0059】

この非常に細かい粒度は、とりわけ支障を来すおそれのあるデーモンの場合に明確に有益になり得る。実際、同じ1つの中央処理装置にある計算の仮想プロセッサは減速するが、一定時間の間サービスの仮想プロセッサからブロックされることはない。支障（デーモンおよび割り込み）による影響は、計算の仮想プロセッサによって著しく減少する。

#### 【0060】

HPC環境では、計算コードを計算の仮想プロセッサに配置し、付属サービスをサービスの仮想プロセッサに配置することができ、これはバッチマネージャなどの使用ツールに対して明白な方法で行う。リアルタイム環境では、これによって重要な命令スレッドを他のすべてのサービス命令スレッドから自動的に隔離し、スケジューリングのために起こる割り込みは一切なく、選択した中央処理装置で利用可能な実行時間の最低50%をこの重要な命令スレッドのために確保することができる。

#### 【0061】

最後に、コンピューティングデバイス18と、各命令スレッドに1つのタイプを先験的に割り当てるこの機能性を考慮していないプログラムとの互換性に関しては、デフォルトとして命令スレッドを1つのサービスの仮想プロセッサに方向付ける必要があることは明らかである。実際、オペレーティングシステムが開始するデーモンはすべて、計算に支障を来さないようにサービスの仮想プロセッサで実行する必要がある。そのため、命令スレッドを計算の仮想プロセッサに方向付けるために計算コードのみを特別に適合させる必要がある。この適合は、必ずしもプログラム自体から行う必要はなく、例えば前述のMPIライブラリなど外部のツールで実行してもよい。

#### 【0062】

さらに、本発明は前述した実施形態に限定されるものではないことがわかるであろう。とりわけ、各中央処理装置が3つ以上の仮想プロセッサを備えられるメカニズムを実装することができる場合、3つ以上のタイプの命令スレッドを定義するこ

とが有利になり得る。

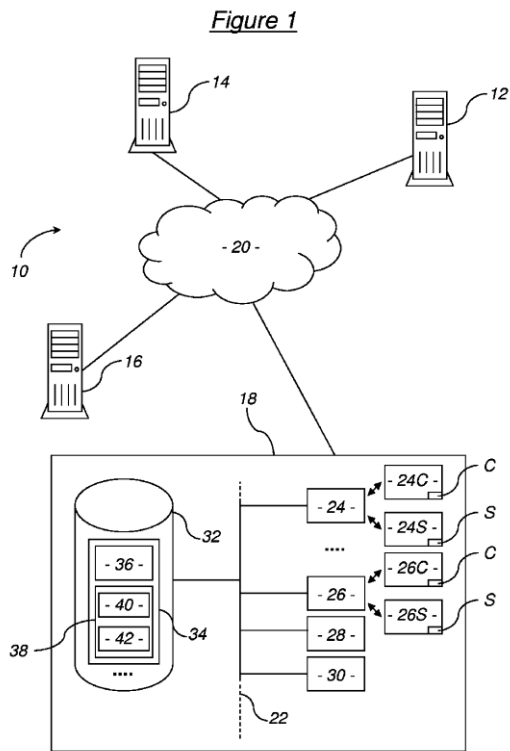
**【0063】**

また、何度も前述して示唆したように、図1に示すHPCスーパーコンピュータの設備が本発明を実装できる唯一のものではない。例えばリアルタイム計算および／またはオンボード処理などこれよりも遙かに簡易なコンフィギュレーションでは、複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する少なくとも1つの中央処理装置を有する単一のプロセッサ（またはこのようなプロセッサを備えるコンピュータ）を備えていれば、本発明を実装することができる。

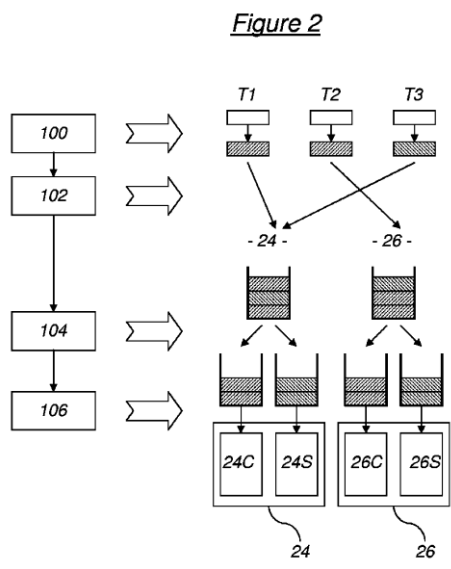
**【0064】**

さらに一般的には、ここに開示した教示に照らし合わせて、上に記載した実施形態にさまざまな修正を加えてもよいことは当業者には明らかであろう。以下の特許請求の範囲では、使用した用語が請求項を本明細書で明らかにした実施形態に限定するものと解釈してはならず、本文の記載およびそこから予見される内容は、ここに開示した教示の実装に一般知識を応用することによって当業者が到達しうる範囲内であることから、請求項が範囲に含めると想定するあらゆる同等のものも含まれると解釈すべきである。

【図1】



【図2】



(3) 前記第2の2の事実及び前記(1)及び(2)の事実によると、本願の出願当初の



発明（以下「当初発明」という。）、本願発明及び本願補正発明（以下、併せて「本発明」という。）について、次のとおり認められる。

#### ア 技術分野

本発明は、命令スレッドの実行を管理するコンピュータシステムであって、同じ中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する少なくとも一つの中央処理装置と、命令スレッドの実行を仮想プロセッサに分散する少なくとも一つのマネージャとを有するコンピュータシステムに関する。本発明は、また、その方法、サーバクラスタへの本方法の適用及び本方法を実行するための命令を含むコンピュータプログラムにも係る。【0001】

ここで、「同じ中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせて実行する中央処理装置」とは、命令スレッド実行の並列処理を、中央処理装置の内部資源（とりわけ実行の「パイプライン」、レジスタ及びキャッシュメモリ）のいくつかを共有することによって増進させるメカニズムを備える中央処理装置のことである。【0005】

#### イ 課題

ハイパースレッディングメカニズムを搭載したマイクロプロセッサ上で、従来のオペレーティングシステム（Linux（登録商標））を実行すると、命令スレッドを実行する際にロードバランシングアルゴリズムも実行するため、命令スレッドは仮想プロセッサの特性を考慮する必要があった。特に、このオペレーティングシステムは、同じ中央処理装置の別の仮想プロセッサに命令スレッドを切り替えて仮想プロセッサをアンロードすることはないという制約があるため、ロードバランシングアルゴリズムが非常に重くなっていた【0007】。この制約は、例えば携帯電話のアプリケーションに内蔵するマイクロプロセッサなどに用いるリアルタイム計算およびオンボード処理のアプリケーションでとりわけ問題が生じていた【0008】。

別の例によると、HPC（High Performance Computi

n g) タイプのスーパーコンピュータの環境では、ユーザが中央処理装置へのアプリケーションの配置を細かくコントロールすることを望む (【0009】)。ユーザ (すなわちプログラマ) は、命令スレッドを管理するアプリケーションプログラミングインターフェースのシステムを呼び出すことによって、命令スレッドを細かく配置し、それによって、命令スレッドに対するスケジューラの挙動を明示することができるが、このためには、コンピュータシステムのトポロジーの知識が必要であり、他のソフトウェア層と競合するおそれのある配置が不可避であった (【0010】)。

仮想プロセッサの特性を考慮することは、複雑であり、HPCスーパーコンピュータのアプリケーションにおいて自動的な管理ができなくなることさえあった。

(【0011】)

そのため、前述の問題および制約を少なくとも部分的に緩和することができる、命令スレッドを組み合わせた実行の管理システムが望まれていた。(【0012】)

#### ウ 課題解決手段

本発明は、前記技術分野において、実行する命令スレッドを複数の所定のタイプに応じて分類する分類手段又は分類ステップを有し、実行する命令スレッドをそれぞれタイプに応じて仮想プロセッサに方向付けるように命令スレッドの実行を分散するマネージャが設計されているものである。(【0013】、【0020】～【0022】)

選択的に、本発明は、実行する命令スレッドを同じタイプの仮想プロセッサに方向付けるようにすることができる。(【0015】)

選択的に、本発明は、同じ中央処理装置の各仮想プロセッサをこれらの所定のタイプの一つに関連付ける手段を備えることができ、中央処理装置は二つの仮想プロセッサを有し、実行するコンピュータプロセスは第1のタイプか第2のタイプとし、仮想プロセッサの一つは命令スレッドの第1のタイプに関連付け、仮想プロセッサのもう一つは命令スレッドの第2のタイプに関連付けることができる。(【0017】)

選択的に、前記第1のタイプ及び第2のタイプは、それぞれ、「計算」と呼ばれるタイプ及び「サービス」と呼ばれるタイプとすることができ、「計算」タイプの命令スレッドは、実行結果を提供するプログラムの実行に参加して結果を直接生成する命令スレッドに関係し、「サービス」タイプの命令スレッドは、このプログラムの実行に参加して「計算」タイプの命令スレッドに付属サービスを提供する命令スレッドに関係する。【0018】

選択的に、本発明によるコンピュータシステムは、少なくとも二つの中央処理装置を備えることができ、マネージャを、中央処理装置が備える仮想プロセッサとは独立して、実行する命令スレッドそれぞれに対して中央処理装置を選択するように設計することができる。【0019】

選択的に、分類ステップでは、命令スレッドを管理するアプリケーションプログラミングインターフェースの関数で用いられるタイプを識別するためのパラメータに基づいて、命令スレッドをそれぞれのタイプに分類してもよいし【0021】、【0042】、あるいは、プログラムの実行コマンドで用いられるタイプを識別するためのパラメータに基づいて、このプログラムによって実行される命令スレッドをそれぞれのタイプで分類してもよい。【0022】、【0043】

#### エ 効果

本発明は、命令スレッドをタイプごとに分類し、これをタイプに応じてそれぞれの仮想プロセッサに方向付けることによって、計算資源が保護できる。その結果、中央処理装置を最適な形で使用することになり、関連するアプリケーションの実行速度が加速する。さらに、ユーザ（プログラマ）がハイパースレッディングメカニズムを理解していないことに起因するエラーのリスクが抑えられる。【0014】、【0016】

本発明によると、一つの中央処理装置にある二つの仮想プロセッサ上で実行メカニズムを平行に実装することにより、以下の利点を得られる。

① 中央処理装置は二つの異なる待ち行列を管理するため、一つの命令スレッドか

ら別の命令スレッドへの移行を管理する必要がなくなる。

② 特に、実行すべき命令スレッドが仮想プロセッサ毎に一つずつの場合には、割り込みをせずに仮想プロセッサがコードを実行できる。

③ 一つのプログラムから別のプログラムへの移行は、命令ごとに行われる。

同じ一つの中央処理装置にある計算の仮想プロセッサは減速するが、一定時間の間サービスの仮想プロセッサからブロックされることはない。支障（デーモン及び割り込み）による影響は、計算の仮想プロセッサにとって著しく減少する。（【0059】）

計算コードを計算の仮想プロセッサに配置し、付属サービスをサービスの仮想プロセッサに配置するので、リアルタイム環境では、重要な命令スレッドを他のすべてのサービス命令スレッドから自動的に隔離し、スケジューリングのために起こる割り込みは一切なく、選択した中央処理装置で利用可能な実行時間の最低50%をこの重要な命令スレッドのために確保することができる。（【0060】）

#### オ 実施例等

本発明の実施形態は、同一の中央処理装置に属する複数の仮想プロセッサ上で複数の命令スレッドを組み合わせる中央処理装置を2個（第1の中央処理装置24及び第2の中央処理装置26）と、仮想プロセッサに命令スレッドの実行を分散するマネージャ38とを備えるコンピューティングデバイス18であって、第1の中央処理装置24及び第2の中央処理装置26を接続する通信バスと、ROM28と、RAM30と、ハードディスク32とを有する。（【0027】、【0032】、【0047】）

本発明の実施形態には、「計算」タイプ及び「サービス」タイプの二つのプロセスのタイプがあり、「計算」と呼ぶ命令スレッドの第1のタイプは、結果を供給するようにコンピューティングデバイス18が実行するようになっているプログラムに対して、このプログラムの実行に参加して結果を直接生成する命令スレッドに関係し、「サービス」と呼ぶプロセスの第2のタイプは、このプログラムの実行に参加して

「計算」タイプの命令スレッドに付属サービスを供給する命令スレッドに関係するものであり、本発明の実施形態は、これら二つのタイプの命令スレッドに応じて、実行する命令スレッドを分類する分類手段 36 を有する。【0039】

さらに、コンピューティングデバイス 18 は、同じ中央処理装置の各仮想プロセッサを前述した二つのタイプの一方に関連付ける手段を有する。この関連付け手段は、例えば計算タイプかサービスタイプかをそれぞれ識別するパラメータ C 及び S である。一例では、第 1 の中央処理装置 24 の仮想プロセッサ 24 C は計算タイプであり、第 1 の中央処理装置 24 の仮想プロセッサ 24 S はサービスタイプであり、第 2 の中央処理装置 26 の仮想プロセッサ 26 C は計算タイプであり、第 2 の中央処理装置 26 の仮想プロセッサ 26 S はサービスタイプである。【0046】

分散マネージャ 38 は、実行する命令スレッドとタイプが同じ仮想プロセッサにも方向付ける。したがって、仮想プロセッサ 24 C 又は 26 C には計算タイプの命令スレッドしか方向付けされず、仮想プロセッサ 24 S 又は 26 S にはサービスタイプの命令スレッドしか方向付けされない。【0047】

アプリケーションに応じて、分類手段 36 はさまざまな形態を取ることができる。【0041】

第 1 の適用例では、一つのプログラムのうちいくつかの命令スレッドが計算に参加し、残りの命令スレッドが付属サービスに参加するものであり、このようなプログラムは、アプリケーションプログラミングインターフェースの関数 `set_scheduler()` を、計算タイプ (HTCALCUL) かサービスタイプ (HTSERVICE) かを識別するためのパラメータに基づいて呼び出すように作られる。分類手段 36 は、前記パラメータを自動的に認識して命令スレッドのタイプを識別する。【0042】

第 2 の適用例では、複数のプログラムが実行され、そのうちのいくつかは計算プログラムであり、残りはサービスプログラムである。この場合、プログラムの実行を開始するには、ユーザは、計算かサービスかのタイプをパラメータで識別するコ

マンドを用いる。このコマンドは、例えば、コマンド `ht__sched` [CALCULまたはSERVICE] <プログラム名>である。分類手段36は、前記パラメータを自動的に認識して命令スレッドのタイプを識別する。【0043】

第3の適用例では、MPI (Message Passing Interface) ライブラリによって、計算タイプで実行された主プログラムから、サービスタイプの命令スレッドが形成される。この場合、MPI ライブラリは、サービスタイプ (HT\_\_SERVICE) を識別するパラメータに基づいて関数 `set_scheduler ()` を呼び出す。命令スレッドの分類手段36は、このパラメータを自動的に認識して命令スレッドのタイプを識別する。【0044】

第4の適用例では、前述の適用例とは異なり、命令スレッドのタイプが事前には明示されない。分類手段36は、命令スレッドの起動元であるアプリケーションを自動的に認識することによってタイプを決定する。【0045】

## 2 取消事由1 (補正の適法性についての判断の誤り) について

### (1) 新規事項追加禁止要件違反について

ア 前記1(1)のとおり、本件補正前の請求項6の命令スレッドのタイプは、いずれも、単に「タイプ」と記載されており、「(S, C)」の記号を伴わないものであったところ、前記第2の1のとおり、原告は、平成26年8月26日付けで、甲1発明等を引用例とする進歩性の欠如を理由とする拒絶査定を受けた(甲7, 10)後、本件補正をした。

イ 本件補正後の請求項6の、「(S, C)」は、それ自体のみからその技術的な意義を読み取れず、また、本件補正後の請求項6の記載中に、その技術的な意義を明確にする定義等の記載は見当たらない。

本件補正後の請求項6の従属項である同請求項7～10にも、本件補正後の請求項6の「(S, C)」の技術的な意義を明確にする記載はない。

ウ(ア) 当初明細書等には、命令スレッドのタイプについては、二つ以上あること【0013】、【0014】、【0017】、【0020】、【0039】、【004

2】、【0046】）、三つ以上ある場合もあること（【0062】）が記載されており、命令スレッドのタイプが第1のタイプと第2のタイプである場合において、命令スレッドの第1のタイプが計算タイプであり、第2のタイプがサービスタイプである場合があることが記載されている（【0039】、【0042】、【0043】、【0044】、【0046】、【0047】、【0050】）。

また、当初明細書等には、命令スレッドのタイプは、①アプリケーションプログラミングインターフェースの関数でタイプを識別するパラメータ（計算タイプ：HTCALCUL、サービスタイプ：HT\_SERVICE）に基づく場合（【0021】、【0042】、【0050】）、②プログラムの実行コマンドでタイプを識別するパラメータ（CALCUL、SERVICE）に基づく場合（【0022】、【0043】、【0050】）、③命令スレッドの起動元のアプリケーションを自動的に認識することによって決定される場合（【0045】）があることが記載されている。

さらに、命令スレッドと仮想プロセッサの関係については、仮想プロセッサのタイプは、パラメータC及びパラメータSによって識別され、パラメータCは計算タイプに、パラメータSはサービスタイプに関連付けられ、仮想プロセッサ24C及び26Cは計算タイプであり、仮想プロセッサ24S及び26Sはサービスタイプである場合（【0046】～【0048】）があることが記載されている。

(イ) 以上によると、当初明細書等においては、「S」及び「C」は、仮想プロセッサのタイプとして記載されていること、命令スレッドのタイプとしては、「計算タイプ」及び「サービスタイプ」という文言が用いられていることが認められる。

しかし、前記(ア)のとおり、仮想プロセッサのタイプは、パラメータC及びパラメータSによって識別され、パラメータCは計算タイプに、パラメータSはサービスタイプに関連付けられ、仮想プロセッサの24C及び26Cは「計算タイプ」、同24S及び26Sは「サービスタイプ」とされている。また、命令スレッドのタイプがアプリケーションプログラミングインターフェースの関数でタイプを識別するパラメータ（計算タイプ：HTCALCUL、サービスタイプ：HT\_SERVICE

E) や、プログラムの実行コマンドでタイプを識別するパラメータ (CALCUL, SERVICE) に基づく場合があることが記載されているところ、C が計算 (calculation), S がサービス (service) の頭文字に由来することも明らかである。

エ そうすると、当初明細書等の記載を考慮して、特許請求の範囲に記載された用語の意義を解釈すると、本件補正後の請求項 6～8 で命令スレッドのタイプとされている記載「タイプ (S, C)」は、「サービスタイプ」、「計算タイプ」の意味であると解することができる。

オ 以上によると、本件補正は、本件補正前の請求項 6 において、命令スレッドの「タイプ」は、どのような種類のタイプが存在するのかについて、記載がなかったのを、「タイプ (S, C)」とし、当初明細書等に記載されていた「タイプ (サービスタイプ, 計算タイプ)」としたものであり、当初明細書等に記載された事項の範囲内を超えるものではない。

したがって、本件補正は、新規事項を追加するものではなく、原告の主張する取消事由 1 のうち、新規事項追加禁止要件違反についての判断の誤りを主張する点については、理由がある。

カ(ア) 被告は、当初明細書等においては、仮想プロセッサのタイプとして「C」及び「S」が用いられているものの、命令スレッドのタイプとして「C」及び「S」は用いられてはならず、命令スレッドのタイプとしては、「計算タイプ」及び「サービスタイプ」を用いているから、命令スレッドのタイプと仮想プロセッサのタイプは明確に使い分けられている旨主張する。

しかし、この主張を採用することができないことは、既に判示したとおりである。

(イ) また、被告は、「S」及び「C」は、参照符号であり、発明特定事項とはならず、単なる例示にすぎない旨主張する。

しかし、本件補正後の請求項 6 の「タイプ (S, C)」は、既に判示したとおり、「タイプ (サービスタイプ, 計算タイプ)」の意味であるから、単に「請求項の記載



の内容を理解するために必要があるときは、当該願書に添付した図面において使用した符号を括弧をして用いる」(特許法施行規則様式29の2[備考]14ロ)場合とは異なっており、発明特定事項であるといえることができる。

したがって、被告の上記主張は、採用することができない。

(2) 目的要件違反について

ア 前記(1)のとおり、本件補正により、請求項6の「タイプ」の後に付加される「(S, C)」は、「(サービスタイプ, 計算タイプ)」を意味するものと解される。

イ したがって、本件補正前は限定のなかった請求項6の「タイプ」に本件補正によって「(S, C)」を付加することにより、サービスタイプと計算タイプが含まれることを明らかにした上、それぞれの命令スレッドがそれぞれのタイプに応じて仮想プロセッサに方向付けられるものであることを特定したものであって、請求項6に、「サービスタイプと計算タイプがそれぞれのタイプに応じて仮想プロセッサに方向付けられる」という本件補正前にはない限定を加えたものであるから、「特許請求の範囲の減縮」に当たるといえることができる。

ウ 被告は、本件補正後の請求項6における「複数の所定のタイプ」という用語は、「命令スレッド」の「タイプ」を意味するものとして明確であり、「複数の所定のタイプ」に付加された「(S, C)」も、本願の願書に添付した図面において使用した符号を括弧をして用いたものとして解釈し得るところ、明細書の発明の詳細な説明で技術的に裏付けられているのが「S」及び「C」のタイプを使用するものだけであるとか、実施例が「S」及び「C」のタイプを使用するものだけであることのみから、特許請求の範囲に記載された「複数の所定のタイプ」を「少なくとも『S』及び『C』のタイプを含むもの」と限定して解することはできない旨主張する。

しかし、本願発明において命令スレッドの「タイプ」として「計算タイプ」と「サービスタイプ」以外のものがあるとしても、特許請求の範囲及び明細書の記載から、既に判示したとおり解することができる。

したがって、被告の上記主張を採用することはできない。

エ そうすると、本件補正は、特許請求の範囲の減縮を目的とするものに該当する。

したがって、原告の取消事由1のうち、目的要件違反についての判断の誤りを主張する点については、理由がある。

### (3) 小括

そうすると、本願補正発明6は、サービスタイプと計算タイプの各命令スレッドが、それぞれのタイプに応じて仮想プロセッサに方向付けられるものであるところ、独立特許要件違反についての審決の判断は、これを前提としていないから、前記の目的要件違反についての判断の誤りは、独立特許要件違反の有無を判断するまでもなく、審決の結論に影響を及ぼす。

そして、無効審判において、本願補正発明6につき上記のような解釈に基づく審理・判断がされたとは認められないから、本件については、再度特許庁において上記の解釈に基づく審理・判断を行う必要があるものと考えられる。

## 第6 結論

以上の次第で、原告の主張する取消事由1には、理由があるから、その余の点を判断するまでもなく、審決は、取消しを免れない。

よって、主文のとおり判決する。

知的財産高等裁判所第2部

裁判長裁判官

---

森 義 之

裁判官

---

永 田 早 苗

裁判官

---

森 岡 礼 子