



(19)中華民國智慧財產局

(12)發明說明書公告本 (11)證書號數：TW I474261 B

(45)公告日：中華民國 104 (2015) 年 02 月 21 日

(21)申請案號：102118985

(22)申請日：中華民國 95 (2006) 年 09 月 29 日

(51)Int. Cl. : G06F9/48 (2006.01) G06F9/50 (2006.01)

(30)優先權：2005/09/30 英國 0519981.5

(71)申請人：新思科技有限公司 (美國) SYNOPSYS, INC. (US)

美國

富士通半導體股份有限公司 (日本) FUJITSU SEMICONDUCTOR LIMITED (JP)

日本

(72)發明人：立派特馬克大衛 LIPPETT, MARK DAVID (GB)

(74)代理人：蔡坤財；李世章

(56)參考文獻：

TW 200405201A

TW 200519605A

US 6711447B1

US 6711691B1

審查人員：劉思芸

申請專利範圍項數：36 項 圖式數：74 共 153 頁

(54)名稱

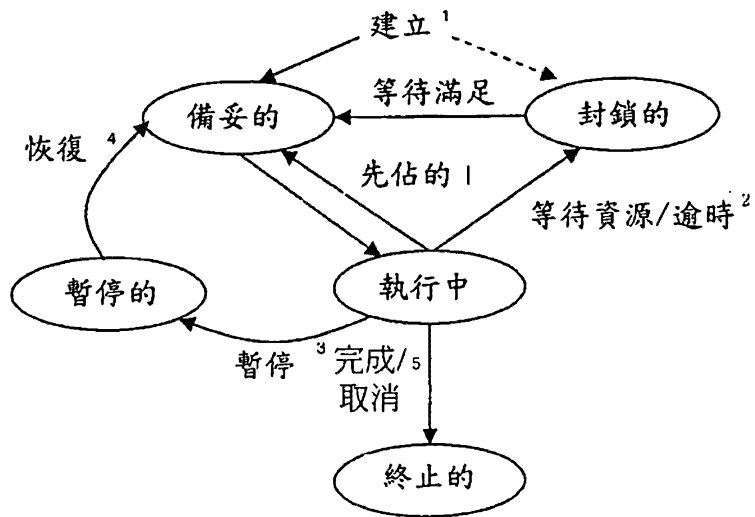
用於多核心架構之排程的方法、電腦程式、及電腦可讀取媒體

METHOD, COMPUTER PROGRAM, AND COMPUTER READABLE MEDIUM FOR SCHEDULING
IN A MULTICORE ARCHITECTURE

(57)摘要

此發明係有關於在一多核心處理器中排程線程之步驟。可使用依據執行適合度之順序列出可執行交易的至少一分配佇列以及含有多個連結之獨立可執行交易排程器之多層排程器來排程可執行交易。這些各者包括一排程演算法以供判定用於執行之最適可執行交易。該最適可執行交易係從該多層排程器被輸出到至少一分配佇列。

This invention relates to scheduling threads in a multicore processor. Executable transactions may be scheduled using at least one distribution queue, which lists executable transactions in order of eligibility for execution, and multilevel scheduler which comprises a plurality of linked individual executable transaction schedulers. Each of these includes a scheduling algorithm for determining the most eligible executable transaction for execution. The most eligible executable transaction is outputted from the multilevel scheduler to the at least one distribution queue.



第1圖：任務狀態圖

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】 (中文/英文)

用於多核心架構之排程的方法、電腦程式、及電腦可讀取媒體
METHOD, COMPUTER PROGRAM, AND COMPUTER READABLE
MEDIUM FOR SCHEDULING IN A MULTICORE ARCHITECTURE

【技術領域】

【0001】 本發明與在一多核心結構中的排程線程的一方法及設備有關。

【先前技術】

【0002】 近年來，已有傾向製造含有多核心之處理器以最大化矽效率(即“應用程式可用的” MIPs/mm² 或 MIPs/mW)。由於一線程定義含有一執行狀態、指令串流以及資料組的一獨立包裝 (autonomous package)，故此多核心結構理想的適於依據線程執行應用程式，而該線程定義上可伴隨其他線程同時執行。

【0003】 排程為對特定處理資源執行之最適線程 (亦即指令組) 之發現及分配的一般名稱，當其被執行時為該應用程式及該潛在硬體平台兩者所需要。

【0004】 在一多核心結構中的同時執行伴隨適於執行一特定線程之多核心的可得性將導致額外問題於用於在這些多核心結構中分配線程的排程之中。

【發明內容】

【0005】 依據本發明之一第一態樣，在如同申請專利範圍第

1 項所提出之一多核心處理器中提供一種排程可執行交易（transaction）的方法，其更常被稱為線程。

【0006】 藉由提供一多層級—更常被稱為階層式—排程系統，本發明允許建構一或多個較簡單排程演算法之外的複雜排程演算法。使用此複雜排程演算法的能力增進含有多線程之一應用程式在執行過程中的效能。本發明藉由更有效地分配該可執行交易或線程至處理資源而增進一應用程式的執行效能。此可增加執行速度並降低特定資源上的瓶頸。其亦可增加該多核心處理器之較不活動中的部分的使用。

【0007】 在該較佳實施例中，本發明係由一專屬硬編碼（hard-coded）故有效的實施例所實施。該較佳硬編碼實施例位於一伺服器-客戶端拓樸結構中，其包含系統伺服器（此處稱為 SystemWeaver）以及該多核心處理器中每個處理資源或核心的一客戶端。在處理資源之能力為有疑問的其他實施例中，一單一客戶端可聚集對多個處理資源的存取。

【0008】 爲了進一步增進整合本發明之整體系統的效能，該較佳實施例使用指標（pointer）於一專屬記憶體中以供排程分配、爲了選擇目的而儲存數值等等。這些指標器較佳地含有欄位以於建立該排程選擇時儲存參數，或僅儲存其他重要的數值。此後將該欄位整體地稱為度量（metrics）或運算子（operators）。

【0009】 依據本發明之第二態樣，提供一種在一多核心處理器中排程可執行交易的一種方法，該可執行交易定義一應用程式，而該多核心處理器具有多個處理元件，該方法包含維

護一可執行交易排程器的一階層式，其中該階層式被用於依據該應用程式在使用時的需求而排程可執行交易，其中該排程器階層式的每一層包含至少一排程器，而其中該至少一排程器包含至少一規則以於該處理器元件之一或多者上排序該可執行交易為最適於執行之交易的一順序。

【0010】 依據本發明之一第三態樣，提供一種在一多核心處理器中管理電源消耗的一方法，如申請專利範圍第 37 項所述。

【0011】 在一多核心處理器具有多個處理資源的情形中，其各自可執行一特定線程，本發明允許這些多處理資源被共同置於一集區（pool）中。隨後即由此處理資源之集區被分配線程。然而在此一情形中，當需要執行之線程數量並未超過該集區之執行能力時，亦即該集區中的某些處理資源並未被利用或者並未完全被利用時，本發明允許每個處理資源被置於一省電模式中。該處理資源甚至可具有多個不同的省電層級。

【0012】 較佳地，當線程執行負載需要時，處理資源被移出省電模式，或者至少被移回完全電源模式之相關成本較低的一較低省電模式。

【0013】 依據本發明之一第四態樣，提供一種在一多核心處理器中排程可執行交易或線程的方法，該多核心處理器包括可重新設置邏輯，如申請專利範圍第 38 項所述。

【0014】 在一或多個該處理器元件具有一可在配置邏輯之情形中，例如在一場式可程式閘陣列（FPGA）之可在配置執行部分的情形中，本發明可藉由聚集利用該邏輯之相同配置的線程而增進效能。此用於降低內容切換的需要或減少其衝

擊，即當該可在配置邏輯部分正被重新配置時。

【0015】 本發明亦可以一區域快取記憶體之形式增進可重新設置邏輯之情形中的效能，該區域快取記憶體儲存可供在疑問之處理元件上的次一執行的指令。

【0016】 此係由於區域快取遺失的影響，或者可藉由聚集使用快取記憶體之相同區域之執行的線程而最小化區域快取清理（flush）。

【0017】 依據本發明之一第五態樣，提供數位邏輯所執行之一電腦程式，其執行如申請專利範圍第 1 項之方法。亦提供含有該電腦程式之一電腦可讀取媒體。

【0018】 當使用本文中之描述的說明時，處理元件、處理資源、核心以及處理器將被視為均等物。一處理器元件之忙碌狀態等同於其目前工作負載。進一步的有利特徵將被定義於附屬之獨立申請專利範圍中。

【0019】 依據本發明之一進一步態樣，其提供一多核心處理器，該多核心處理器包含多個處理器元件、至少一分配佇列（distribution queue），該分配佇列列出可執行交易為執行資格（eligibility）之順序；並提供一多層排程器，其包含：多個獨立可執行交易排程器，其中每個獨立可執行交易排程器包括一排程演算法以從多個可供執行之候選可執行交易中判定最適於執行之交易；其中該排程器被連結在一起，而該多層排程器被用於自其中輸出該最適合可執行交易至至少一分配佇列。

【0020】 一可執行交易可包含一線程描述項，其可由多個狀

態中加以選擇。該線程描述項可依據一狀態變遷配置而於多個狀態之間被改變，因而識別該可執行交易，故可於多個可執行交易中加以管理以提供低排程延遲以及該排程階層式的完整性。該排程器可包含由多個排程狀態所選擇的一排程狀態。控制該排程器狀態以支援一動態排程階層式，其中可於該系統之正常操作過程中調整該排程階層式，而同時維護該階層式中被排程之元件的順序及完整性。

【0021】 該多核心處理器可更包含一硬體時間資源，其可被分配以提供一監看器逾時（watchdog timeout），而其中該監看器逾時指示一處理資源實體已進入一不可操作狀態。該硬體時間資源可替代地提供一時間片段（timeslice）逾時，其中該時間片段逾時指示一處理資源實體或處理資源實體群組係由多個同等適於執行之交易所平分。平分可包含提供該時間之一平分，或該可執行交易所需之時間比例的一平分。該硬體時間資源可被分配以於用於提供該監看器逾時之一第一模式以及用於提供該時間片段逾時之一第二模式之間切換。該硬體時間資源較佳地被用於在一時間片段逾時被提供時切換至該第一模式。

【0022】 該多核心處理器之等候管理員（pending manager）可更包含一計時器佇列，每個計時器佇列被用於接收計時器佇列元件。計時器佇列元件可包含可執行交易。該計時器佇列中的一第一可執行交易可與一第一時間參數相關。該第一時間參數指示該逾時時間，該逾時時間為該相關可執行交易應成為適於執行的時間。較佳地，該第一可執行交易之逾時

時間最靠近目前時間。一第二可執行交易可與一第二時間參數相關。該第二時間參數指示該第二可執行交易之逾時時間與該第一可執行交易之逾時時間之間的差異。一第三可執行交易可與一第三時間參數相關。該第三時間參數指示該第三可執行交易之逾時時間以及該第二可執行交易之逾時時間之間的差異。

【0023】 一佇列中的一第一可執行交易可與一時間參數相關。此時間參數指示該相關可執行交易之逾時與亦位於該佇列中之一第二可執行交易之逾時之間的差異。較佳地，該第二可執行交易為該佇列中先前已具有一逾時發生且最接近該第一可執行交易之逾時的執行交易。

【0024】 該多核心處理器可更包含多個分派佇列（dispatch queue）。較佳地，一分派佇列被用於識別一進一步分派佇列。每個分派佇列可包含一分派佇列描述項。此賦予一彈性數量的伺服處理資源實體，且亦使該分派佇列描述項得以被連續詢問。一分派佇列可被進一步用於從一組先佔（pre-emption）可執行交易中識別目前使用一處理器元件的一可執行交易。該分派佇列可被進一步用於從該組先佔可執行交易中識別一進一步可執行交易，其將接續地使用該處理器元件。此分派佇列因而保持該排程管理員所建立之最新排程選擇的索引。

【0025】 用於提供該多層排程器之個別可執行交易排程器的連結定義一排程階層式，其中每個可執行交易排程器具有一相關排程階層。一可執行交易排程器可用於識別一可執行交

易排程器，其先前排程一可執行交易。任意地，該可執行交易排程器可識別可執行交易是否源自與一處理器元件相關的一分配佇列。該可執行交易排程器可識別該可執行交易是否來自一組先佔可執行交易。可最佳化該排程事件為一“推進（push）”事件時的處理。當一可執行交易由一可執行交易排程器所排程時，該可執行交易排程器可被進一步用於通信一校正參數至先前排程該可執行交易的每個可執行交易排程器。該校正參數允許該排程選擇的傳遞以及該多層排程器中計數器之完整性的維護。

【0026】 亦可提供一種操作一多核心處理器系統的方法，其包含：提供一客戶端；以及選擇用於該客戶端的一互動狀態。該互動狀態可包含：一閒置狀態，其中該客戶端可被用於在一電源管理模式中操作；以及一使用者狀態，其中該客戶端被用於在一使用者或正常模式中執行一可執行交易。較佳地，該互動狀態可更包含一 API 互動狀態，其中該客戶端被用於在一優先狀態中執行一可執行交易。任意地，該互動狀態可包含一客戶端填隙狀態（shim state），其中該客戶端被用於準備一內容以供一可執行交易。較佳地，本方法亦包含提供一伺服器，其中該互動狀態被分享於該客戶端及伺服器之間。較佳地，可提供一帶外信號（out of band signal）以使該互動狀態改變。該伺服器可提供該帶外信號。任意地，一可執行交易可使該互動狀態改變。

【圖式簡單說明】

【0027】 第 1 圖顯示一任務狀態圖，其近似於在

SystemWeaver 中所實作之一者，其用於在一多核心系統中任務或線程的管理。

【0028】 第 2 圖顯示排程點及關於其所形成之該等佇列點的表示圖。

【0029】 第 3 圖顯示一軟體客戶端填隙之主要元件。

【0030】 第 4 圖顯示該 SystemWeaver 客戶端在動作中之一狀態圖。

【0031】 第 5 圖顯示一概念排程架構。

【0032】 第 6 圖顯示一排程器階層體系圖。

【0033】 第 7 圖顯示一範例排程錐形實作。

【0034】 第 8 圖顯示一典型處理資源集區組態。

【0035】 第 9 圖顯示在一任務描述項內所呈現的排程度量。

【0036】 第 10 圖顯示用於具有一單一處理資源實體之一單一處理器的組態以及其強制排程根節點。

【0037】 第 11 圖顯示一更具代表性之排程架構，其用於一單一處理資源實體。

【0038】 第 12 圖顯示該等 FIFO 排程層(tier)之一的細部檢視圖。

【0039】 第 13 圖顯示用於含有兩處理資源實體之一集區之此類型架構的表示圖。

【0040】 第 14 圖顯示位於一處理資源參與兩集區之一範例架構。

【0041】 第 15 圖顯示包含五處理資源實例及兩分配集區之一組態。

【0042】 第 16 圖顯示排程分析、策略及運算子。

【0043】 第 17 圖顯示在一基礎排程階層體系上對於一事件之重新排程範圍。

【0044】 第 18 圖顯示用於一單一兩實體處理資源集區之再排程範圍。

【0045】 第 19 圖顯示用於一單一推送操作之再排程範圍。

【0046】 第 20 圖顯示如第 18 圖中所示之推送事件的結果所發生般，該再排程之順序圖。

【0047】 第 21 圖顯示在處理資源集區中之度量傳遞。

【0048】 第 22 圖顯示該運行狀態之兩部分；正規的及反對的。

【0049】 第 23 圖顯示一典型時間片段組態。

【0050】 第 24 圖顯示一任務優先度差異圖。

【0051】 第 25 圖說明具有一單一處理器及三時間片段任務之情形。

【0052】 第 26 圖說明如第 25 圖般的相同情境，但是具有兩處理器。

【0053】 第 27 圖說明具有三處理器、三時間片段任務及一先佔任務之情形。

【0054】 第 28 圖顯示分配層排程。

【0055】 第 29 圖顯示當時間在一使用向上優先權之閒置狀態中流逝時，一處理資源實體(PRI)之「執行優先權」。

【0056】 第 30 圖顯示一重新組態管理結構。

【0057】 第 31 圖顯示一範例 SystemWeaver 系統結構。

【0058】 第 32 圖顯示用於最小內容猛移(thrashing)之排程器群組化。

【0059】 第 33 圖顯示一範例滯後排程組態。

【0060】 第 34 圖顯示根據第 33 圖之範例，滯後式排程之模擬結果。

【0061】 第 35 圖顯示一範例混合排程演算法。

【0062】 第 36 圖顯示對該等排程圖之關鍵。

【0063】 第 37 圖顯示一根據本發明之系統。

【0064】 第 38 圖顯示在該核心周圍上所尋得之介面群組。

【0065】 第 39 圖顯示該 SystemWeaver 伺服器實體之主要邏輯組件。

【0066】 第 40 圖顯示該 SystemWeaver 結構之主要子方塊。

【0067】 第 41 圖顯示該 SystemWeaver 內部線程概念圖。

【0068】 第 42 圖顯示一 SystemWeaver 排程器層狀態圖。

【0069】 第 43 圖顯示該線程排程器介面管理器主要輸入及輸出。

【0070】 第 44 圖顯示於每週期之監看岔斷控制的流程圖。

【0071】 第 45 圖顯示於每個系統時脈週期時，在該線程排程器介面管理器內部之時間片段支援邏輯的行為。

【0072】 第 46 圖顯示一動態計時器週期流程圖。

【0073】 第 47 圖顯示該線程排程器輸入管理器主要輸入及輸出。

【0074】 第 48 圖顯示該線程排程器輸入管理器內部結構。

【0075】 第 49 圖顯示該線程排程器擱置管理器主要輸入及

輸出。

【0076】 第 50 圖顯示該線程排程器擱置管理器擱置序列架構。

【0077】 第 51 圖顯示該暫停佇列之基本架構。

【0078】 第 52 圖顯示一動態時脈週期流程圖。

【0079】 第 53 圖顯示該暫停邏輯之操作模式的基本表示圖。

【0080】 第 54 圖顯示在線程 1 之提取後，一基本計時器佇列架構。

● 【0081】 第 55 圖顯示在線程 2 之提取後，一基本計時器佇列架構。

【0082】 第 56 圖顯示在線程 3 之提取後，一基本計時器佇列架構。

【0083】 第 57 圖顯示該線程排程器輸出管理器主要輸入及輸出。

【0084】 第 58 圖顯示該線程排程器輸出管理器內部結構。

● 【0085】 第 59 圖顯示該線程排程器排程管理器主要輸入及輸出。

【0086】 第 60 圖顯示一再排程操作之基本流程。

【0087】 第 61 圖顯示一單通排程操作。

【0088】 第 62 圖顯示用於提取及一般排程操作之內層排程。

【0089】 第 63 圖顯示層間排程之基本流程。

【0090】 第 64 圖顯示在該層間排程例式中之分派佇列處理。

【0091】 第 65 圖顯示在集區間排程中重複集區分配。

【0092】 第 66 圖顯示通過第 65 圖之集區間層排程操作的流

程。

【0093】 第 67 圖顯示集區間層排程之流程。

【0094】 第 68 圖顯示一基本排程階層體系順序圖。

【0095】 第 69 圖顯示一串接式排程階層體系順序圖。

【0096】 第 70 圖顯示一集區式排程階層體系順序圖。

【0097】 第 71 圖顯示該線程排程器排程管理器主要輸入及輸出。

【0098】 第 72 圖在一推送事件中於該線程排程器輸出管理器及線程排程器排程管理器之間的互動。

【0099】 第 73 圖顯示在一提取事件中於該線程排程器介面管理器、線程排程器輸出管理器及線程排程器排程管理器之間的互動。

【0100】 第 74 圖顯示該 SystemWeaver 伺服器子區塊間工作佇列結構。

【實施方式】

【0101】 第 1 圖顯示一系統架構 10 之邏輯圖，該系統包含根據本發明一實施例之功能特徵。該架構 10 至少包含複數處理資源（processing resources）150，每一處理資源可能與其他處理資源 150 中之資源相似或相異，且每一處理資源可能有不同之複雜性（complexity）。每一處理資源可共同存取一共用系統記憶體（common system memory）140，其共享之資料係透過一連接介面（interconnect）160 儲存。習知技藝者皆明瞭，並非所有系統記憶體 140，皆須讓所有處理資源 150 共用。

【0102】 第 1 圖說明與實施於 System Weaver 中類似的一任務狀態圖，其用於一多核心系統中的任務及線程的管理。

【0103】 在多核心系統中，該排程器依據一組預定規則（該“排程”）於最佳時間提供工作封包至該最佳資源。該應用程式以及潛在硬體平台均需要排程：

【0104】 -應用程式排程包含同步及適用性。同步確保該系統中的資源可被分享而無須妥協於該資料甚至可能為該系統整體的完整性。適用性確保準備的任務利用如該排程策略所表示之符合該應用程式之需求的一方式被送至該處理資源。

【0105】 -平台式/分配排程定義分配應用程式任務於適於處理資源實體之間的一策略。此可意指在多個使用者及/或多個不同演算法間分享一處理資源。

【0106】 第 2 圖說明排程點的一表示，該排程點係與其產生的佇列點有關。

【0107】 由左至右所遇到的第一佇列點為等候佇列。被封鎖之任務依據其優先度而被儲存於該等候佇列中且由同步事件所釋放，該等候佇列結構及行為的進一步描述已超越本發明之範圍。該第二佇列點為該準備佇列，其包含該應用程式及分配排程兩者。在第 2 圖中此被分為三個邏輯部分：該應用程式及分配佇列。概念上，在這兩排程階層之間的點上，所有目前準備應用程式任務已依據其適用性（描述於使用者定義之度量及排程策略中）加以排序。此點被稱為分配節點。

【0108】 在該分配節點之間配置一組應用程式特定排程策略，其判定不同等級之任務以及共同等級之任務實體如何競

相存取該處理資源實體。此排程策略之階層式被稱為排程錐（scheduling cone）且為應用程式特定的。

【0109】 在該排程錐之後配置一組平台特定排程策略，其判定該最適合的應用程式任務如何被分配至存在於該潛在硬體平台中的處理資源實體。此排程策略之階層式被稱為分配錐（distribution cone）且為平台特定的。

【0110】 一排程策略以及實施的有效性可於許多屬性之一結合上加以判斷：

● 【0111】 - 處理能力—每秒可建立的排程選擇數量。

● 【0112】 - 延遲—該系統中的一事件以及與該事件有關之排程操作完成之間消逝的時間。

【0113】 - 可預測性/決定性—決定該系統在所有狀態下將如何作為的能力。

● 【0114】 - 有效性—可實施任一特定排程演算法的效率。此可以每個選擇之指令（指令組效率的一測量）及/或矽腳印（silicon footprint）（記憶體及其他晶粒區域）加以測量。

● 【0115】 - 策略—支援策略之多樣性以及加以結合以形成複雜階層式的能力。

【0116】 元件

【0117】 SystemWeaver 有兩種主要元件：該伺服器核心以及該客戶端填隙（client shimes）。這些可用數種方式加以連接。— SystemWeaver 使系統可包含一伺服器核心以及至少一客戶端填隙。

【0118】 SystemWeaver 核心

【0119】 該 SystemWeaver 核心包含一硬體引擎以及一密接記憶體。該記憶體含有排程配置以及動態描述項用以表示該系統中的工作單位。在多個客戶端聚集每個 SystemWeaver 核心，其可為指令設置式結構或硬體加速器。SystemWeaver 透過兩個邏輯分離資料路徑分別與每個客戶端通信：

【0120】 - 一帶外信號以警告該客戶端在需要注意之系統狀態中的一改變。該 SystemWeaver 核心為此介面的主角，該介面典型地被實施為一岔斷（於此處及下文中假設）。

● 【0121】 - 該客戶端可藉以詢問 SystemWeaver 的一資料路徑。該客戶端為此介面之主角，該介面典型地被實施為一匯流排、雙工序列介面或任何其他雙向實施。

【0122】 該 SystemWeaver 核心在該啓動處理過程中必須被起始。典型地一客戶端將被指定為該啓動主角且將起始 SystemWeaver，而與其相關之記憶體代表該系統之其餘部分。

【0123】 SystemWeaver 客戶端填隙

● 【0124】 在一典型配置中，每個客戶端具有一獨立客戶端填隙，然而更保守之實施可聚集許多客戶端的客戶端填隙。可利用硬體或軟體實施客戶端填隙。第 3 圖說明一軟體客戶端填隙的主要元件：

【0125】 - 該 SystemWeaver HAL 實施該 SystemWeaver 核心之註冊介面所需的命令格式化。

【0126】 - 該 SystemWeaver API 使該應用程式得以於一任務式抽象上建立對該 SystemWeaver 的呼叫。

【0127】 - 該使用者線程為使用該 SystemWeaver 任務管理

能力的應用程式線程。在過程中的任何實體僅有一使用者線程直接地被每個獨立客戶端所管理。

【0128】 - 該客戶端填隙處理該帶外信號服務（典型地為岔斷服務）。該客戶端管理內容主要位於該處理資源實體中，其透過任務切換以及先佔而確保完整性的保存。典型地該客戶端填隙含有一結構未知部分以及一指令組結構特定部分。

【0129】 - 該閒置代理程式實施一管理任務，其處理獨立處理資源實體或巨結構的電源降低模式。

● 【0130】 第 4 圖說明該 SystemWeaver 動作中的一狀態圖。該客戶端填隙執行兩種主要功能：

● 【0131】 - 該處理資源實體執行於其中之“內容”（對於一傳統處理器而言，內容可包含處理器堆疊空間及內容、註冊值、程式計數器等等）的管理（在適當時間之配置、儲存及復原）。內容有兩種類型—使用者或任務內容，其為一使用者任務執行於其中的內容，；以及處理實體特定內容，其為專屬於客戶端填隙管理操作的一內容。

● 【0132】 - 操作模式（在一傳統處理器使用者（正常）以及管理者（優先的）模式中為可用的，其定義存取特定關鍵系統資源的權利—舉例來說，不像一管理者模式任務，一使用者模式任務將不被允許存取資源，其可不當影響其他使用者模式任務分享該處理資源）。

【0133】 以下描述係用於一般目的處理器，即使所有客戶端類型亦為類似的。

【0134】 - “閒置”狀態—在該閒置狀態中，使用者定義演

算法可利用該結束處理資源實體（例如時脈閘控或其他低電源狀態）或該系統結構整體（例如時脈閘控或降低或消除對一特定處理資源實體的電源供應）所提供的電源降低模式。然而在此狀態中該處理資源實體可操作於一優先模式中且將被使用該處理資源特定內容。

【0135】 注意該 SystemWeaver 伺服器並未指示該客戶端進入該閒置狀態—該客戶端係由於不具有排程任務而進入該閒置狀態。每個客戶端保持於該“閒置”狀態中直到被指示透過源自該 SystemWeaver 伺服器之帶外信號(典型地為一岔斷)加以回應。

【0136】 - “客戶端填隙”狀態—該客戶端填隙狀態管理該使用者的執行內容以及閒置任務。當位於該“客戶端填隙”狀態中時，該客戶端填隙儲存已完成執行、被先佔之任何任務的內容，或已封鎖及回復或建立用於該次一執行任務的一內容（在該閒置任務之情形中此為該處理資源實體特定內容）。然而在此狀態中該處理資源實體可操作於一優先狀態中。任何時間中該客戶端填隙可被操作於該處理資源特定內容或該使用者或任務內容中。

【0137】 由於源自 SystemWeaver 伺服器之帶外信號（自該“使用者”或“閒置”狀態轉換為該“使用者填隙”狀態）或由於已使該執行中任務成為被封鎖之一 SystemWeaver API 呼叫（“SyWAPI”狀態轉換為“客戶端填隙”狀態，例如由於嘗試鎖定一信號的失敗）而進入該“客戶端填隙”狀態。在完成處理時，該客戶端填隙可轉換該“客戶端填隙”狀態

為該“閒置”狀態（若該處理資源實體沒有傑出任務）或為該“使用者”狀態（若該處理資源實體具有適合的任務存在）而不需要任何進一步的外部傳信。

【0138】 - “使用者”狀態—當位於該“使用者”狀態中時，該客戶端填隙執行使用者應用程式碼。然而在此狀態中該處理資源實體可正常地操作於一“使用者”或“正常”模式中。該“使用者”狀態可完全地操作於該使用者或任務內容中。

● 【0139】 可由於開始或恢復一使用者任務而從該“客戶端填隙”狀態進入該“使用者”狀態，或由於源自一SystemWeaver伺服器API呼叫的一傳回而自該SyWAPI狀態進入該“使用者”狀態。

● 【0140】 該客戶端填隙可由於任務完成或先佔（自該SystemWeaver伺服器接收帶外信號）而自該“使用者”狀態轉換為該“客戶端填隙”狀態。可由於對該SystemWeaver伺服器API之一呼叫而自該“使用者”狀態轉換至該客戶端填隙。

【0141】 - “SyWAPI”狀態—當一使用者任務需要與該SystemWeaver之互動時，其透過使一客戶端填隙狀態改變至“SyWAPI”的SystemWeaver API而進行。然而在此狀態中該處理資源實體可操作於一優先模式中。該“SyWAPI”狀態將完全操作於該使用者或任務內容中。

【0142】 在對該SystemWeaver API之一呼叫後進入該“SyWAPI”狀態。對於非封鎖呼叫而言，該客戶端填隙將自

該 “SyWAPI” 狀態返回至該 “使用者” 狀態，然而某些存取（例如該些與信號相關者）可使該使用者任務成為被封鎖的（封鎖的任務必須等到某些分享的系統資源成為可用的）。在此情形中該客戶端填隙轉換為該 “客戶端填隙” 狀態。

【0143】 概念

【0144】 以下段落討論瞭解該 SystemWeaver 排程器所需的觀念。

【0145】 SystemWeaver 記憶體元件

【0146】 SystemWeaver 需要一附加密接記憶體，此記憶體之使用以供排程策略之儲存可賦予完全排程修改以及該系統發展處理過程中的調整彈性。該 SystemWeaver 記憶體被分為 SystemWeaver 記憶體元件（WMEs）。WMEs 被用於表示以下討論之任務及排程描述項。

【0147】 任務描述項

【0148】 任務描述項為該 SystemWeaver 結構的關鍵 “流通單位（unit of currency）” 其表示依據該排程階層式中配置之規則而競相存取處理資源實體之任務的單位。任務描述項包含：

【0149】 - 對於一任務控制封鎖的一參照，其依序含有對於待執行之任務以及當其被執行時之資料組的一參照。

【0150】 - 排程度量，其定義該任務之資格。

【0151】 - 亦可包括同步參照以及逾時用於初始將被封鎖的任務。封鎖任務行為的一更詳細描述已超過此文件之範圍。

【0152】 - 對一 “登錄節點（Entry Node）” 的一參照，其

定義必須加入該任務描述項（可能在同步之後）之排程階層式的部分。

【0153】 排程及分配錐

【0154】 使用兩種分配錐類型以描述 SystemWeaver 排程行為：排程及分配錐。排程錐被用於描述聚集許多“登錄”點為一單一匯聚點——多對一映射——之排程器的一階層式。分配錐被用於描述自一單一匯聚點散布至多個“分派（dispatch）”點——一對多映射——之排程器的一階層式。

● 【0155】 排程錐

【0156】 排程錐（如第 5 圖中紅色所示）定義該“應用程式選擇節點”階層式，其係由該應用程式之需求所驅動（亦如第 2 圖之準備狀態中的“應用程式排程”所示）。排程錐為多對一映射，其定義多類任務及多類任務實體藉以競逐系統資源的規則。

● 【0157】 分配錐

【0158】 分配錐（如第 5 圖中紫色所示）定義該“分配選擇節點”階層式，其主要由該潛在硬體平台之屬性所驅動（亦如第 2 圖之準備狀態中的“分配排程”所示）。分配錐定義該排程錐之最適候選者藉以被分配於該可取得及適當處理資源之間的規則。

【0159】 主要排程節點

【0160】 有三種主要節點被用於描述排程配置：登錄節點、分配節點以及分派節點。該主要節點為該潛在第二節點結構之上的一重疊，其更接近地反映該排程器的詳細實施。

【0161】 登錄節點

【0162】 登錄節點定義加入新任務至佇列中的位置。登錄節點典型地多對一映射於分配節點上，如同一排程錐的兩末端。登錄節點可與特定任務類別有關，或依據其他應用程式導出的策略。一特定登錄節點可僅映射至一單一分配節點之上。

【0163】 分配節點

【0164】 分配節點定義排程及分配錐之間的輪廓。其典型地表示處理資源之一種類別。排程錐典型地映射一或多個登錄節點至一單一分配節點之上，而分配錐典型地映射一單一分配節點至多個分派節點之上，因而最終處理資源實體。

【0165】 分派節點

【0166】 分派節點定義與一獨立處理資源實體有關的一出口點。其典型地將利用存在於該硬體平台中的 IP 核心（即使硬體多線程處理器核心可被分配於多個分派佇列）一對一地映射。多個分配錐可映射於獨立分派節點之上。

【0167】 第二排程節點

【0168】 定義兩種類行的選擇節點：應用程式選擇節點以及分配選擇節點。雖然應用程式以及分配選擇節點直接映射於排程層上，但其並不徹底地定義該排程器實施中潛在層的數量或類型。

【0169】 應用程式選擇節點

【0170】 應用程式選擇節點定義過渡排程或該排程錐中的聚集點。每個應用程式選擇節點定義一集合之最適候選者可被

選擇的一規則。

【0171】 分配選擇節點

【0172】 當多個分配節點映射至一單一分派節點上時，需要一分配選擇節點以設置判定取得對該處理資源實體之存取之分配錐的策略。

【0173】 排程配置

【0174】 該準備狀態結構（第 2 圖）含有準備執行之線程。該整體準備狀態結構可含有數個排程及分配錐。這些線程係以獨立線程基本指令所建立（亦即其係於該準備狀態中所建立），或者其已於依賴時接收該同步基本指令或逾時。同步線程先前已自該封鎖狀態而轉換。

【0175】 該準備狀態結構可含有排程器節點描述項以及獨立線程描述項。該結構主要在系統初始化過程中被定義，即使線程描述項以及其相關的動態排程器層描述項在執行期間中被允許來來去去。

【0176】 該準備狀態結構允許線程被排程至處理節點或一特定處理節點的一集區中。此賦予多個相容處理資源之間的負載平衡或其他分配行為，同時維護在特定可用處理資源（例如硬體加速器或 IO 裝置）上針對特定任務的能力。

【0177】 排程層為用於實施建立該準備狀態結構之主要及第二排程節點的基本指令資源。排程層與其他排程層及任務描述項可具有主層（parent）、子層（parent）或對等（peer）關係。

【0178】 第 6 圖說明一排程器階層式圖，其表示該主與從關

係。在此範例中，y 為 a、b 及 c 的主層。y 為 x 及 z 的對等層。主層可僅為排程層，而子層可為排程層或任務描述項。一特定對等組（例如 a、b 及 c）可由任務描述項及排程層之一混合所組成。再者，所有排程層具有主層（該分派節點為唯一不定義一主層的描述項）。

【0179】 在執行過程中，主層可依據使用者定義策略而自該最適子層承繼“度量”（優先度等等）。此特徵可被用於深嵌排程策略向被比較之排程分支最適候選者之某些認識時（此議題將於以下度量傳遞章節中加以詳述）。

【0180】 以下章節描述任何 SystemWeaver 排程階層式之建構區塊。

【0181】 基本排程器層

【0182】 排程器層定義用於排程線程描述項之階層式。每個排程器層典型地定義一排程演算法、用於定義排程選擇之某些度量、用於定義如何自該子層傳遞該度量至該主層之一承繼策略、以及可為進一步排程器層或線程描述項之子層元件的一列表。有三種類型的排程器層描述項：根、靜態及動態（其中動態層為靜態排程層的一特別類型）。第 7 圖說明一示範排程錐實施。第 36 圖說明第 4 圖之後所有排程圖所需的圖示參照圖例。

【0183】 排程器根描述項

【0184】 排程器根描述項具有與分派佇列的一一對一映射。其表示該準備狀態結構中的根本節點（ultimate node）。根描述項度量總是含有依據該定義之承繼策略而自該準備狀態結

構導出之度量的的一副本。

【0185】 排程器根描述項在系統初始化過程中被配置且永遠存在。

【0186】 靜態排程器描述項

【0187】 排程器靜態描述項在該排程階層式中存在於根節點的下方。靜態排程器描述項的主層可為其他靜態排程器描述項或根描述項。其依據其主層定義之排程器演算法以及其自身之排程器度量而與同輩節點（sibling node）相競爭。

● 【0188】 排程器靜態描述項在系統初始化過程中被配置且永遠存在。在操作過程中，SystemWeaver 依據該選擇排程以及度量傳遞演算法而維護該排程器度量。

【0189】 動態排程器描述項

● 【0190】 排程器動態描述項在該排程階層式中存在於根節點之下方，且選擇性地存在於靜態節點之下方。動態排程器描述項之主層可為靜態排程器描述項或根描述項。其依據其主層定義之排程器演算法以及其自身之排程器度量而與同輩節點競爭。

【0191】 動態排程器描述項可於任何時間被配置。此使該系統得以支援遠大於以一單純靜態條件可得的排程層數量。SystemWeaver 藉由提高在一有限期間中該短暫需求會較小的可能性而達成此一效果—即使在整體時間中使用大量且多種的線程及動態排程器層。舉例來說，在具有最大支援 4k 動態元件（線程及動態排程器描述器）之附加記憶體的一網路系統中，可於任何瞬間支援 16k 連線，該整體連接空間之一部

分的資料單元在該處理器中為使用中的。僅需少量犧牲效能便可達成此彈性，然而若不具有一動態排程描述項，其必須在加入子線程描述項之前輩建立。

【0192】 在操作過程中，SystemWeaver 依據該選擇之排程演算法維護該排程器度量。在某些情況下 SystemWeaver 將釋放該動態排程器描述項回到該 WME 閒置列表。

【0193】 處理器資源集區

【0194】 處理器資源集區使一特定處理資源之實體得以聚集為一單一分配節點。該分配節點隨後可提供該處理資源集區之個別成員之間的負載平衡、智慧先佔以及電源管理。

【0195】 第 8 圖說明一典型處理資源集區配置 Weaver。記憶體元件的三種新定義支援該處理器集區配置結構：

【0196】 集區附加節點

【0197】 集區附加節點（PAN）被用於附加該排程器根層至處理器資源集區根層。PANs 必須存在於該排程器根層（即其主層必須為一排程根節點）。在操作過程中，PAN 度量係以自該排程錐承繼之集區根節點（PRN）之度量的一副本而加以自動更新。

【0198】 並未使用定義於該 PANs 中的排程運算子。

【0199】 集區靜態節點

【0200】 集區靜態節點（PSN）被用於附加該排程器根層至該處理器資源集區根層。其存在於該集區根層中（亦即其主層必須為一 PAN）且自動保持該分派節點之度量的一副本（即目前執行中的線程）。

【0201】 位於一特定集區之 PSN 中的排程器運算子全部必須被設置為相同的演算法，其定義用於選擇將被先佔之適當處理資源實體的策略。

【0202】 每個處理資源集區有一單一集區根節點（PRN）。該集區根節點定義該處理資源集區的分配節點。該 PRNs 中的度量反映在與該分配節點有關之排程錐中保持之最適線程。該 PRN 主層指標必須被設置以指向該集區靜態節點之一。

【0203】 正常情形中應依據該排程錐之鄰近層之需求而設置該排程器演算法。

【0204】 動態排程器配置

【0205】 SystemWeaver 支援執行中排程節點之建立及刪除，並且提供自一登錄節點遷移一任務類型至另一者而不會遺失或失序的能力。當討論動態排程器配置時必須引入兩種額外概念：靜止（dormant）排程層以及標記線程

【0206】 - 靜止排程層存在於該階層式中且可接收推送操作（push operations）（亦即可聚集子登錄）但不得用於排程，因而永不動作。

【0207】 - 標記線程僅於其為該排程階層式之一特定部分上的最後線程隨從（dependant）時加以排程。在該排程階層式之一部分上的線程隨從數量包括準備線程之數量以及封鎖線程之數量，後者於其成為準備時而使用此部分之排程階層式。該標記線程可如同任何其他線程而攜帶任務參照，且將典型地被用於完成該排程階層式之一部分與另一部份之間的一轉換操作的管理。

【0208】 以下章節詳述該階層式之一部分之一任務串流之一轉換的一示範順序。注意此為排程階層式之一部分之刪除的一超集合（superset）。

【0209】 操作順序

【0210】 高層軟體有責任確保操作之適當順序被遵守。無法遵守此順序將導致無法預料之行爲，特言之，新線程必須被引入至已插入一標記線程至其中的排程階層式的一部分。

【0211】 在此示範順序中假設一任務之串流 `tstream` 自排程器階層式 `h1` 被轉換至新排程器階層式 `h2`。

【0212】 - 建立靜止排程器階層式 `h2`。

【0213】 - 分配 `tstream` 上的所有新任務描述項至 `h2`。

【0214】 - 插入一標記線程至 `h1` 中。

【0215】 - 等待標記線程的出現。

【0216】 - 喚醒該靜止階層式 `h2`。

【0217】 排程器分配、演算法、運算子以及運算域

【0218】 排程分析採取許多形式（EDF、RMA 等等）且典型地為應用程式一至少為區段一特定的。排程分析之結果為一組策略，其靜態或動態地控制該應用程式之執行時配置。透過其獨特的微結構，SystemWeaver 於執行時有效地執行這些預定之策略/演算法。

【0219】 排程演算法

【0220】 SystemWeaver 被用於使特定演算法得以被定義於矽設計時間而不會破壞結構或實施。然而，預設提供數個演算法：

【0221】 - FIFO 排程：單純的先進先出佇列化。

【0222】 - 優先度排程：最適候選者具有最高（升冪優先度）或最低（降冪優先度）優先度度量。

【0223】 - 循環排序（Round Robin）分配：當一任務自該排程階層式中提取時更新該排程選擇為次一對等層。注意該循環排序分配並非未於一排程階層式之“最左側”終端的一相關排程策略。

【0224】 - 加權公平佇列化：一種複雜排程器，其中的適格候選者係依據一分配權值以及某種負載測量（即封包之長度）而加以選擇。

【0225】 藉由對整體排程器階層式之部分注意，可建立排程器演算法之複雜組合以提供應用程式系統中的精密流量及任務管理能力。

【0226】 運算子

【0227】 排程演算法被進一步分解為個別排程及度量運算子，兩者均被定義於該主層節點中：

【0228】 - 排程運算子：定義儲存於該子層節點中的運算域被用於判定該最適候選者的方式。排程運算子並未修改該子層節點中的運算域。

【0229】 - 度量運算子：定義最適子層之運算域被傳遞至主層運算域的方式。該傳遞運算子可為空值（不更新該主層）、一複製（覆寫該主層的運算域）或涉及在某個或所有該子層及主層運算域上的一數學函數。在所有情形中，該子層運算域不被改變。

【0230】 該排程及度量運算子被內部地實施於該 SystemWeaver 排程器硬體中。排程及度量運算子的一組合將典型地被用於定義一特定排程器演算法。排程器演算法一般在一推送事件（push event）（其中一新任務被推送至該排程器階層式中）而非一提取事件（其中一任務係自該排程階層式提取）下係指不同的行為。舉例來說，考慮一 FIFO 排程器，當一新任務被推送至一非空的 FIFO 排程階層中時，並不執行排程更新；然而當一元件係自一 FIFO 排程階層提取時，該排程器必須被更新。

【0231】 排程運算子

【0232】 排程運算子被設計為可延伸性的，但定義預設運算子的一選擇。該排程運算子典型的為比較性的，因而該結果總是為一布林值。在下表中 M 表示在一排程層或該排程層描述項本身之任一者的成員中的兩度量之一，此係依據以下方案：

【0233】 - Mcurrentn 係指屬於該目前最適格候選者的該些度量。

【0234】 - Mtiern 係指屬於附加該目前及候選者之排程器層描述器的該些度量。

指令	排程器更新選擇
GTR	$M_{current0} > M_{candidate0}$
GTR_EQ	$M_{current0} \geq M_{candidate0}$
LESS	$M_{current0} < M_{candidate0}$
LESS_EQ	$M_{current0} \leq M_{candidate0}$

EQ	$M_{\text{current}0} = M_{\text{candidate}0}$
UPDATE_EXIT	TRUE
UPDATE_NO_EXIT	TRUE
NO_UPDATE_EXIT	FALSE
HYSTERESIS	$(M_{\text{current}1} + M_{\text{tier}1}) < M_{\text{candidate}1}$

第 1 表：排程運算子

- 【0235】 混合排程運算子(Compound scheduling operators)
- 【0236】 混合排程運算子亦為可用的，其為第 1 表中該排程運算子之組合。舉例來說：
- 【0237】 $\text{Update Required} = (M_{\text{current}0} > M_{\text{candidate}0}) \&\& (M_{\text{current}1} < M_{\text{candidate}1})$
- 【0238】 其中討論中的參數可利用主層及子層描述項兩者中的度量。這些混合運算子可被用於傳統排程層及集區分配層兩者之中。
- 【0239】 關於進一步資訊及範例，可參閱以下的排程順序圖章節。
- 【0240】 度量運算子
- 【0241】 度量運算子一般為運算的（arithmetic）。至於排程運算子被設計為可延伸的但具有一組預設運算子，參見第 2 表。度量運算子範圍複雜，從空值或單純複製操作至複雜的乘法累積操作。

指令	排程元件度量		舊度量內容		新度量內容	
	0	1	0	1	0	1
ALL	A	B	X	Y	A	B
NONE	A	B	X	Y	X	Y

PROP_0	A	B	X	Y	A	Y
PROP_1	A	B	X	Y	X	B
PROP_WFQ	A	B	X	Y	$X+Y*B$	Y
ADD	A	B	X	Y	$X+A$	Y
SUB	A	B	X	Y	$X-A$	Y

第 2 表：度量運算子

【0242】 運算域

【0243】 排程運算域或度量被分為兩群組：

【0244】 - 區域度量係與處理資源實體、排程器層及線程描述項相關。一區域度量的操作自動地產生一再排程事件。

【0245】 - 整體度量為選擇性的，且典型地係與系統資源狀態（某些探索（heuristics）例如匯流排繁忙或空閒記憶體）有關。

【0246】 一特定排程演算法可僅使用兩度量，其中之一必須為區域性的。該第二度量的類型係由 Metric1lsGlobal 旗標所決定：

【0247】 - 當 Metric1lsGlobal 被重置時，度量 1 為區域的且將被使用作為排程操作中的一常值（literal）。

【0248】 - 當 Metric1lsGlobal 被設置時，度量 1 為進入整體度量埠之陣列中的一索引。

【0249】 區域度量

【0250】 任務描述項及排程層描述項含有兩個 32 位元運算域或排程度量。這些運算域由其個別之主層在排程操作中所使用且可於該排程操作中被轉換及/或傳遞至該主層之運算域

中，以供該階層式中的後續更高排程。

【0251】 第 9 圖說明表示於一任務描述項中的排程度量。在一排程描述項中，度量 0 將典型地被用於表示該任務之優先度。保留此度量之最不重要位元組以供該 SystemWeaver 硬體及客戶端軟體中的內部使用。在排程器層度量上沒有此限制。

【0252】 整體度量（Global metrics）

【0253】 整體度量一般為被動的，該整體度量中的一改變並未產生一再排程事件於所有潛在影響的排程資源上整體度量在排程一依賴排程資源時的某些其他系統事件被詢問。雖然該 SystemWeaver 結構對整體度量的使用不施加任何限制，但其可被用於系統探索（匯流排利用、記憶體填滿一時間窗格等等），因而該改變率將相對較低。亦可使用過濾器以平均該資料。

【0254】 排程階層式配置細節

【0255】 下文中參照之所有配置圖使用如第 36 圖所示之一共用格式。

【0256】 第 10 圖中說明一單一處理器的最基本配置，此說明一單一處理資源實體（單一分派節點）的配置，伴隨其必要的排程根節點。在此最簡單的情形中，由於僅有一單一處理器，故該排程錐係由一單一 FIFO 階層所組成且該分配階層為空的。因此，該排程器根節點同時為該登錄節點以及該分配節點。

【0257】 注意圖中所示之排程節點實施上的箭頭係由右向左（主層至子層），此與任務之“流程”相反，後者係由子層

流向該處理器資源實體。

【0258】 以一模組化方式實施排程器並配置於依電性記憶體中，此使得非常精密之排程器階層式可由一串不同策略之排程器層所構成，且在該發展處理過程中可加以調整及修改。然而，無效的配置亦為可能的，且必須採取特別注意以確保適當的度量可用於深層巢狀之排程器層。

【0259】 內層結構

【0260】 該排程層依據到達之順序或依據一預設 FIFO 佇列策略而儲存登錄。該定義排程策略被覆蓋於此結構上的方式將於稍後加以描述。新節點（或描述項）藉由一推送操作被增加至該內層結構，且由於一提取操作而被移除。排程操作並未控制該內層連結。

【0261】 第 11 圖說明一單一處理資源實體的一更代表性排程結構。由左而右，在此範例中兩 FIFO 階層構成一優先階層。在階層式之兩階層具有三排程層。注意該排程層僅具有一“離開”節點（顯示於該圖之右側），但可能具有許多登錄節點（顯示於該圖之左側）。

【0262】 第 12 圖說明該 FIFO 排程層之一的詳細圖示。此圖示說明一組指標，其維護該層上所有對等層之間的一雙連結列表。一雙連結列表被用於最大化該層之一任意成員之移除（提取）的效能。

【0263】 雖然此詳細圖示僅說明任務描述項，該結構同樣適用於含有線程及排程節點之任何混合的層級。

【0264】 對等元件之間的內層連結僅於推送及提取操作過程

中被控制。

【0265】 層間結構

【0266】 除了集區根層之外，第 12 圖中說明層間連結之結構。每層具有一主節點，其必須為一排程根節點或一排程節點兩者之一。這些節點儲存對此層之最適成員的一指標。在依據該排程策略以及分別定義於該主節點及該子節點中的度量而收到一排程事件時更新這些子節點。

【0267】 每個子節點必須亦參照其主層。

【0268】 集區根層結構

【0269】 該集區根層結構為一層級具有一單一登錄節點及許多離開節點的一特殊情形。該登錄節點為一排程錐聚集時（如第 2 圖中該準備佇列結構之“應用程式排程”部分所示）的點，亦稱為該“分配節點”。該“離開節點”連結該集區根層至可分配任務之處理資源實體的“分配排程”結構。第 13 圖說明用於含有兩處理資源實體之一集區之此類型結構。

【0270】 每個集區分配層必須含有一集區根節點（PRN）以及一或多個集區靜態節點（PSN），但不允許其他節點類型。該 PRN 含有對該排程錐之第一層的一參照（儲存於該 HeadIndex 欄位中）以及將被考慮分配之第一 PSN 登錄的參照。一共用分配及度量更新策略必須被儲存於每個 PSN 個別之該排程器及度量推送及提取運算子之中。

【0271】 每個 PSN 必須參照該 PRM 為其子層（使用該 HeadIndex 欄位）。

【0272】 一集區靜態節點之主層必須為一集區附加節點

(PAN)。PANs 及 PSNs 必須具有一一對一映射。然而，每個處理資源實體可具有多個與其參與之每個分配集區有關的 PANs。第 14 圖說明一處理資源參與兩集區中的一示範結構。對於特定處理資源可為其中之一成員之集區的數量不施加任何限制。再者，任何集區可與一任何數量的其他集區分享一任意數量的其組成處理資源。

【0273】 在該排程根節點中具有與該處理資源實體參與之各個分配錐有關的兩 PANs。此外具有一排程節點，其於需要時提供對於該處理資源實體的特定存取。

【0274】 定義於一集區分配層之每個 PSN 中的排程策略定義選擇最適處理資源實體以供執行一特定任務的方法。舉例來說，該策略之一可為優先度，其中選擇目前執行該最低優先度任務之處理資源實體以於一高優先度任務自該相關排程追到達時被先佔。

【0275】 第 15 圖說明含有五處理資源實體以及兩分配集區的一配置。注意 PRI#3 參與於兩集區中。

【0276】 行為

【0277】 下文描述 SystemWeaver 排程的行為。

【0278】 一般原理

【0279】 下文提供最基本背景資訊，其說明該 SystemWeaver 排程結構之關鍵基本原理。

【0280】 指標式佇列化

【0281】 雖然第 2 圖中詳述在 SystemWeaver 中具有多個潛在佇列點，但其僅使用指標而加以實現。該佇列實體一一

SystemWeaver 記憶體元件（WME）—從未被複製。

【0282】 事件式排程

【0283】 SystemWeaver 僅於系統狀態中的某改變要求時更新排程選擇。該狀態改變可被分為三種事件類型：

【0284】 - 一“推送事件”，其中該系統狀態之改變造成一新線程描述項被引入至該準備佇列結構中（注意此可為一新線程描述器，或為該系統狀態之改變已造成該線程成為準備的一既有線程描述項）。

● 【0285】 - 其中該系統狀態之改變造成一線程描述項自該準備佇列結構中移除。

【0286】 - 一“更新事件”，其中排程參數已被修改為需要該排程選擇之重新評估。

【0287】 這些改變可為：

【0288】 - 一岔斷（一“推送事件”，由於與該岔斷相關之區塊線程被移至該準備狀態）。

● 【0289】 - 由一執行任務所建立之一新任務的到達（此可為一推送事件，若該新任務並非依據其他因素或事件）。

【0290】 - 一同步事件- 舉例來說，一號誌信號（semaphore signal）（假設有一區塊線程等待該信號，此於該區塊線程描述項轉換為該準備狀態時為一“推送事件”）

【0291】 - 一任務之執行“優先度”的一改變，一“更新事件”。

【0292】 - 在一處理資源實體中（一“提取事件”）中一任務的使用（從準備轉換為執行中）。

【0293】 - 一任務之排程度量的修改（一“更新事件”）。

【0294】 - 一排程器層排程演算法或度量的修改（一“更新事件”）。

【0295】 其中位於準備狀態 SystemWeaver 中的系統保持閒置的。原則上，在該最有效電源解決方案中，此可使 SystemWeaver 被降低電源等待需要額外排程之一事件的到達。注意對整體度量的改變不會造成重新排程事件。

【0296】 “即時（Just in time）”排程

● 【0297】 被排入佇列至一特定排程層的新登錄僅與該目前最適登錄（由該主層 HeadIndex 所識別）加以比較。依據該排程層策略，若其較目前標題更為適合，則更新該 HeadIndex 欄位以參照該新登錄。新登錄總是被置於該目前連結列表結構的後方。

● 【0298】 若該排程策略為一 FIFO，則該 HeadIndex 指標於一新登錄到達時永不被更新，除非該佇列為空的。因此該預設行為於該新登錄被置於該佇列之後方時等於一 FIFO 演算法。

● 【0299】 此方案確保花費最少時間處理推送操作，其一般被視為排程效能上的延遲。因此，提取排程在必須評估一排程層之整體內容以於每個提取操作上更新該排程選擇的最壞情形中為更加繁重的。然而，在該實體佇列結構中總是使用一區域 FIFO 演算法為較佳地，由於該排程演算法之修改並不要求該排程器層被重新連結。再者，提取排程一般可伴隨應用程式執行而被執行，因此對於整體系統效能具有一較少衝擊。

【0300】 排程分析、策略及運算子

【0301】 有數種分析系統的方法以確保即時截止期限之滿足，其範例為 EDF（最早截止期限優先）、RMS（速率單調排程）以及各種其他隨機（stochastic）方法。這些方法一般傾向為應用程式特定的且可能為專屬的。然而，在所有情形中此排程分析之結果為排程策略（即優先度、FIFO、循環排序、加權公平佇列化）之一集合。SystemWeaver 技術係針對排程分析所識別之策略的有效即時執行。對於 SystemWeaver 中的配置而言，每個排程策略被進一步解碼為一組排程運算子。

【0302】 每個排程器層具有兩運算子，其被用於判定該排程選擇如何依據對該排程層（或從屬排程層）之一推送或依據由該排程層（或從屬排程層）之一提取而被更新。在某些情形中，該排程運算子將需要演算域，其被儲存於排程器及任務描述項兩者之類的度量欄位中。

【0303】 排程度量及度量傳遞運算子

【0304】 排程度量儲存可為該選擇排程演算法所需的資訊，其最基本之範例為優先度。在某些情形中，有必要自該最適候選者轉送度量至主節點，因而該資訊可直接於後續排程選擇中被使用，或作為一度量更新操作中的一操作域。該度量傳遞運算子定義對於推送及提取方案兩者而言如何達成此結果。

【0305】 依據該階層式中的排程節點之位置，該度量欄位亦可反映在一特定處理資源上該目前執行線程的優先度。在此情形中，其被用於判定是否需要一先佔（參見下文之排程行

爲）。

【0306】 排程資源

【0307】 下文描述用於在執行中實施該排程演算法的各種資源。

【0308】 層級

【0309】 排程器層

【0310】 一排程器層係由一主層—其可爲一排程器根節點、一集區根節點或一基本排程節點—以及數個子層所構成，如第 6 圖所示。該子層可爲基本排程節點、線程或任務描述項或集區附加節點。藉由使子層節點可爲其自身權限中的排程節點（即進階排程層之主節點），可建立複雜排程層階層式。

【0311】 集區分配層

【0312】 集區分配層可僅含有集區根節點（僅有 1 個）以及集區靜態節點。每個處理類別僅有一集區根節點。

【0313】 分配佇列描述項

【0314】 - 排程器運算子：用於定義判定該目前執行任務是否應被先佔之排程策略。

【0315】 - 度量傳遞運算子：在一分派佇列描述項中沒有度量傳遞運算子。

【0316】 - 度量：該度量元件一般儲存該目前執行線程的度量。

【0317】 排程器及集區根節點

【0318】 - 排程器運算子：用於判定該排程錐之最適候選者。

【0319】 - 度量傳遞運算子：其總是被設為承繼該排程錐之最適候選者的度量。

【0320】 - 度量：保持該排程錐之目前最適候選者的度量。

【0321】 排程器層元件

【0322】 - 排程器運算子：用於判定該附加層的最適子候選者。

【0323】 - 度量傳遞運算子：由使用者定義。依據後續排程階層之需要而設置。

● 【0324】 - 度量：使用者定義的。依據後續排程階層之需要而設置。注意某些度量傳遞運算子將自動更新這些欄位。

【0325】 集區靜態節點

【0326】 - 排程器運算子：用於判定在一集區分配層中用於先佔的最適候選者。

【0327】 - 度量傳遞運算子：用於判定該執行任務度量的傳遞。

● 【0328】 - 度量：依據該集區分配演算法之需求而設置。預設地其將反映該目前執行線程的度量；然而對於特定分配策略可能需要一靜態分配。

【0329】 集區附加節點

【0330】 - 排程器運算子：不使用。

【0331】 - 度量傳遞運算子：用於控制該最適任務度量的傳遞。

【0332】 - 度量：用於儲存附加至該相關集區根節點之排程錐之最適任務的度量。

【0333】 線程元件

【0334】 - 度量：用於傳遞直接關於該排程任務之適當性或一排程器可據以計算該適當性的資訊。

【0335】 排程行爲

【0336】 該排程操作被分爲兩種子類型：

【0337】 - 標準層級排程，其中一排程器層中的一或多個登錄競相成爲一層級中的最適登錄。

【0338】 - 集區分配排程－識別處理資源實體之一選擇之何者應被岔斷。

【0339】 沒有排程動作發生，除非接收一排程事件。

【0340】 排程推送及提取事件

【0341】 如前所述，系統狀態之改變可產生一“推送事件”或一“提取事件”－這些事件造成一重新排程發生。所有排程操作爲工作保留的。僅有可想像一特定事件所影響之排程階層式的部分被重新評估，此被稱爲存在於該重新排程範圍中。第 17 圖說明一基本排程階層式上之一事件的重新排程範圍。第 18 圖說明一簡單兩實體處理資源集區的重新排程範圍。

【0342】 層級排程

【0343】 層級排程爲該 SystemWeaver 排程演算法的最基本建立區塊。一排程實體可造成一串層級排程操作，如該使用者可配置排程階層式所定義。每個層級排程操作的結果爲該主排程器（一排程節點或一排程器根節點之一）HeadIndex 指標的更新。該主排程器之度量亦可依據該定義之度量傳遞演算法而加以更新。

【0344】 原則上，層級排程開始於該目前 HeadIndex 且在該排程器層之成員附近重複（雖然實際上爲了最小化延遲，推送操作僅對該目前標題指標更新該排程選擇），該排程器層依據下列建立對該 HeadIndex 是否需要更新：

【0345】 - 該事件，其可爲一推送或一提取操作

【0346】 - 與該事件類型有關之排程演算法（推送或提取）

【0347】 - 該層級成員之度量

【0348】 若發現一更適合登錄，則相應地更新該 HeadIndex。

● 優化該排程操作之行爲的數個特別情形被發現。在所有情形中靜止的排程器層在該排程操作中被忽略。

● 【0349】 任何時候每個排程節點必須知道存在於其子階層式之線程或任務描述項之數量以確保關鍵填滿參數被維護。然而，並不總是需要完全排程每一層一識別該即時下行（downstream）重新排程操作已造成一排程選擇更新的一旗標被維護—若確實需要，則該主層亦必須被完全評估；若不需要，則不再需要重新排程該剩下的上行（upstream）排程器階層式（即使需要特定其他狀態更新）。

【0350】 任何重新排程中的最後操作係用以判定最適準備任務是否應被允許先佔一特定 PRI 上的目前執行任務。該分派佇列描述項含有一排程演算法以及該目前執行中任務的度量—其可對該排程器根節點度量加以評估，該度量含有該排程錐之最適線程度量的一副本。

【0351】 集區分配排程

【0352】 集區分配排程僅發生於集區分配層中。由於基本層

排程尋求發現用於執行之最適線程/任務候選者，集區分配排程尋求發現用於先佔之最適處理資源實體候選者。典型地此係指識別以最不適合之資源集區執行該任務的處理資源實體，以及與該附加排程錐之最適‘準備’任務的度量相比較。

【0353】 雖然該最適準備任務相較於所有執行中任務係最低適合度，在每個附加處理資源實體上的剩餘分配錐被更新以確保每個排程層保持對全部可存取下行任務的瞭解，然而，不需要進一步的排程。

● 【0354】 雖然一先佔候選者被識別，該排程更新僅傳遞至該處理資源實體。

● 【0355】 第 20 圖說明由於第 18 圖所示之推送事件而發生之重新排程的一順序圖。一基本階層排程操作於一集區分配排程操作前發生於該集區根層（階層#6）。在此範例中，節點 5 被選擇為適於先佔的，因此在階層#1 中的階層排程操作被執行。該後續分派階層排程操作造成對該附加處理資源實體之一先佔。因此階層#2 亦被更新以確保其下行任務/線程之數目被維持。

【0356】 協同及先佔排程

【0357】 先佔排程使該目前執行任務可由一更適合（更高優先度）任務非同步地插入。先佔建立該執行處理資源及內容的特定需求，例如儲存狀態及一旦該先佔者已離開該資源時回復的能力。典型地可先佔任務或線程將透過該準備及執行狀態而維持相同的排程適合度。

【0358】 相反地，協同線程僅於一較高優先度任務必須等待

之完成建立。在該 SystemWeaver 任務管理解決方案中，協同線程於進入該執行狀態中最大化其適合度，因而排除一較高優先度任務以及該潛在後續先佔的存在。

【0359】 度量傳遞行為

【0360】 度量傳遞可被排程事件或該執行中任務或線程之度量的修改而造成。

【0361】 排程事件度量傳遞

【0362】 當對一排程層之主層的 HeadIndex 由於一排程事件而被更新時，該度量依據該主層中被定義之階層傳遞運算子而自該最適格子度量被傳遞至該主度量中。這些係依據該操作之性質以及從一單純副本乃至於一多重聚集的複雜範圍而定。

【0363】 執行線程度量傳遞

【0364】 目前執行線程之度量可被動態地修改—此可被用於避免封鎖資源上的優先度反向狀況。當該執行中處理資源實體並未參與一分配錐中時，僅更新該分派佇列描述項度量。在一分配集區的情形中，該執行度量被傳遞至與該處理資源實體相關的集區靜態節點（第 21 圖）。該 PSN 度量的更新由保持於該 PSNs 自身之中的度量傳遞運算子所控制。在特定排程方案中，靜態數值必須保持於集區靜態節點中。

【0365】 在兩種情形中啟動一重新排程事件以確保該新執行度量並未造成該執行及準備任務之比較性適合度的一改變。在該集區情形中，此僅為關於該新執行度量之排程器根節點度量的一重新排程。在該集區情形中，該集區分配層以及所

有後續階層必須被重新評估。

【0366】 閒置控制

【0367】 當一處理資源實體進入該閒置狀態時，其使用該執行度量以告知該排程結構。本質上，一閒置處理資源實體為“執行”該最低可能優先度任務者，且因而將被任何任務之到達先佔。該執行度量被設置為閒置數值以一般方式啟動一重新排程事件，因而造成該閒置任務由在該準備狀態結構中等待此處理資源實體之任務所“先佔”。

【0368】 對於與該“閒置任務”有關之一更詳細描述以及其對於處理資源集區中的電源管理的影響而言，參見下述集區方案章節中的區段電源管理。

【0369】 進階排程模式

【0370】 內部中可具有數個進階模式及行為或藉由採取特定 SystemWeaver 配置而達成。在以下章節中描述以下模式。

【0371】 注意此並非 SystemWeaver 中可用排程模式的一完全列表（exhaustive list）。

【0372】 時間片段（Timeslicing）

【0373】 雖然 SystemWeaver 系統原則上為事件驅動的，傳統計時器式系統如時間片段亦為可使用的。時間片段任務依據個別時間片段期間而分享一處理資源，其判定一任務可佔有一處理資源的間隔（假設在此間隔中沒有先佔任務成為準備的）。

【0374】 時間片段任務對正規任務（如第 1 圖所示）呈現一稍微修改的“進行（running）”行為。第 22 圖說明該進行狀

態的兩部分：正常及反對的（deprecated）。

【0375】 此章節描述時間片段任務的行為，伴隨在配置該任務時必須遵守的規則。

【0376】 SystemWeaver 核心資源

【0377】 以下章節討論用於在該 SystemWeave 伺服器核心中實施該時間片段特性的資源。

【0378】 計數器

【0379】 一前置處理資源實體計數器被用於該 SystemWeaver 核心中以協助時間片段行為。亦提供一單一前置計數器（per-scaler），其係由該系統計時器提供。該前置計數器之位元解析度被設置於晶片設計時間。

【0380】 時間片段狀態指標

【0381】 該前置處理資源實體岔斷狀態暫存器中的一狀態位元被撥出用於時間片段行為。此狀態位元暫存該時間片段計數器之逾期且可被軟體用以判斷一時間片段事件是否發生。

【0382】 配置

【0383】 一時間片段群組中的所有任務必須分享該相同優先度以及該相同主排程層，再者，時間片段任務不應與其他非時間片段任務分享排程層。該時間片段主層之排程演算法應被設為 FIFO。第 23 圖說明一典型時間片段配置，其中一時間片段群組在背景中與需要服務而取得優先度之一前景群組一事件驅動任務一同操作。

【0384】 行為

【0385】 當一時間片段任務首先開始執行時，一系統範圍之

時間片段數值被複製至與該處理資源實體有關之時間片段計數器中。該時間片段任務被稱為進入其‘正常’執行狀態（第 24 圖）。在該正常狀態中，每個週期降低此計數器。當到達零時，該任務之執行優先度（儲存於該分派佇列描述項中）被該硬體自動降低，而該任務進入該‘反對’狀態。此時該時間片段間隔計數器切換至該傳統監看模式。

【0386】 在一單處理資源實體提供數個時間片段任務的情形中，降低該執行優先度之動作（以及相關重新排程操作）將造成該準備狀態結構中該時間片段群組之另一成員的一先佔。藉由詢問該時間片段狀態位元，該軟體客戶端可判定該時間片段過程已逾期，並且推送如今被先佔之任務至該 FIFO 佇列之後方。因此該群組遵守該配置之時間片段規則同時大量地維護 SystemWeaver 核心排程及客戶端行為的正常操作模式。

【0387】 當一時間片段任務被一非時間片段任務先佔時，即使位於該‘正常’時間片段狀態中，該傑出時間片段過程被複製至該任務控制區塊中。該任務隨後被推回至該 FIFO 佇列之前頭的時間片段群組中。當任何先佔任務處理完成時，該時間片段任務係利用被重新設置該時間片段計數器之先佔任務的剩餘時間片段而加以回復。

【0388】 當一時間片段任務在該‘反對’狀態中被先佔時，其被推回至該時間片段群組 FIFO 佇列之尾端。在兩情形中該時間片段任務之優先度度量維持於其原始配置數值。

【0389】 當一時間片段群組係由處理資源實體之一集區所提

供（假設沒有先佔任務），進入該‘反對’狀態中的登錄不必然造成一立即切換至另一時間片段群組成員。注意以下資料：

【0390】 $T_p = (t*d)/p$ 其中 $(1 \leq p \leq t)$

【0391】 $T_{ready} = T_p - d$

【0392】 T_p 一時間片段群組之完全循環（執行每個成員任務一次）的期間

【0393】 T_{ready} 每週期中一特定任務等待於該穩定狀態中所經歷的時間量

【0394】 t 時間片段任務的數量

【0395】 p 在該集區中處理資源實體的數量

【0396】 d 每個時間片段間隔的經歷時間

【0397】 注意當 $p=t$ 時，若沒有任何其他先佔任務，該時間片段任務持續執行。

【0398】 執行描述檔

【0399】 以下執行描述檔說明 SystemWeaver 時間片段行為：

【0400】 第 25 圖說明具有依單一處理器以及三時間片段任務的傳統情形。其各自依據該時間片段間隔分享時間，直到依先佔任務到達該時間片段任務（在此情形中為編號 2）產生之位置。一旦該高優先度先佔任務完成時，該原始時間片段任務回復以完成該岔斷間隔。

【0401】 第 26 圖說明具有兩處理器之相同方案。一開始該三

時間片段任務被分享於兩處理器之間。當該先佔到達時，該時間片段任務分享該剩餘處理器，且當該先佔任務完成時，其回復在兩處理器上執行。

【0402】 第 28 圖說明當可用處理資源之數量等於時間片段任務之數量時，每個時間片段任務持續在該處理器之一者上執行。當一高優先度任務取得該處理器之一者的控制時，該時間片段群組依據該定義之時間片段間隔自動地分享該剩餘處理器。

● 【0403】 集區方案中的電源管理

【0404】 一處理資源集區的預設行為係由一分配層中的第一集區靜態節點評估一排程選擇。該集區根節點（PRN）具有一主指標器，其一般地指向一分配層中的第一級區靜態節點（PSN）（第 28 圖）。當評估用於先佔之一候選者時，該比較從此登錄開始並進行於使用該對等指標的列表周圍。

● 【0405】 若所有靜態節點的適合度相同，且該適合度低於該排程錐之候選者時，將選擇被遭遇之第一節點。因此在低負載方案中，當一或多個處理器為閒置時（已設置其度量為該閒置數值），最靠近該 PRN 之主指標器之處理器資源實體將被偏好於執行新任務，而最遠離該主指標之處理資源將呈現長閒置期間。

【0406】 不同的省電方法於該處理資源必須被重新喚醒時傾向呈現不同的衝擊，例如時脈閘控可維持一微結構中的所有狀態，然而積極電壓/頻率調整於重新喚醒時可犧牲所有既有狀態且呈現不佳的突波電流。在一特定 PRI 具有許多不同損

失之降低電源選擇之情形中，藉由該之排程行為，依據該時間消耗閒置而管理其使用是有道理的。

【0407】 可依據該處理資源實體以及系統巨結構之能力而將該‘閒置’狀態分為多個子狀態。可能從某狀態重新開始相較於其他狀態為更損失的（例如一狀態保持時脈閘控降低電源相較於一電源隔離狀態）。為了支援這些方案 SysterWeaver 支援多個閒置優先度。

【0408】 對於具有多個子狀態之該些處理資源而言，岔斷回應於線程執行回復時可被穩定地轉換回該閒置狀態中。此允許漸進引入一處理資源至一特定分配之動態集合中。

【0409】 範例

【0410】 第 29 圖使用升冪優先度說明當時間通過一閒置狀態中時一處理資源實體（PRI）的“執行優先度”。在該第一實體中，該閒置任務設置該優先度為其最低可能設定，其給予該 PRI 該排程器之一任務分配的最高可能機會—相較於其分配集區同儕。

【0411】 之後該閒置任務引用一電源降低模式，其或許被支援於該處理資源微結構中。在此處該閒置任務增加該 PRI 之執行優先度以降低一任務分配的可能性（在先前狀態中的 PRIs 將取得優先）。

【0412】 類似地在另一期間之後，該閒置任務（或某些其他代理程式）進一步提升該執行優先度（或許隔離對該 PRI 之電源—因而避免靜態洩漏）。該優先度之調變使一任務分配之 PRI 適合度進一步降低—符合重新喚醒此處理資源之損失

（在此情形中為突波電流、冷快取效應（cold cache effect）等等）。

【0413】 注意：在一分配層中該第一登錄之偏好先佔有時為不佳的。雖然有此種情形，可於該先前排程選擇成為後續排程操作之新起點時選擇一不同行為。此選擇在一分配集區中的處理資源實體之間呈現高優先度先佔任務的一更公平分配。

【0414】 滯後式排程

【0415】 在某些情形中希望維持一排程選擇，不論其進行的絕對正確性。一般而言此情形發生於對一特定類型之任務或資料組建立一內容的損失很高因而應盡可能聚集多個任務時。此情形之範例包括：

【0416】 - 處理器快取一已充滿一過去演算法或資料組的一快取記憶體對於一不同演算法及/或資料組將展現低的吸引力。此被稱為一冷快取效應且呈現一高的快取遺失比例並導致效能低落。

【0417】 - 可重新設置 FPGA 分割區一部分執行時重新設置性於該晶片被配置且可操作時使該 FPGA 之一部分可被動態地重新設置，故賦予不同演算法在整體時間之執行。

【0418】 然而，自一演算法切換至另一者的損失很高而必須聚集更大資料組以確保系統效率。

【0419】 此兩者為高損失內容切換的範例。

【0420】 滯後式排程藉由聚集多個使用者操作之內容切換的損失而可被用於避免某些不良效果。藉由使用該度量之其中

一者而表示一‘系統損失’參數將可使用滯後式排程。該滯後度量可依據該系統中損失的各種測量：

【0421】 - 任務記憶體佔據：其中記憶體位於一優良狀態，一特定任務佇列的累積足跡將被用於判定何時排程該新配置。

【0422】 - 處理需求：其中聚集一實質“主動處理”期間之一內容的損失為較佳的。

【0423】 - 時間片段：其中延遲上的時基誤差（jitter）為重要的。

【0424】 舉例來說，在一動態重新設置 FPGA 之情形中，一記憶體可被累積在可配置結構之一單一部份上多工處理之各個演算法內容的工作—在此情形中該記憶體佔據可為選擇何時再規劃該陣列之一因素。在所有情形中均可能設計一排程器階層式為允許由於高優先度任務之到達的強制切換。

【0425】 該關鍵系統階層挑戰被歸納如下：

【0426】 - 一內容切換之損失的衝擊（切換時間、突波電流）

【0427】 - 切換的時機

【0428】 - 如何管理一非主動特定內容之工作的累積。

【0429】 下文描述使用 SystemWeaver 管理動態 FPGA 在配置之可能方式。可使用相似排程計數以及較單純軟體客戶端填隙行為而達成冷快取之管理。

【0430】 FPGA 執行期重新設置

【0431】 雖然十分類似於處理器內容切換，但需要數個（再）定義：

【0432】 配置——組編寫變數之一，其可被針對該 FPGA 結構的一特定部分。

【0433】 內容切換——改變該 FPGA 之一可重新設置部分之配置的動作。

【0434】 任務——特定 FPGA 配置所執行之工作的一單一單元。

【0435】 在此提案中針對該 FPGA 之可重新設置部分的配置被視為協同的（與先佔的相反），亦即該個別任務為不可分割的，而必須於一內容切換可發生前完成。此確保任務無須再進入並限制內容之間的狀態保存問題為等待一特定配置之任務數量必須為一整數值。該傑出任務之管理的一邏輯圖被顯示於第 30 圖中。

【0436】 任務被組織於佇列之中。這些佇列永遠存在；特言之，其累積目前並未作用中之 FPGA 配置的工作。該排程器判斷何時切換任務並管理一任務群組中任務的執行。該重新設置支援邏輯管理再規劃該結構之機制並於完成時傳送信號。依據此模型之協同性質，當一內容切換被排程時並無需要保留於該結構中的資料。

【0437】 排程器

【0438】 該排程器執行兩種明確功能：

【0439】 - 其依據該任務佇列之改變狀態持續評估目前的排程選擇。

【0440】 - 其管理一個別任務佇列中的任務執行順序。

【0441】 每個任務到達造成該排程選擇的一更新以確保該

FPGA 結構在任何時間係位於正確狀態中（貪婪排程）。在該任務佇列中該排程器依據該系統結構所定義之特性而命令執行之順序。該排程器至少應提供 FIFO、循環排序及優先度策略。

【0442】 使用 SystemWeaver 之重新設置管理

【0443】 該 SystemWeaver 解決方案提供排程及內處理器通訊能力的一傑出組合，其可被用於管理執行時平行執行及內處理通信。SystemWeaver 之特徵可於傳統指令組結構、固定硬體元件即可重新設置 FPGA 區塊等等中有效地管理任務及內容切換。第 31 圖說明一示範結構。

【0444】 SystemWeaver 處理 FPGA 結構內容切換之排程的管理以及該個別任務佇列中的任務排序。一般而言，此係附加於排程更多典型任務至該平台中的固定配置元件。

【0445】 該重新設置本身係利用一特定 SystemWeaver 硬體客戶端填隙加以處理。注意用以管理快取“暖度”之類似排程技術並未對該標準客戶端填隙施加額外需求。該客戶端填隙所接收之每個排程任務控制區塊係與該結構之既有配置相比較。若該目前載入之配置以及該排程任務之配置不同，則該客戶端填隙重新設置該結構，而未進一步與該 SystemWeaver 核心互動。隨後該結構更新選擇依據該排程器之支配而單獨地由該任務之輸出順序所控制，該客戶端填隙可被重新設計以容納不相似的重新設置策略。

【0446】 排程策略

【0447】 該排程策略應由該系統設計者所決定。然而仍有必

須可被用於支援此特徵的關鍵能力。特言之，排程器演算法應可呈現滯後—亦即停留在一排程選擇上，直到一足夠損失已於他處聚集以警告切換至一替代選擇。

【0448】 在顯示之範例中，每個“任務具有一隨機建立之度量，其被加入至表示該任務群組織聚集度量中。當一特定任務群組被提供自該佇列移除一任務之“提取”操作時，則降低此聚集計數。

【0449】 當一“推送”（一新任務到達）或一“提取”操作發生時，該結構排程器對每個候選者評估該目前執行之任務群組的度量。依據第 32 圖中的演算法：

【0450】 需要的更新 = $(C_{\text{candidate}} > C_{\text{current}} + \text{Hysteresis})$

【0451】 $C_{\text{candidate}}$ 在該候選者排程層中的累積任務損失

【0452】 C_{current} 在該目前選擇之排程層中的累積傑出任務損失

【0453】 Hysteresis 被增加以避免內容猛移（thrashing）的滯後

【0454】 第 33 圖說明可被選擇以實施以上描述之演算法的一排程階層式。在此情形中該“滯後排程器”之度量 1 儲存該滯後運算域。度量 0 可被用於儲存該滯後群組之靜態優先度，此優先度被使用於在該滯後群組及該先佔群組之間排程時。該假設為有某些任務，其具有足夠的優先度以強迫一內容切換。

【0455】 結果

【0456】 第 32 圖說明此一系統之概念效果。該任務排程輸出

一般而言被謹慎地封鎖，其於管理系統流量設計之衝擊時取得任何特定配置之最大使用。

【0457】 第 34 圖說明該呈現之排程演算法的一模擬。等待四配置之各者的累積任務“損失”係伴隨所有損失之整體總和（“累積的”）而加以繪製。該選擇線指示該演算法將選擇之可用配置。

【0458】 混合排程演算法範例

【0459】 混合排程運算子為有用的，例如在集區分配層中排程處理資源集區時。舉例來說，或許該成員之一子集合僅應於等待處理之任務的佇列填入超過一特定臨界值時為適合的。

【0460】 考慮可使用三處理器資源的情形：一 RISC 處理器以及兩數位信號處理器（DSPs）（第 35 圖）。這些資源之每一者理論上可執行一語音編碼操作，但該 DSP 裝置為更有效的。在此情形中，該 RISC 處理器如圖所示可被呈現於該語音編碼集區中，但其參與於該執行中的適合度將視等待該功能之任務佇列的深度而定。

【0461】 在此一配置中，該集區根節點度量可表示優先度，而度量 1 可表示佇列填入。在該候選者 PSNs 之各者中，度量 0 一般將表示在其個別處理資源實體（PRIs）上執行之任務的執行優先度。在此情形中，度量 1 將表示使其相關 PRI 適於排程所需的佇列填入。在此情形中該混合排程演算法為：

【0462】 所需更新 = (Mcurrent0 > Mcandidate0) &&
(Mcurrent1 > Mcandidate1)

【0463】 在與該 DSP 裝置相關之 PSNs 中，M1 將被設為零，因而該演算法僅依據優先度而加以判定。在該 RISC 處理器之情形中，M1 將不為零，因而 Mcurrent1 所表示之佇列填入必須增加以超過此數值以供該 RISC 處理器參與該演算法之執行。

【0464】 下文描述一巨結構或處理層級之 SystemWeaver 伺服器。

【0465】 如前所述，該 SystemWeaver 硬體解決方案具有四種元件：

【0466】 - 該 SystemWeaver 伺服器核心

【0467】 - 該 SystemWeaver 密接記憶體

【0468】 - 該 SystemWeaver 除錯管理員

【0469】 - 該 SystemWeaver 客戶端填隙

【0470】 整體度量代理程式為選擇性的，且於該系統設計需要包括系統整體狀態於該排程選擇中時加以使用。

【0471】 主要連接群組

【0472】 第 38 圖說明在該核心周圍上發現的介面群組。

【0473】 為了確保該 SystemWeaver 核心可被簡單地整合，所有信號為同方向的且與一單一時脈同步。下文說明這些群組之構成成員。所有的信號方向係與該 SystemWeaver 核心有關。

【0474】 系統控制群組

【0475】 該系統控制群組含有用於確保該 SystemWeaver 核心之正確操作的各種信號。這些包括系統時脈、即時時脈及重置信號。

【0476】 整體度量群組

【0477】 在某些系統中希望在排程選擇過程中使用特定系統度量。這些度量可表示各種因素，例如互連繁忙度、快取選中（hit）比例、記憶體佔用等等。

【0478】 周邊岔斷群組

【0479】 該周邊岔斷群組係由該 SystemWeaver 控制系統外部所提供之一組岔斷所組成。在該周邊岔斷群組中的信號可由例如連接外部世界之輸入介面所驅動，或直接由該 SoC 裝置之外部透過針腳所驅動。周邊岔斷輸入之數量係定義於 SoC 設計時。

【0480】 內部岔斷群組

【0481】 該內部群組係由該 SystemWeaver 系統所提供之兩同步岔斷群組以及執行時系統除錯信號之一單一群組所組成。一單一群組中每個信號的編號將典型地對應該系統中的處理資源的編號且將被定義於 SoC 設計時。

【0482】 密接記憶體介面群組

【0483】 此群組連接 SystemWeaver 至其私有之密接記憶體資源。假設該附加記憶體為一同步 SRAM 裝置。該位址路徑之寬度 n 以及該資料路徑之寬度 m 被定義於 SoC 設計時。

【0484】 互連群組

【0485】 該個別互連策略一包括協定及層數一必須被設置於 SoC 設計時。任何特定匯流排介面信號之細節可參見該對應之特定匯流排實施。

【0486】 除錯介面群組

【0487】 關於該除錯管理員之介面的細節，請參見共同申請中的國際 PCT 申請書第 PCT/GB2005/003525 號，其藉由參照而被納入於本文中。

【0488】 密接記憶體 (TCM)

【0489】 SystemWeaver TCM 為各種 EDA 廠商所提供之一標準編譯程式 SSRAM 技術。依據本申請書之需求，該 TCM 含有於 SoC 設計時所定義之 SystemWeaver 記憶體元件(WMEs)之一整數數量。每個 WME 使用 256 位元的記憶體空間。SystemWeaver 最大支援 65536 個 WMEs，或一 16Mb 記憶體。

【0490】 雖然佇列描述項確實使用 WMEs，但在一典型系統中需要之 WMEs 數量將由線程支援需求所決定。舉例來說，可在該 SystemWeaver 伺服器中同時支援 400 線程的一系統將需要大約 128kb 的附加記憶體。

【0491】 在 SoC 設計時，可修改該記憶體介面以簡化路徑。

【0492】 伺服器核心子區塊描述

【0493】 第 39 圖說明。該 SystemWeaver 伺服器實體之主要邏輯元件這些功能被映射至第 40 圖中說明之結構上。該功能被劃分於四個主要內部平行處理元件之間，其執行以下功能：

【0494】 - 線程排程器輸入管理員 (TSIM)：空閒列表維護、WME 恢復。

【0495】 - 線程排程器等待管理員 (TSPM)：等待列表維護、同步、提升至準備佇列結構。該線程同步管理員維護該等待佇列結構之完整性（插入及取出）。

【0496】 - 線程排程器輸出管理員 (TSOM)：準備佇列維

護、分派佇列維護、處理資源電源管理、岔斷建立。維護該準備佇列結構之完整性（插入及取出）。

【0497】 - 線程排程器排程管理員（TSSM）：維護該準備佇列結構中每個處理資源之排程選擇。

【0498】 額外的數個區塊提供支援功能：

【0499】 - 線程排程器記憶體管理員（TSM）：聚集對該附加 SystemWeaver 記憶體之存取，其包括雙方獨佔及鎖定。

【0500】 - 線程排程器岔斷管理員（TSIC）：轉換進入系統岔斷為國際同步基本指令。

【0501】 - 線程排程器介面管理員（TSIF）：提供與 SystemWeaver 資源之互連連接、配置以及執行時存取。

【0502】 第 40 圖說明該 SysmtemWeaver 結構之主要子區塊。以下章節詳述這些元件之間的子區塊內互動。每個子區塊呈現一組“公開方法”至其他子區塊，使各個子區塊得以指示其同儕在其個別維護之結構上執行操作。

【0503】 可利用特定條件完成一命令時，狀態旗標被管理於該子區塊之中。

【0504】 在子區塊介面圖上的箭頭方向指示該匯流排之主從性（master-ship）且並非反映該信號群組之個別元件的方向。

【0505】 狀態描述項行為

【0506】 在 SystemWeaver 之操作過程中使用各種描述項類型（關於進一步細節可參見共同申請中的國際 PCT 申請書第 PCT/GB2005/001154 號，其藉由參照而納入本文中）。這些描述項之多數為無狀態的，然而，線程描述項及排程器描述

項在特定情況下可轉換為各種狀態。此文件描述這些狀態轉換以及造成轉換之事件。

【0507】 線程描述項

【0508】 有兩種該 SystemWeaver 內部認識之兩種線程描述項：該標準線程描述項以及該標記線程描述項。後者被專門用於同步移除排程階層式之處理，同時確保先前排程之線程描述項的完整性及順序。

【0509】 第 41 圖說明兩線程描述項及標記線程描述項經過的內線程狀態圖。注意狀態新及空閒為中介狀態，其與 SystemWeaver 中的一永久狀態無直接關聯。一字面狀態變數不存在於該線程描述項中，除非該狀態係由數個旗標所表示。第 3 表說明旗標狀態及第 41 圖中該線程存在之狀態之間的關聯。

【0510】 標準線程描述項狀態描述

【0511】 下文簡要說明造成進入以及自其離開的事件及狀態。

【0512】 新的

【0513】 該新的狀態為暫時的。新的線程由一推送獨立或依賴線程命令而被導入至該 TSIF。該兩種情形被處理如下：

【0514】 - 獨立線程（不具有時間或同步依賴性的線程）立即轉換為該推送狀態。在此情形中該 TSIF 指示該 TSSM 導入該線程至該準備佇列結構之中。

【0515】 - 依賴線程（具有一時間或一同步依賴性的線程）轉換為該封鎖狀態。該 TSIF 指示該 TSPM 適當地導入該線程

至該等待及時機佇列結構中。

【0516】 封鎖的

【0517】 在該封鎖狀態中，線程描述項等待一外部同步及/或一時機式同步之一。封鎖的線程起始於該 TSIF。當接收該該適當同步時，該 TSSM 轉換該線程至該推送狀態中並指示該 TSSM 導入該線程至該準備佇列結構之中。

【0518】 推送的

【0519】 在該推送狀態中的線程已被同步或原始為獨立的，該 TSPM（依賴線程）及 TSIF（獨立線程）在這些個別情形中將管理轉換為該推送狀態。該 TSSM 推送該線程至該準備佇列結構中並轉換該線程為該準備狀態。轉換為該準備狀態造成一再排程發生。

【0520】 準備的

【0521】 在該準備狀態中的線程已自該推送狀態中轉換或已被清回該準備佇列結構中（由該 TSSM）。該轉換為該準備狀態總是促使一再排程。線程可從該準備狀態中移出而進入該提取狀態或該清理狀態（flushed state），後者為越過該提取狀態以於一單一操作中清理之一特定情況。線程於被提名為該 TSSM 所排程之最適候選者時轉換為該提取狀態。

【0522】 提取的

【0523】 線程在該提取狀態中已被該排程器（TSSM）提名為一特定處理資源實體或一群實體所處理之最適線程，其由該 TSOM 轉換為此狀態。線程可由該提取狀態轉換為該清理狀態或該停滯狀態（zombie state）：

【0524】 - 該 TSOM 由於一再排程識別一更適合線程而轉換線程為該清理狀態。

【0525】 - 該 TSOM 由於在該系統處理資源實體之一者中的開始處理而轉換該線程為該停滯狀態。該停滯狀態維持該線程描述項直到其可為空閒的。

【0526】 停滯的

【0527】 停滯線程在該 TSSM 中被處理。該停滯狀態之存在係用於確保對一特定線程描述項之所有依賴在該線程為空閒之前已被除去。此可由該線程一旦已到達該 TSSM 處理佇列之前端時所保證，因而不需要進一步的處理。

【0528】 清理的

【0529】 清理線程係由該 TSSM 所處理。清理線程必須被重新導入至該準備佇列結構中，其造成一再排程操作。一旦此被完成，該 TSSM 將該線程轉換回該該準備狀態中。

【0530】 空閒的

【0531】 該空閒狀態為為一暫時狀態，其指出該線程描述項所使用之 WME 被放回該空閒列表中。

【0532】 下文簡要說明造成進入及自其離開的事件及狀態

【0533】 新的

【0534】 該新狀態為暫時的，新標記線程由一推送標記線程命令導入至該 TSIF。標記線程總是越過該封鎖狀態，以確保存在於該 TSPM 之輸入及輸出工作佇列中的任何處理—其可影響標記線程被最終刪除之排程器層的狀態—已於該標記線程本身之到達之前被完成。

【0535】 封鎖的

【0536】 該 TSPM 立即轉換標記線程為該推送狀態並指示該 TSSM 導入該線程於該準備佇列結構之中。

【0537】 推送的

【0538】 該 TSSM 推送標記線程至該準備佇列結構中並轉換該線程為該準備狀態。轉換為該準備狀態造成一再排程發生。

【0539】 推送的

【0540】 該 TSSM 推送標記線程至該準備佇列結構中並轉換該線程為該準備狀態。轉換為該準備狀態造成一再排程發生。

【0541】 準備的

【0542】 當標記線程被導入至該準備佇列結構中時，其解開（unlock）其緊接的主層。該主排程層之釋放隨後由依賴線程之數量的計數所控制。在其主層之依賴線程計數已到達零時—亦即在 SystemWeaver 中以不再有依賴該主排程層之存在的線程描述項，標記線程僅適於轉換離開該準備狀態。

【0543】 與標準線程類似，標記線程可轉換離開該準備狀態進入該提取狀態或該清理狀態其中一者，其後者為跨過該提取狀態以於一單一操作中清理的一特定情況。標記線程於其已被提名為該 TSSM 所排程之最適候選者時轉換為該提取狀態。

【0544】 提取的

【0545】 標記線程在該提取狀態中已被該排程器（TSSM）提名為一特定處理資源實體或一組實體所處理之最適線程，注意此排程選擇為一特定情形，其指示該排程層為空的且不再

具有依賴線程。標記線程可從該提取狀態轉換為該清理狀態或該停滯狀態：

【0546】 - 標記線程由於一再排程之結果識別一更適合線程而被轉換為該清理狀態。

【0547】 - 標記線程由於在該系統處理資源實體之一者中開始處理而被轉換為該停滯狀態。

【0548】 停滯的

【0549】 該停滯標記線程之處理與正規線程描述項相似。該標記線程之主排程層亦將於此狀態中被刪除。

【0550】 空閒的

【0551】 該空閒狀態為指示該線程描述項所使用之 WME 被置回於該空閒列表上的一暫時狀態。

【0552】 排程器層狀態圖

【0553】 排程器層亦具有內含狀態。對於持續於該系統之執行時間的一靜態層而言，該唯一狀態為主動的。該剩餘狀態係由動態排程層所使用，亦即該些在執行時來來去去者。第 4 表提出旗標狀態以及第 42 圖中該排程器層存在之狀態之間的關聯。

【0554】 排程器層描述項狀態描述

【0555】 下文簡要說明造成進入以及自其離開的事件及狀態。

【0556】 新的

【0557】 該新狀態為暫時的。新排程器層由一推送獨立元件在起始過程中或在執行時導入至該 TSIF。

【0558】 靜止的

【0559】 在該靜止狀態中，該排程器層被允許聚集線程及潛在額外子階層式，但將永不為該 TSSM 所排程。可以兩種方式進入該靜止狀態：

【0560】 - 一新排程描述項可被建立於該靜止狀態中。

【0561】 - 一排程層可於執行時被修改並透過經由該 TSIF 送出之專屬系統命令而被置於該靜止狀態中。

【0562】 該靜止狀態僅可由一專屬系統命令離開。

【0563】 主動的

【0564】 在主動狀態的排程器層主動地參與排程且被鎖定，意指當其變成空的時不會被移除。靜態排程器將典型地被建立於此狀態中。動態排程器藉由透過該 TSIF 接收之專屬系統命令而轉換為此狀態。該排程器層僅於接收一標記線程時離開該主動狀態，其中其進入該“等待中空閒”狀態。

【0565】 等待中空閒

【0566】 排程器層保持於該等待中空閒狀態中直到以下兩條件均被符合：

【0567】 - 依賴元件之數目—其為保持描述項之數量之一測量並加以參照的一計數—成為零時。

【0568】 - 子元件之數量成為零時。

【0569】 空閒的

【0570】 該空閒狀態為指示該排程層描述項所使用之 WME 被置回該空閒列表的一暫時狀態。

【0571】 內子區塊行為

【0572】 下文涵蓋涉及多個子區塊之 SystemWeaver 行爲的太樣。

【0573】 動態排程階層式操作

【0574】 在系統執行時可增加及移除排程階層式。若特定使用者層級程序爲受尊重的，則該 SystemWeaver 保證該系統之完整性以及自排程階層式之一部分轉換至另一部份之線程描述項的順序。關於進一步細節可參見該。

【0575】 動態排程度量更新

【0576】 在一標準或標記線程中可更新該度量。行爲係依賴線程狀態：

【0577】 - 若該線程位於封鎖狀態中，則一適當命令被傳送至該 TSPM。由於該等待中佇列被排序，則該線程於該度量被更新時被移除並再插入至該佇列中以確保其再出現於該適當位置。

【0578】 - 若該線程位於任何其他永久狀態中，一命令被送至該 TSSM，其執行該度量更新並再排程該排程階層式之適當部分。

【0579】 用於排程之 TSOM/TSSM 互動

【0580】 該 TSOM 及該 TSSM 在排程線程描述項至處理資源實體中參與一部份。第 72 及 73 圖說明該 TSOM 及該 TSSM 之間的互動的某示範順序圖。

【0581】 內子區塊結構及行爲

【0582】 每個子區塊係以其主要 IO、對其同儕及外部世界之實體介面之形式加以討論，或者以其命令介面或可使用該命

令協定在該適當實體介面上呼叫之方法的形式加以討論。

【0583】 TSIF-介面管理員

【0584】 該介面管理員負責指揮自該互連群組接收之命令的執行以及將其分配至該其他子區塊。

【0585】 下文描述存在於該 TSIF 中的功能實體，且其不僅是命令之翻譯以供內部使用。

【0586】 結構

【0587】 該 TSIF 主要將在該互連介面上接收之命令解譯為該剩餘子區塊的一或多個內部命令。下文詳細描述存在於該 TSIF 中的結構資源。

【0588】 號誌（semaphore）區鎖定

【0589】 號誌區鎖定（SRL）提供一資源，其可被任何系統資源細微地測試及鎖定以取得對一系統資源的專屬存取。

【0590】 其可由於任何數量之原因而加以使用：

【0591】 - 爲了鎖定含有一或多個共享資源（例如號誌物件、任務控制物件等等）之系統記憶體之一區域，因而保證完整性。

【0592】 - 爲了用於一多循環命令存取而鎖定該 SystemWeaver 命令介面。

【0593】 - 爲了用於多週期事件而鎖定該 SystemWeaver 除錯事件介面。

【0594】 SRLs 具有兩狀態：鎖定的及未鎖定的。自 SRLs 讀取被定義爲嘗試取得鎖定，而寫入 SRLs 被定義爲嘗試解鎖。在鎖定一特定 SRL 之處理資源實體以及將其釋放者之間不需

要關聯。以下描述該行為：

【0595】 - 未鎖定的。在該未鎖定狀態中，自一 SRL 之一讀取傳回一控制碼，其對該讀取者指示該鎖定嘗試是否成功。在此狀態中寫入不具有影響。

【0596】 - 鎖定的：在該鎖定狀態中自該 SRL 之一讀取指示該 SRL 為不可用的。一寫入釋放該 SRL。

【0597】 命令處理

【0598】 該 TSIF 破壞自該系統接收之命令為可能多個內部命令。

【0599】 監看器及時間片段支援

【0600】 一雙模計時器計數器被選擇性地提供給每個處理資源實體。該兩計時器模式為監看器及時間片段，該預設模式為監看器。

【0601】 監看性行為係由一監看器週期及該個別計數所定義（第 44 圖）。監看器週期為該系統計時器之一降級（stepped down）版本，該降級係由一系統常數於矽設計時所定義。第 46(a)圖說明每個處理資源實體於每個預定監看器時脈週期上的行為。

【0602】 - 詢問該個別監看器計數器以判定一監看器岔斷是否適合。

【0603】 - 於需要時建立一監看器岔斷。

【0604】 該監看器計時器計數於自該計時器相關之處理資源實體的每個控制存取上被重設，因而僅有不在該監看器間隔中存取 SystemWeaver 的處理資源經歷一監看器岔斷。

【0605】 該之時間片段行為於該 TSIF 及該 TSOM 之間被分割。在該 TSIF 中，分配至每個處理資源之計時器可被專屬於時間片段支援。當一時間片段間隔發生時，自動度量操作被提供於該 TSOM 中（該執行優先度被降低）。該處理資源本身僅於一標準先佔由於該自動度量更新而變成適當時被更新。

【0606】 第 45 圖說明該 TSIF 中該時間片段支援邏輯的每個系統時脈週期的行為。第 46(b)圖說明每個處理資源於該預定時間片段時脈之每個週期上的行為。注意在一時間片段事件已發生時，該計時器模式被回復為監看器模式。

【0607】 岔斷處理

【0608】 SystemWeaver 自接受源自該系統之岔斷、周邊岔以及對該系統之來源岔斷、處理資源實體岔斷。

【0609】 周邊岔斷

【0610】 周邊岔斷可被遮蔽及限制於該 TSIF 中（邊緣/層級觸發、負/正邏輯等等）。

【0611】 處理資源實體岔斷

【0612】 岔斷處理資源被提供於該 TSIF 中以提供以下能力：

【0613】 - 維持岔斷狀態，包括一岔斷斷言的來源（先佔、時間片段、監看器）。

【0614】 - 遮蔽能力。

【0615】 一特別特徵被實施於該 TSIF 中以自動化岔斷遮蔽於該 SystemWeaver 命令介面上。由於該命令介面為該系統中所有處理資源的一共用存取點，故完整性及效率必須被維

持。對於此目的而言，SystemWeaver 於一處理資源在該 SystemWeaver 命令介面上取得一鎖定時自動處理岔斷遮蔽以確保該處理資源在此關鍵編碼區段不被岔斷。

【0616】 一計數器被維持給每個處理資源實體，其追蹤已被接收之成功 SystemWeaver 命令介面號誌區鎖定請求的數量。每個鎖定請求增加該計數而每個解鎖降低之。當該計數器從零被增加時，岔斷被自動遮蔽；而當該計數被降低至零時，岔斷被自動解除遮蔽。

● 【0617】 TSIM—輸入管理員

【0618】 該輸入管理員管理該 WME 空閒列表，其處理該 TSIF 之提取請求以及各種子區塊(TSIF、TSPM、TSOM、TSSM) 之推送請求。

● 【0619】 結構

【0620】 該 TSM 僅含有一結構實體—該 SystemWeaver 記憶體元件 (WME) 空閒列表。第 48 圖說明該列表之結構。

● 【0621】 該空閒列表依據一後進先出 (LIFO) 策略操作。該列表之每個成員為 C_SCHED_ENTRY_FREE 類型且使用 C_FREE_QUEUE_POINTER_INDEX 所參照之指標而被單獨連結。

【0622】 方法介面

【0623】 除了連結列表內勤方法外 (取得及設置狀態) ，該輸入管理員提出以下命令於其介面上：

【0624】 推送空閒索引 (C_TSIM_CMD_PUSH_INDEX)

【0625】 呼叫端: (TSIF, TSSM, TSOM, TSPM)

【0626】 該推送空閒索引命令被用於推送一自由 WME 索引回到該空閒列表上。其參數被歸納如下：

未簽署短索引	新釋放之 SystemWeaver 記憶體元件之索引
--------	----------------------------

【0627】 提取空閒索引 (C_TSIM_CMD_POP_INDEX)

【0628】 呼叫端：TSIF

【0629】 該提取空閒索引命令備用於從該空閒列表提取一空閒 WME 索引。

【0630】 其參數被歸納如下：

未簽署短*p 索引	對於置放新分配之 WME 節點 ID 所在之位置的參照
-----------	-----------------------------

【0631】 TSPM—等待管理員

【0632】 該等待管理員管理被封鎖之任務或線程描述項等待某事件：同步或依據計時器。個別等待列表關聯係由使用者控制（或由更高層軟體所控制），其可表示一號誌、一爭用領域（contention domain）、一岔斷或這些的任何組合。

【0633】 結構

【0634】 該等待管理員包含兩主要元件：一可變數量的等待佇列以及一計時器佇列。該等待佇列儲存等待由某外部事件所同步之線程的列表，而該計時器佇列儲存等待一逾時之線程的一列表。一線程描述項可能且事實上經常為該兩列表之一成員，藉此一線程描述項被允許在一受限長度時間中等待一外部同步事件。

【0635】 等待佇列結構

【0636】 第 50 圖之等待佇列結構主要被實施於該密接記憶

體中，伴隨該 TSPM 中的資源及功能以處理其內容。該 TSPM 本身含有一標題指標以及參照等待佇列描述項之一列表的數個元件。每個等待佇列含有線程描述項之一列表，該等待佇列之數量於執行中可動態地增加及減少。該封鎖狀態中的所有線程描述項存在於一等待佇列中（與該計時器佇列相反，線程於此佇列中僅於其具有一定義逾時時出現）。多個等待佇列之使用係依據該應用程式及該應用程式設計者的需要及偏好。除此之外，等待佇列可與下列相關：

● **【0637】** - 一號誌。此可能發生於大量的等待佇列各自含有少數線程。

【0638】 - 一爭用領域（contention domain）。一爭用領域為多個實體競爭相同資源所位於之區域。舉例來說，一處理（與一線程相對）可被視為一爭用領域。

【0639】 - 岔斷。對於最快反應時間而言，岔斷將典型地被群聚為一專屬等待佇列。

● **【0640】** 不像計時器佇列，該等待佇列僅依據事件。這些事件為：

【0641】 - 一推送事件，其中一新線程被導入至已存在或必須被建立之一等待佇列中。

【0642】 - 一同步事件，其中一或多個線程必須轉換為該準備佇列結構。

【0643】 下文描述這些情形下的行為：

【0644】 推送事件

【0645】 在一推送事件上該 TSPM 必須盼訂在該列表之何處

插入該線程描述項（依據一識別順序運算子度量[0]）。有兩種必須被考慮的情形：

【0646】 - 推送至一既有等待佇列

【0647】 - 推送至一新等待佇列

【0648】 前者情形為平凡的，該列表被依序詢問直到發現一插入點。在升冪優先度之傳統情形中，該排序運算子被設為‘大於’（C_PEND_MNGR_PUSH_GTR）並搜尋該既有列表。該插入點係於該新線程之度量[0]大於該次列表成員度量[0]被定義。

【0649】 當需要一新等待佇列時，該等待佇列插入立即於該新線程登錄之插入前發生。

【0650】 同步事件

【0651】 可從該岔斷控制器（TSIC）或透過該命令介面接收同步化事件。同步化可發生於兩種模式中：

【0652】 - 一字面模式，其中該命令引數建立對一 WME 索引的一字面參照。

【0653】 - 一關聯模式，其中在該線程描述項中的一欄位以及該同步化基本指令中的一欄位之間尋找一關聯。

【0654】 在該字面模式中，由於該線程描述項之索引被傳入該指令中，且不需要尋找最適候選者。該關聯模式需要該最適同步化候選者被首先發現。

【0655】 該關聯模式含有三種子類型：

【0656】 - 正規，其中僅有該特定等待佇列中的最適候選者被同步。

【0657】 - 多播，其中該特定候選佇列中的所有適合候選者被同步。

【0658】 - 廣播，其中所有等待佇列中的所有適合候選者被同步。

【0659】 該最適候選者係由伴隨該命令所傳送之一引數以及亦位於該命令之一元件識別記號所識別。該元件識別記號指定該候選者線程描述項欄位之何者與該傳送引數相比較以識別適合度。在該正規模式中，該演算法繼續重複於該等待佇列中直到發現一適合候選者，於該處將其自該等待列表以及於可用時自該計時器佇列中移出，並被轉送至該準備佇列結構。對於多播及廣播模式而言，此處理持續直到該等待佇列或每個等待佇列分別變成空的。

【0660】 特別情形係與該計時器佇列之成員的移除有關。參見下文之詳細描述。

【0661】 時機佇列結構及操作

【0662】 每個被導入至該逾時佇列的新線程初始含有一絕對逾時數值。此數值可由該介面管理員自以一引數接收之 32 位元相對或絕對逾時而導出。

【0663】 該計時器佇列使用該 C_THREAD_PENDING_QUEUE_TIMER_PTR_INDEX 以依據逾時之順序儲存線程描述項之一列表，而最近之逾時位於該列表之首。注意這些線程描述項亦將為該優先度列表的成員，藉由該第二組指標而由 C_THREAD_PENDING_QUEUE_PRIORITY_PTR_INDEX 所指

示。第 51 圖中顯示該逾時佇列的基本結構。相對於該立即處理器之逾時的逾時被儲存，而非儲存絕對逾時數值於該計時器佇列之個別線程描述項成員中。此數值係使用一浮動點表示而被儲存於一 16 位元欄位中，其中該浮動點計時器格式之尾數包含該絕對數量之 11 重要位元以及一 5 位元指數。該計時器佇列之標題元件的逾時欄位總是被複製於該 TSPM TimeHeadTimeout 暫存器中，且隨後重置。

【0664】 該時間同步結構之操作使用數個永久內部資源：

● 【0665】 - TimerHeadIndex：對該計時器列表之一標題指標。

● 【0666】 - TimerNumElements：在該計時器佇列中的元件數量。

● 【0667】 - TimerHeadTimeout：該標題元件之逾時的一快照。

● 【0668】 - TimeDivider：該系統時脈之一預除器（prescaler）。

● 【0669】 - TimerDividerCounter：該除法器之一倒數資源。

● 【0670】 - TimerCounter：一 32 位元計數器資源單調地於每個預除時脈刻度上增加。

● 【0671】 - TimerErrorAdjustCounter：用於聚集及容納錯誤的一 32 位元計數器。

● 【0672】 若該 TimerDivider 暫存器被設為零，則計時器功能被關閉。

● 【0673】 計時器佇列週期行為

【0674】 第 52 圖說明在每個預除時脈脈動上發生的操作。該 TimerHeadTimeout 於沒有等待計時器式同步時（該計時器佇列位於一等待狀態中）不為零，因此不採取行動。當 TimeHeadTimeout 變成零時，且該計時器佇列不為空的時，該系統依據 TimerErrorAdjustCounter 之數值而採用兩種狀態之一。若該 TimerErrorAdjustCounter 為零，則 TimerHeadTimeout 之逾時已發生於此週期上且建立一計時器同步基本指令，其將最終造成自該計時器佇列（以及後勤目的之優先佇列）之一提取。緊接其後該 TimerErrorAdjustCounter 被單調（monotonically）增加直到其於一時間事件基本指令之處理已完成後被重置。

【0675】 計時器佇列事件

【0676】 有三種造成計時器佇列操作的事件：

【0677】 - 一時間事件基本指令
（C_TSPM_CMD_TIME_PRIMITIVE）

【0678】 - 一線程推送事件伴隨一非零逾時（具有被設為零之逾時的線程並未被至於該計時器佇列中）

【0679】 - 發生於一後勤實施而將一線程自該計時器佇列移除的一非計時器式同步事件。

【0680】 第 53 圖說明該逾時邏輯之操作模式的一極基本表示。當在該等待狀態中該 TimerHeadTimeout 不為零且依據該預除時脈單調地降低。TimerErrorAdjustCounter 在此狀態中被維持於零。當該 TimerHeadTimeout 到達零時，該計時器佇列之標題已逾時而該 FSM 轉換為提供提取操作之動態狀態。

在此狀態中，TimerHeadTimeout 為零而該 TimerErrorAdjustCounter 於每個週期上被單調地增加。此錯誤計數被用於判定執行一先前逾時事件所耗費的時間是否已造成後續逾時事件為適合的。一旦沒有進一步適合的逾時事件出現，該 FSM 轉換回該等待狀態中而該 TimerErrorAdjustCounter 被重置。

【0681】 導出提取操作之一潛在計時器序列的第一者係從該線程描述項中的一重置逾時欄位所導出（參見第 52 圖）。此後，該 TSPM 必須評估後續提取是否為適合的，一額外資源 TimerLastError 為此被用於維持所有累積提取線程描述項逾時差異（deltas）的一總和。在該計時器佇列之後續成員的每個進階重複上，該 TimerLastError 係自該 TimerErrorAdjustCounter 去掉以建立一正規化錯誤計數，其與在該計時器佇列標題之新線程描述項的逾時相比較。若此線程中的逾時差異小於該正規化錯誤計數，此線程描述性亦應被提取。一開始該 TimerLastError 為零，因而該線程逾時差異被直接與該 TimerErrorAdjustCounter 相比較。第 54 圖說明該線程 1 逾時已經過且該相關提取操作已發生之後的先前計時器佇列結構。注意該 TimerLastError 已被該線程 2 差異更新，而該線程 1 提取操作之過程意指線程 2 現在亦為適合的。

【0682】 第 55 圖說明該線程 2 之提取之後的佇列狀態。注意該線程 2 差異已被加入至 TimerLastError 而建立該線程描述項差異之總和的一執行中累積。注意線程 2 上的提取操作耗

費一足夠時間長度，故線程 3 現在為適合的。

【0683】 第 56 圖說明線程 3 之提取後的狀態。在此情形中該後續線程一線程 4 不適合於一提取，因而該計時器佇列之狀態可回到等待。該 `TimerHeadTimeout` 必須如圖被重置。

【0684】 注意當自一動態狀態轉換回該等待狀態時，該 `TimerHeadTimeout` 必須被正確地重置。此藉由自該新時機佇列標題之差異除去該 `TimerErrorAdjustCounter` 以及該 `TimerLastError` 之間的差而達成。

● 【0685】 當一新線程被導入或推送至該時機佇列時，必須考慮兩種情形：對該標題之一推送以及對該時機佇列之本體的一推送。

【0686】 - 對於對該標題之一推送而言，該 `TimerHeadTimeout` 被單純地設於該線程差異。當該佇列為非空的時，該舊標題描述項差異被設為該 `TimerHeadTimeout` 檢去該新線程差異。

● 【0687】 - 對於對該計時器佇列區塊之本體的一推送而言，區塊必須遍尋該計時器列表以識別在何處插入該線程描述項。該列表中次一者的差異隨後被調整以容納該新線程之差異的附加（該新線程差異係自該次一線程差異除去）。

【0688】 提取操作

【0689】 提取操作不論是由於一計時器或一外部事件同步事件均被相似地處理。考慮兩種情形：該線程提取之處位於該計時器佇列之開頭以及不在該處。在前者情形中有三種方案；其中該計時器佇列係位於該等待狀態中以及該計時器佇

列係位於動態狀態。

【0690】 - 對於自一‘等待’計時器佇列之開頭的一提取操作而言，該 `TimerHeadTimeout` 被加入至該計時器佇列之次一成員的計時器差異以形成新的 `TimerHeadTimeout`（注意在計時器式提取中 `TimerHeadTimeout` 之數值將總是為 0）。

【0691】 - 對於自一‘動態’計時器佇列之開頭的一提取操作而言，且該 `TimerErrorAdjustCounter` 大於該次一線程描述項中的差異時一亦即該次一線程適於一計時器式同步，該錯誤計數器—`TimerErrorAdjustCounter` 被重新置於該提取線程之差異的底部。

【0692】 - 對於自一‘動態’計時器佇列之開頭的一提取操作而言，且該 `TimerErrorAdjustCounter` 不大於該次一線程描述項中的差異時一亦即該次一線程不適於一計時器式同步，該差異由該錯誤計數器降低且以該結果更新該 `TimerHeadTimeout`。

【0693】 當該提取線程描述項不位於該計時器列表之開頭時，該計時器佇列中次一線程的差異必須由目前被移除之線程中的差異所降低。

【0694】 方法介面

【0695】 除了用於等待及計時器佇列之連結列表狀態操作（取得及設置狀態）之外，該等待管理員表示以下命令於其介面上：

【0696】 同步基本指令（`C_TSPM_CMD_SYNC_PRIMITIVE`）

【0697】 呼叫端：TSIF, TSIC

【0698】 該同步基本指令命令送出一控制封包，其釋放儲存於一等待佇列中的一封鎖線程描述項。該引數被說明於下：

未簽署短 PendingQueueID	該同步基本指令相關之等待佇列的識別
未簽署字元 Type	同步之類型。其為以下任一者： <ul style="list-style-type: none"> - UNICAST—同步該參照等待佇列的第一相符線程 - MULTICAST—同步該參照等待佇列的所有相符線程 - BROADCAST—同步所有等待佇列的所有相符線程
未簽署長（unsigned long）	參照。對於此同步基本指令相關者的依賴參照

【0699】 增加線程至等待佇列。

【0700】 呼叫端：TSIF

【0701】 此命令增加一線程描述項至一新的或一既有工作佇列。該命令藉由該工作佇列中一線程描述項之存在而被暗示該等待管理員。下表說明關於該命令之該線程描述項的欄位。

CREATE_NEW_PQ_FLAG	判定是否必須對此線程建立一新等待佇列
未簽署短 PendingQueueID	增加該線程描述項之等待佇列
未簽署長度量	該線程之適合度，以用於排序該線程描述項之插入
TierID	當變成未封鎖時將此線程描述項加入佇列的準備佇列排

	程層
DependancyTimeout	用於在該計時器佇列中排序的逾時

【0702】 處理標記線程

【0703】 呼叫端：TSIF

【0704】 此命令僅傳送一標記線程至該排程管理員工作佇列。其被用於確保在一標記線程有造成一排程器拆卸（tear-down）之任何可能性時所有依賴線程已被該等待管理員所處理。

【0705】 同步基本指令

【0706】 呼叫端：TSIF

【0707】 此命令送出一同步基本指令至一特定等待佇列。以下引數被呈現於該命令結構中：

WME 索引	此引數可被用於透過一字面參照釋放一 WME 至其 WME 位址
等待佇列索引	應用之等待佇列
同步參照	將與該特定等待佇列上的線程中的封鎖參照相比較的同步參照
同步類型	該同步之類型。其為以下任一者： <ul style="list-style-type: none"> - UNICAST—同步該參照等待佇列的第一相符線程 - MULTICAST—同步該參照等待佇列的所有相符線程 - BROADCAST- 同步所有相符線程

【0708】 更新度量

【0709】 呼叫端：TSIF

【0710】 此命令更新一封鎖線程之度量並造成該適當等待佇列的一重排（resort）。若該識別線程不再被封鎖，則該命令可被傳送至該 TSSM。以下引數被表示於該命令結構中。

度量	用於該識別線程的新度量
----	-------------

【0711】 解鎖等待佇列

【0712】 呼叫端：TSIF

【0713】 此命令解鎖一等待佇列，因而當其變成空的時，其可被釋放回該空閒列表。以下引數被表示於該命令結構中。

WME 索引	將被解鎖之 WME 索引
--------	--------------

【0714】 TSOM—輸出管理員

【0715】 該輸出管理員管理該分派佇列結構—其參照該次一執行線程描述項—以及該目前執行線程的執行度量。

【0716】 1.1.1 結構

【0717】 該 TSOM 之結構被集中於該分派佇列結構上，如第 58 圖所示。該輸出管理員維持分派佇列描述項之一列表，其中每個 DQD 係透過該 ProcElementID 欄位而與一系統處理資源實體相關。對於該 DQD 有兩群組之元件，該 DQD 緊密地整體參照該 TSOM 的功能。該第一群組—該執行中心元件—含有該 ProcElementID、度量 0、度量 1 而預設度量 0 參照並被管理於該任務之執行狀態中。該第二群組—該準備佇列中心元件—含有該根排程器索引，該先佔索引以及該次先佔索引參照等待執行之線程的準備佇列結構。該 TSOM 亦管理該帶外傳信（典型地為一岔斷）至該系統中的處理資源實體以及從該準備佇列結構之線程描述項的提取。

【0718】 該執行中心元件

【0719】 下文為該分派佇列描述項中該執行中心元件之使用的一簡要描述：

【0720】 - ProcElementID—此為一靜態欄位，其儲存參照該分派佇列描述項所連接之處理資源實體的一索引。

【0721】 - 度量、1 為動態更新欄位，其被用於儲存該目前執行中任務的度量（包括該‘閒置任務’以及潛在的各種電源降低狀態）

● 【0722】 - 預設度量 0 為用於支援一最佳化的一靜態欄位，藉以於推送該目前執行中線程回到該 SystemWeaver 伺服器因而定義上成為閒置時可自動儲存該閒置度量。

【0723】 該準備佇列中心元件

【0724】 下文為該分派佇列描述項中該準備佇列中心元件之使用的一簡短描述。

● 【0725】 - 根排程器索引為對於與該分派佇列有關之排程根層的一靜態參照。

【0726】 - 該先佔索引為一動態欄位，其儲存用於一特定處理資源之次一執行的最適候選者。該先佔索引完全被管理於該 TSOM 之中且於適當處被設為一分派佇列事件之一結果。

【0727】 - 該次一先佔索引為一動態欄位，其儲存該排程階層式中的次一最適線程的一主層或該線程索引本身。該次一先佔索引僅被該 TSSM 設置且被用作為告知該 TSOM 該最適線程之準備佇列結構中的位置的一媒介。

【0728】 分派佇列事件

【0729】 分派佇列事件發生於兩原因：

【0730】 - 一再排程事件：一分派佇列事件發生於該排程管理員（TSSM）中的一再排程操作在任何時間識別該分派佇列所需之狀態的一改變—即一先佔一時，典型地係由於處理一準備佇列事件（推送、提取或度量操作）。

【0731】 - 一分派佇列提取事件：該先佔索引所參照之線程索引已被該介面管理員（TSIF）所提取。此事件藉由該分派佇列描述項本身之中的一“分派佇列提取”旗標（由 C_DISPATCH_DESC_POPPED_FLAG）而被傳信至該 TSOM。

【0732】 在一分派佇列提取事件的情形中，該提取線程轉換為該停滯狀態（參見第 41 圖），且若其並未出現於該處，則其被推回至該工作佇列中以供該 TSSM 釋放。此後該 TSOM 從該準備佇列階層式切斷該線程。

【0733】 對於再排程及分派佇列提取事件兩者而言，一分派佇列事件之處理持續以起始該分派佇列描述項中的先佔索引的一重填（repopulation）。一般而言，若該次一先佔索引並非具有資格的一線程索引，則藉由自該次一先佔索引所識別之排程層下行該排程階層式而將其填入。一旦完成，該識別之線程索引被置於該先佔索引欄位中且自該準備佇列虛擬地提取該線程，亦即其從該準備狀態轉換為該提取狀態（參見第 41 圖）。此虛擬提取係藉由鎖定該最適線程、標記其為提取的並加入旗標於一事件回到該 TSSM 以供再排程而顯露。

【0734】 在特定情形下，一分派佇列事件造成一岔斷。若該再填入先佔索引含有一有效線程描述項索引且該岔斷被啓

動，則對於與此分派佇列相關之系統處理器的岔斷將被確立（asserted）於所有情形中—除非該早期岔斷確立旗標被啟動且該次一先佔索引為一線程，其中其將已被該 TSSM 所確立。

【0735】 該分派佇列度量亦被更新。

【0736】 雖然該分派佇列度量係表示目前執行線程的適合度，但該分派佇列係被先佔或者一分派佇列提取已發生。因此，該執行線程將被先佔（在此情形中該度量之更新最差為稍微過早的），或者一新線程正在執行且該分派佇列度量無論如何將被覆寫。

【0737】 當一再排程已發生而已侵佔（usurp）一既有次一先佔索引時，該既有先佔索引必須被虛擬清理回該準備佇列結構中（參見第 41 圖）。在該 TSOM 中該線程僅被標記為清理的且被推回該 TSSM 工作佇列以供處理。

【0738】 若無法找到一適合線程，則該操作便完成，該客戶端（處理資源）將於這些情形下閒置。

【0739】 設置分派佇列適合度度量

【0740】 該分派佇列適合度度量反映目前在由 ProcElementID 所索引之處理資源實體上執行之任務的執行優先度。然而在特定最佳化中，該分派佇列適合度度量亦可反應將開始執行之任務的執行優先度。

【0741】 該分派佇列優先度度量被用於控制先佔，其由於多種理由而加以操作：

【0742】 - 開始一新任務

【0743】 - 完成目前任務

【0744】 - 優先度反轉

【0745】 - 電源管理

【0746】 在所有情形中該目的係調整準備佇列之排程的操作。當該更新造成一集區根節點之修改時，該節點被加入旗標至該 TSSM 以供一再排程操作。在集區參與者及非集區參與者中的度量傳遞的細節可參見前文。

【0747】 作為一最佳化，當該處理資源實體推送該目前執行之線程回到 SystemWeaver 中的準備或封鎖狀態。該執行優先度被自動地重設至該預設度量。

【0748】 方法介面

【0749】 除了在該分派佇列連結列表上的基本狀態操作之外（取得及設置狀態），該輸出管理員表示以下命令於其介面上：

【0750】 自動執行度量更新
(C_TSOM_CMD_SERVICE_TIME_SLICE_EXPIRE)

【0751】 呼叫端：TSIF

【0752】 此命令自動修改該識別處理器 ID 之分派佇列中保持之度量之度量 0 的最不重要位元。由於該引數為該處理器之識別而非該分派佇列描述項本身，故該命令一開始逐步進行該分派佇列描述項列表以找出適當的描述項。該修改僅為該最不重要位元的一單純反轉，假設該度量欄位之保留部分被適當地設置，此影響為降低該執行中線程的優先度，不論優先度係升幂或降幂的。

【0753】 該引數被說明於下：

未簽署特性 ProcID	執行優先度必須被更新之處理器的處理資源 ID
--------------	------------------------

【0754】 設置分派佇列度量

【0755】 呼掉端：TSIF（源自一專屬系統呼叫）

【0756】 此命令設置一特定分派佇列的執行度量。以下引數被呈現於該命令結構中：

WMEIndex	該分派佇列描述項之索引
度量	該識別分派佇列的新執行度量

【0757】 呼叫端：TSIF

【0758】 此命令設置一特定分派佇列之執行度量(0)為該預設數值，其亦被持有於該分派佇列中。對於此功能並無引數。

【0759】 分 派 佇 列 事 件
(C_SCHED_ENTRY_DISPATCH_LIST)

【0760】 呼叫端：TSSM

【0761】 此命令造成該分派佇列描述項於該準備佇列之一狀態改變需要時被更新。

【0762】 TSSM—排程管理員

【0763】 該排程管理員管理該準備佇列結構之主層及子層連結中固有的排程選擇。

【0764】 結構

【0765】 該 TSSM 為專門源自其工作佇列介面的供給命令（ fed commands ）且僅為事件驅動的。然而，特定行為為數個命令所共有，此行為被描述於下：

【0766】 再排程

【0767】 該再排程功能再評估該排程階層式中一定義點的排程選擇。該再排程操作為工作保留的，沒有額外的工作在其上被完成，且其係與階層式之部分有關，該階層式之狀態理解上可被一事件所影響。

【0768】 對於該再排程功能具有一特別有趣的引數：UpdateRequired。UpdateRequired 被用於在排程器層操作間傳遞更新狀態。舉例來說，雖然其他狀態仍須被管理，但不更新一子層中的選擇的一推送操作不需造成該主層中的一整體排程重複。在此情形中 UpdateRequired 可能為假。

【0769】 第 60 圖說明一再排程操作的一基本流程。內層排程器（intratierscheduler）執行一層中的排程，且造成該主層標題索引指標的更新。層間排程器（intertierscheduler）縮放該排程階層式中的後續階層朝向該分派節點。注意該層間排程功能係以該主索引加以呼叫，因而立即縮放階層式之一階層。集區層間排程為一特殊情形排程演算法，其為自一單一節點、該集區根節點扇出（fanout）至多個節點、該集區靜態節點的唯一演算法

【0770】 此章節之剩餘者描述該排程演算法之操作。

【0771】 內層排程器

【0772】 該最基本排程操作橫越一排程層中的元件的連結列表以識別何者為最適合且相應的更新該主層之標題指標。

【0773】 在該普通情形中，當一適當描述項被推送至一空的排程層時，該主層之標題指標以及元件數量被無條件地更新且該度量被有條件地自該新子層傳遞（依據度量傳遞運算子）

至該主層。

【0774】 第 61 圖說明一推送操作之更一般情形。該目前選擇之適合度以及該候選者選擇為數個因素之一組合：

【0775】 - 該選擇必須具有可被排程之內容。對於一線程而言此總是為真，但對於一排程器而言此係依據其從屬階層式之內容。

【0776】 - 若該描述項為一排程器，其必須為非靜止的，且其無效選擇旗標必須不被設置。。

● 【0777】 - 若該描述項為一標準線程，其必須為未鎖定的，或者其必須位於一工作佇列中。

【0778】 - 若該描述項為一標記線程，該主層總依賴計數必須為零，且該標記線程必須為留在該層中的唯一登錄。

● 【0779】 注意該候選者僅與該目前選擇一源自一先前排程操作之最適描述項相比較，若該候選者擊敗該優勝者，則其必須為新的優勝者。如前所述，“排程對”係與該主排程層中具有的演算法有關。該“更新主層”變數攜帶一指令回到該呼叫端，而此層之主層亦應由於此操作而被更新。

【0780】 在一再排程之一般情形中，例如當度量已被更新時，或者當一提取已發生時，該整個階層必須被重新更新以找出該新的最適候選者。此處理透過第 61 圖之操作進行數次直到整個階層已被重新評估，如第 62 圖所示。

【0781】 層間排程

【0782】 層間排程於每個排程器層執行一次，其於每個排程事件可總計為數次。層間排程高度依賴該主層之類型。概括

言之，層間排程持續呼叫內層排程直到該主層成為一分派佇列。此具有之一例外為遭遇一集區分配節點時。

【0783】 第 63 圖說明層間排程的基本流程。具有一唯一行為與一分派佇列描述項（DQD）主層及一集區根節點（PRN）主層有關。然而在其他情形中，該層間排程器僅以該目前主層取代該目前子索引（藉以向上重複該排程階層式）以及自我重新呼叫。

【0784】 第 64 圖說明處理於該層間排程常式中的分派佇列。首先，該分派佇列描述項（DQD）中含有之執行度量依據該 DQD 中定義之演算法對於該排程器根節點中具有之度量而被加以排程。若此操作判定需要一先佔，則依據開始該再排程操作之事件類型而實施一更新至該 DQD 次一先佔索引。若此事件為一線程推送，則該排程線程索引被直接置於該次一先佔索引欄位，否則使用該排程器根節點索引。其隨後落於該 TSOM 上以重複該排程階層式以找出該線程描述項。

【0785】 集區層間排程

【0786】 集區層間排程被用於識別該相關處理資源實體之何者一若有任一者的話一應被選擇以供服務該目前集區中具有之一線程描述項。在該情形中其以一獨特方式操作，由於不像此處描述之所有其他排程演算法，其典型地搜尋最不適合候選者。

【0787】 第 66 圖說明透過一集區層間排程操作的流程，由第 65 圖加以圖形化表示。該初始內層排程操作判定源自該排程階層式之候選者是否較該執行任務之任一者更為適合，如該

集區分配層中的集區靜態節點中具有之度量所指示。具有兩種結果，分別為是否需要一更新的一指示以及必須應用該更新之節點的識別。

【0788】 該演算法隨後進行以重複於該整體集區分配層周圍，其開始於該集區根節點之適當“次一指標”所指示之集區靜態節點。

【0789】 在每個重複中以及管理內勤功能時（例如線程計數器以及其他狀態之維護），在該集區附加層上呼叫層間排程。通常情形中層間排程持續向上前進該整體排程階層式直到其到達該分派節點。層間排程採用一引數“update required”，其指示進一步排程層是否應完全再評估該排程選擇。此旗標在兩種情形下被設置於集區間排程之內容中：

【0790】 - 目前被處理之集區靜態節點為被內集區排程識別為在該集區根節點下處理該最適線程的最適節點。

【0791】 - 該集區根節點下的階層不具有任何適合的排程候選者。

【0792】 內集區層排程

【0793】 第 67 圖說明內集區層排程的流程。在該第一實體中具有一最佳化，其降低僅推送（push only）操作的排程時間—此增進該系統對新任務成為可用的反應度。

【0794】 假設此並非一專屬推送操作，該排程操作設置該目前選擇以及候選者選擇至該集區中的頭兩個 PSN 節點。在每個重複上，對於該候選者排程該目前選擇。注意該相同排程演算法被用於集區分佈層作為其他排程層，然而由於在此方

案中該正確選擇係以正規標準顯示該最不適合度量，故選擇的個別演算法可為不同的。

【0795】 若該候選者指揮該目前選擇，亦即該目前選擇被更新至該候選者，則該候選者被更新至該層中的次一登錄且該處理持續直到該候選者成為該 PRN。

【0796】 在一專屬推送操作中避免此重複。該目前選擇及更新節點被單純設置至該 PRN 之主層（其定義該目前排程選擇），且該候選者為該 PRN 本身。

● 【0797】 在所有情形中該集區根節點隨後被檢查可排程內容，若無，則“無更新”狀態被設置且該演算法返回。然而若有可排程內容，該處理持續至該第二階段，藉此源自該 PRN 之既有選擇（在專屬推送操作中）或者該重複之結果（在其他情形中）被對於該 PRN 本身加以排程。若該 PRN 贏得此競爭則需要一更新，反之則否。

● 【0798】 範例

● 【0799】 第 68 圖說明與一線程描述項（節點#5）推送至一靜態排程元件（節點#3）有關的 TSSM 排程器處理。該第一內層排程操作發生於該再排程功能之內容中且係與該主節點#3 有關。再排程隨後向上移動該階層式之一層級且利用該主節點#2 呼叫層間排程。緊接其後該重複找出該主節點#1，其為一 DQD。因此沒有進一步對於該內層排程器之呼叫且一排程比較於該根節點中儲存之最適候選者之度量與該分派節點中儲存之執行線程之間完成。在此情形中，一先佔為適合的且一分派佇列事件被通信至該 TSOM。

【0800】 第 69 圖說明一更階層式的排程階層式。此處建立對該內層排程功能的一額外呼叫用於該階層式之額外階層。注意即使如此，該排程層 5 及 3 保持未受影響的，由於造成該再排程操作之排程事件超出其範圍。

【0801】 第 70 圖提供一第三範例，其為一集區排程操作之情形。如前所述，該線程推送事件發生於其中的該階層受到一內層排程操作。在此範例中，該新線程最適合於該集區根層中。該內集區層排程器隨後建立一呼叫至該內集區層排程器以判定該系統處理資源之任一者是否應被該新到達線程先佔。在此情形中，內集區層排程的結果為與在 WME 索引 7 之分派佇列描述項有關之處理資源實體應被先佔。

【0802】 內集區層排程隨後重複於該集區分配層周圍，首先呼叫節點 4 上的內層排程。隨後內層排程呼叫內層排程以更新排程器層 1，即使由於此並非該先佔層而不需完全排程一處理限於狀態資訊之維護，因此對於該 TSOM 沒有分派佇列事件，亦沒有對於該系統處理資源的岔斷。

【0803】 該次一呼叫為第 2 層上的層間排程。在此情形中該層被適當地排程以建立該新推送之線程描述項是否較任何其他候選者更為適合。該候選者度量最終被與該些儲存於該分派佇列描述項中的執行中線程以判定一先佔是否為適當的。在此範例中確實如此，因此一分派佇列事件被傳送至該 TSOM 且該系統處理資源實體岔斷被加上旗標。

【0804】 再者，僅有該些被該推送線程事件之範圍所觸及的排程層被再評估。

【0805】 方法介面

【0806】 透過該工作佇列專門地引導該 TSSM。

【0807】 服務線程描述項事件

【0808】 呼叫端：TSIF、TSPM、TSOM

【0809】 此命令之行爲係依賴該接收線程之旗標中的設定：

【0810】 - 若該推送或清理旗標被設置，則該主層之總線程元件計數被增加。若該提取旗標被設置，則其被降低（注意該之一組合的淨效果可爲 NULL）。

● 【0811】 - 已設置推送旗標的線程隨後被連結至該準備佇列階層式中。當其已自該封鎖狀態轉換時，該依賴線程之主層計數亦被降低。

【0812】 - 專屬於標記線程之爲一行爲係該主層之解鎖（轉換該排程器層爲該“免等待”狀態—參見第 42 圖）。

【0813】 - 該 TSSM 於處理此命令時請求在該停滯狀態中接收之任何線程描述項被釋放。

● 【0814】 此命令總是請求一再排程。

【0815】 再排程集區分配層

【0816】 呼叫端：TSOM

【0817】 此命令係由於一執行中線程之度量中的一改變而被呼叫。當該分派描述項參與一集區中時，這些度量被傳遞通過該分配階層式而最終呼叫此功能以再評估該排程選擇。

【0818】 再排程或刪除排程器層

【0819】 呼叫端：TSOM

【0820】 此命令係由 TSOM 基於兩原因而被呼叫：

【0821】 - 由於一執行中線程之度量中的一改變，其中該分派描述項並未參與一集區中，以再評估一先佔是否為適當的。此操作傳送一排程根節點。當該排程器含有線程描述項於其階層式中時，一再排程被請求。

【0822】 - 為了刪除一排程層，其中該 TSOM 以判定適合如此。此判定係依據該鎖定旗標、依賴線程計數以及子元件之數量（分別為不實的、0 及 0）該描述項之虛擬釋放自我表示為對該 TSIM 之一請求。

● 【0823】 推送一獨立元件

【0824】 呼掉端：TSIF

【0825】 此命令可被用於初始化過程中以推送靜態排程層至該排程階層式中，或者可於執行時被動態地用於增加線程描述項或動態排程階層式。

【0826】 更新線程度量

【0827】 呼叫端：TSIF

● 【0828】 更新該準備佇列階層式中的一線程描述項的度量。度量可僅於該線程描述項位於該準備狀態中而加以更新（第 41 圖）

【0829】 此命令造成一再排程。

【0830】 更新排程器狀態

【0831】 呼叫端：TSIF

【0832】 此命令提供排程器演算法、度量以及度量傳遞演算法的更新。該命令造成一再排程。

【0833】 啟動排程器層

【0834】 呼叫端：TSIF

【0835】 此命令啓動一靜止排程器層。該命令造成一再排程。

【0836】 撤銷排程器層

【0837】 呼叫端：TSIF

【0838】 此命令撤銷一靜止排程器層。

【0839】 TSM—記憶體管理員

【0840】 該記憶體管理員（第 71 圖）提供多工/去多工行爲以供聚集對該 TCM 之存取。其亦提供鎖定能力以確保在多個子區塊間分享之資源的完整度。

【0841】 結構

【0842】 從一結構觀點看來，該記憶體管理員主要可被視爲一多工/去多工，其聚集該六個可能請求者中對於該 TSM 的存取。其亦維持 WMEs 的完整度，其中多個子區塊藉由實施一區塊快取而嘗試存取該相同資源。

【0843】 存取聚集

【0844】 存取聚集係由一排程器所控制。此排程器爲非對稱的：

【0845】 - TSIF 具有最高優先度

【0846】 - TSOM 具有次一最高優先度

【0847】 - 所有剩餘請求者具有相同優先度，且被處理於一工作保留循環排序分配中。

【0848】 鎖定快取

【0849】 每個區塊具有一與四個鎖定之間的一分配。這些數量表示該子區塊請求者對於 WMEs 可具有專屬存取的數量。

請求一鎖定資源的一子區塊被封鎖直到該資源成為可用的。圍繞於該相同資源上的多個區塊之間的爭用係由優先度所解決。

【0850】 排程順序圖

【0851】 推送事件

【0852】 第 72 圖的順序圖說明一推送事件之後的子區塊間互動。注意該工作佇列於該 TSSM 及 TSOM 本身為單一線程時已被導入以儲存命令。請參照第 68 圖的一代表性排程階層式。

【0853】 在該推送事件之前的狀態為線程#4 係分派佇列描述項#1 之中的目前先佔索引。該第一再排程識別線程#5 較線程#4 更適合，由於分派佇列描述項#1 被推送至該 TSOM 工作佇列中。

【0854】 在該 TSOM 工作佇列中的一分派佇列描述項造成一分派佇列事件於該 TSOM 中。此事件虛擬地推送線程描述項#5 並相應地設置分派佇列度量。此推送操作造成該準備佇列中的一狀態改變，因而一再排程必須被呼叫。此藉由推送線程#5 至已設置推送旗標之 TSSM 工作佇列中而達成。

【0855】 由於線程#4 先前已被虛擬地推送，現在期需要被清理回到該準備佇列結構中。此亦構成該準備佇列結構中的一狀態改變，因而需要另一再排程。此藉由推送線程#4 至已設置清理旗標之 TSSM 準備佇列結構中而達成。

【0856】 注意該第二及第三再排程操作無法被合併為該虛擬推送線程，且該清理線程可位於該準備佇列階層式之大量區

別部分中。

【0857】 推送事件

【0858】 第 73 圖中的順序圖說明一線程描述項自該“虛擬”分派佇列被推送時該 TSIF、TSOM 及 TSSM 之間的互動。

【0859】 該推送命令本身係由該 TSIF 在該命令介面或系統互連上被接收。該 TSIF 藉由推送該分派佇列描述項至已設置該推送旗標之 TSOM 上而送出一分派佇列推送命令至該 TSOM。

【0860】 該 TSOM 工作佇列中的該分派佇列描述項造成一分派佇列事件於該 TSOM 中。該分派佇列事件控制器指示該 TSSM 請求剛已被推送之線程描述項，在此情形中線程#5 被置回該空閒列表上。儲存用於執行之次一最適候選者的次一先佔索引隨後被該 TSOM 虛擬地推送。此表示該準備佇列結構之一狀態改變，因而該 TSSM 被指示由該 TSOM 再排程一該次一先佔線程被通送已以設置該推送旗標之 TSSM 工作佇列中。

【符號說明】

無

【生物材料寄存】

國內寄存資訊【請依寄存機構、日期、號碼順序註記】

無

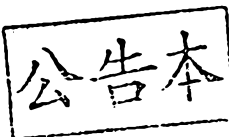
國外寄存資訊【請依寄存國家、機構、日期、號碼順序註記】

無

【序列表】(請換頁單獨記載)

無

發明摘要



※ 申請案號：102118985

※ 申請日：2006 年 09 月 29 日

※IPC 分類：G06F 9/48 (2006.01)

G06F 9/50 (2006.01)

原申請案號：95136470

【發明名稱】（中文/英文）

用於多核心架構之排程的方法、電腦程式、及電腦可讀取媒體

METHOD, COMPUTER PROGRAM, AND COMPUTER READABLE
MEDIUM FOR SCHEDULING IN A MULTICORE ARCHITECTURE

【中文】

此發明係有關於在一多核心處理器中排程線程之步驟。可使用依據執行適合度之順序列出可執行交易的至少一分配佇列以及含有多個連結之獨立可執行交易排程器之多層排程器來排程可執行交易。這些各者包括一排程演算法以供判定用於執行之最適可執行交易。該最適可執行交易係從該多層排程器被輸出到至少一分配佇列。

【英文】

This invention relates to scheduling threads in a multicore processor. Executable transactions may be scheduled using at least one distribution queue, which lists executable transactions in order of eligibility for execution, and multilevel scheduler which comprises a plurality of linked individual executable transaction schedulers. Each of these includes a scheduling algorithm for determining the most eligible executable transaction for execution. The most eligible executable transaction is outputted from the

multilevel scheduler to the at least one distribution queue.

【代表圖】

【本案指定代表圖】：第（ 1 ）圖。

【本代表圖之符號簡單說明】：

無

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：

無

multilevel scheduler to the at least one distribution queue.

【代表圖】

【本案指定代表圖】：第（ 1 ）圖。

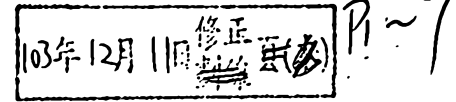
【本代表圖之符號簡單說明】：

無

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：

無

申請專利範圍



1. 一種管理電源消耗的方法，該方法包括以下步驟：

藉由一多核心處理器的一第一複數個處理器元件決定配置用於執行的至少一個線程，其中該等第一複數個處理器元件可操作性地在一第一電源狀態中執行該至少一個線程；

將該多核心處理器的一第二複數個處理器元件進行配置以在一第二電源狀態中操作，其中在該第二電源狀態中操作的一特定處理器元件較在該第一電源狀態中操作的該特定處理器元件可操作地消耗較少電源，及其中配置該等第二複數個處理器元件的該步驟進一步包含執行一第一操作以允許該第二複數個處理器元件維持狀態資料；及

將該等第二複數個處理器元件的至少一個處理器元件進行配置以在一第三電源狀態中操作，其中在該第三電源狀態中操作的至少一個處理器元件較在該第二電源狀態中操作的該至少一個處理器元件可操作地消耗較少電源，及其中配置該至少一個處理器元件的該步驟進一步包含執行一第二操作以允許該至少一個處理器元件犧牲該狀態資料。

2. 如請求項 1 所述之方法，其中該第一操作包括時脈閘控。
3. 如請求項 1 所述之方法，其中該第二操作從由以下各者所組成的一群組中選擇：電壓調整、頻率調整及電源隔離。
4. 如請求項 1 所述之方法，其中配置該至少一個處理器元件的該步驟進一步包括將該至少一個處理器元件進行配置以在一預定週期的時間後在該第三電源狀態中操作。

5. 如請求項 4 所述之方法，其中該預定週期的時間包括在該第二電源狀態中操作的一週期的時間。

6. 如請求項 1 所述之方法，其中配置該至少一個處理器元件的該步驟進一步包括回應於決定與該等第二複數個處理器元件關聯的工作負載之一減少，而將該至少一個處理器元件進行配置以在該第三電源狀態中操作。

7. 如請求項 1 所述之方法，進一步包括以下步驟：

● 回應於決定工作負載之一減少，而將該至少一個處理器元件配置以在該第一電源狀態中操作。

8. 如請求項 1 所述之方法，進一步包括以下步驟：

回應一信號，而將該至少一個處理器元件配置以在該第一電源狀態中操作。

9. 如請求項 8 所述之方法，其中該信號從由以下各者所組成之一群組中選擇：一帶外信號與一岔斷。

● 10. 如請求項 1 所述之方法，進一步包括以下步驟：

使用在該第一電源狀態中操作的該等第一複數個處理器元件執行該至少一個線程。

11. 如請求項 1 所述之方法，其中該等第一複數個處理器元件包括一集區的處理器元件。

12. 如請求項 1 所述之方法，其中該第二電源狀態與一優先模式相關聯。

13. 一種具有用於致使一電腦系統執行一種管理電源消耗的

方法之電腦可讀取程式碼於其中的非暫態電腦可讀取媒體，該方法包括以下步驟：

藉由一多核心處理器的一第一複數個處理器元件決定配置用於執行的至少一個線程，其中該等第一複數個處理器元件可操作性地在一第一電源狀態中執行該至少一個線程；

將該多核心處理器的一第二複數個處理器元件進行配置以在一第二電源狀態中操作，其中在該第二電源狀態中操作的一特定處理器元件較在該第一電源狀態中操作的該特定處理器元件可操作地消耗較少電源，及其中配置該等第二複數個處理器元件的該步驟進一步包含執行一第一操作以允許該第二複數個處理器元件維持狀態資料；及

將該等第二複數個處理器元件的至少一個處理器元件進行配置以在一第三電源狀態中操作，其中在該第三電源狀態中操作的至少一個處理器元件較在該第二電源狀態中操作的該至少一個處理器元件可操作地消耗較少電源，及其中配置該至少一個處理器元件的該步驟進一步包含執行一第二操作以允許該至少一個處理器元件犧牲該狀態資料。

14. 如請求項 13 所述之非暫態電腦可讀取媒體，其中該第一操作包括時脈閘控。

15. 如請求項 13 所述之非暫態電腦可讀取媒體，其中該第二操作從由以下各者所組成的一群組中選擇：電壓調整、頻率調整及電源隔離。

16. 如請求項 13 所述之非暫態電腦可讀取媒體，其中配置該

至少一個處理器元件的該步驟進一步包括將該至少一個處理器元件進行配置以在一預定週期的時間後在該第三電源狀態中操作。

17. 如請求項 16 所述之非暫態電腦可讀取媒體，其中該預定週期的時間包括在該第二電源狀態中操作的一週期的時間。

18. 如請求項 13 所述之非暫態電腦可讀取媒體，其中配置該至少一個處理器元件的該步驟進一步包括回應於決定與該等第二複數個處理器元件關聯的工作負載之一減少，而將該至少一個處理器元件進行配置以在該第三電源狀態中操作。

19. 如請求項 13 所述之非暫態電腦可讀取媒體，其中該方法進一步包括以下步驟：

回應於決定工作負載之一減少，而將該至少一個處理器元件配置以在該第一電源狀態中操作。

20. 如請求項 13 所述之非暫態電腦可讀取媒體，進一步包括以下步驟：

回應於一信號，將該至少一個處理器元件進行配置以在該第一電源狀態中操作。

21. 如請求項 20 所述之非暫態電腦可讀取媒體，其中該信號從由以下各者所組成的一群組中選擇：一帶外信號與一岔斷。

22. 如請求項 13 所述之非暫態電腦可讀取媒體，其中該方法進一步包括以下步驟：

使用在該第一電源狀態中操作的該等第一複數個處理

器元件執行該至少一個線程。

23. 如請求項 13 所述之非暫態電腦可讀取媒體，其中該等第一複數個處理器元件包括一集區的處理器元件。

24. 如請求項 13 所述之非暫態電腦可讀取媒體，其中該第二電源狀態與一優先模式相關聯。

25. 一種包括一處理器與一記憶體的系统，其中該記憶體包含經該系統執行時而施行管理電源消耗的一方法之指令，該方法包括以下步驟：

藉由一多核心處理器的一第一複數個處理器元件決定配置用於執行的至少一個線程，其中該等第一複數個處理器元件可操作性地在一第一電源狀態中執行該至少一個線程；

將該多核心處理器的一第二複數個處理器元件進行配置以在一第二電源狀態中操作，其中在該第二電源狀態中操作的一特定處理器元件較在該第一電源狀態中操作的該特定處理器元件可操作地消耗較少電源，及其中配置該等第二複數個處理器元件的該步驟進一步包含執行一第一操作以允許該第二複數個處理器元件維持狀態資料；及

將該等第二複數個處理器元件的至少一個處理器元件進行配置以在一第三電源狀態中操作，其中在該第三電源狀態中操作的至少一個處理器元件較在該第二電源狀態中操作的該至少一個處理器元件可操作地消耗較少電源，及其中配置該至少一個處理器元件的該步驟進一步包含執行一第二操作以允許該至少一個處理器元件犧牲該狀態資料。

26. 如請求項 25 所述之系統，其中該第一操作包括時脈閘控。
27. 如請求項 25 所述之系統，其中該第二操作從由以下各者組成的一群組中選擇：電壓調整、頻率調整及電源隔離。
28. 如請求項 25 所述之系統，其中配置該至少一個處理器元件的該步驟進一步包括將該至少一個處理器元件進行配置以在一預定週期的時間後在該第三電源狀態中操作。
29. 如請求項 28 所述之系統，其中該預定週期的時間包括在該第二電源狀態中操作的一週期的時間。
30. 如請求項 25 所述之系統，其中配置該至少一個處理器元件的該步驟進一步包括回應於決定與該等第二複數個處理器元件關聯的工作負載之一減少，而將該至少一個處理器元件進行配置以在該第三電源狀態中操作。
31. 如請求項 25 所述之系統，其中該方法進一步包括以下步驟：
- 回應於決定工作負載之一減少，將該至少一個處理器元件配置以在該第一電源狀態中操作。
32. 如請求項 25 所述之系統，進一步包括以下步驟：
- 回應一信號，將該至少一個處理器元件進行配置以在該第一電源狀態中操作。
33. 如請求項 32 所述之系統，其中該信號從由以下各者所組成的一群組中選擇：一帶外信號與一岔斷。
34. 如請求項 25 所述之系統，其中該方法進一步包括以下步

驟：

使用在該第一電源狀態中操作的該等第一複數個處理器元件執行該至少一個線程。

35. 如請求項 25 所述之系統，其中該等第一複數個處理器元件包括一集區的處理器元件。

36. 如請求項 25 所述之系統，其中該第二電源狀態與一優先模式相關聯。