(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 February 2017 (02.02.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/016480 A1

(51) International Patent Classification:
G06F 9/46 (2006.01)

(21) International Application Number:
PCT/CN2016/091893

(22) International Filing Date:
27 July 2016 (27.07.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/810,205 27 July 2015 (27.07.2015) US

(71) Applicant: **HUAWEI TECHNOLOGIES CO., LTD.** [CN/CN]; Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong 518129 (CN).

(72) Inventors: **CHEN, Liya**; 3391 Pinnacle Dr., San Jose, California 95132 (US). **TIAN, Chen**; 32531 Seaside Drive, Union City, California 94587 (US). **YE, Feng**; 3650 Kaneff Cres 1202, Mississauga, Ontario L5A 4A1 (CA). **HU, Ziang**; 3001 Belize Way, Union City, California 94587 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: SYSTEM AND METHOD FOR MULTITHREADED PROCESSING

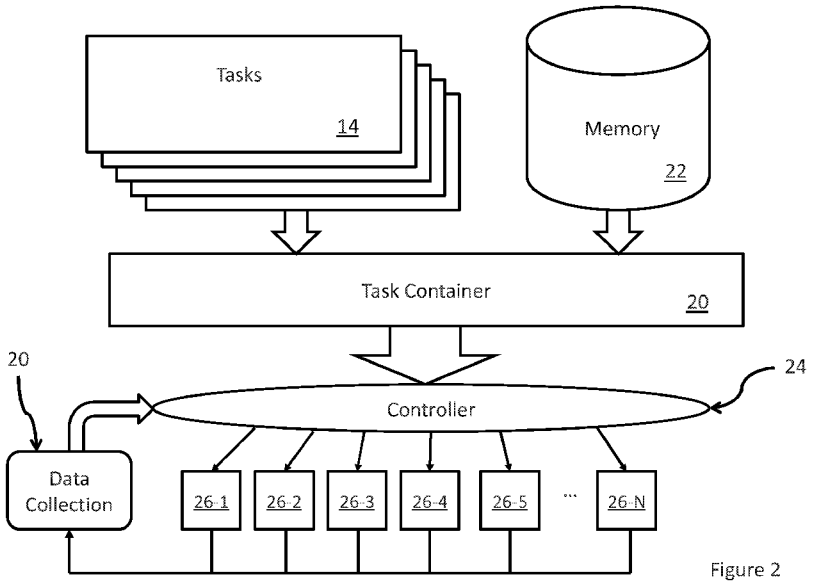

Figure 2

(57) **Abstract**: A method for operating a multithread processing system is provided, including assigning, by a controller, a subset of a plurality of tasks to a plurality of threads during a time N, collecting, by the controller, data during the time N concerning the operation of the plurality of threads, analyzing, by the controller, the data to determine at least one condition concerning the operation of the plurality of threads during the time N, and adjusting, by the controller, a number of the plurality of threads available in time N + 1 in accordance with the at least one condition.

WO 2017/016480 A1

System and Method for Multithreaded Processing

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to and benefit of U.S. non-provisional patent application Serial No. 14/810,205, filed on July 27, 2015, and entitled "System and Method for Multithreaded Processing," which application is hereby incorporated by reference.

TECHNICAL FIELD

[0002] The present invention relates to systems and methods for managing the operation of multiprocessor computing systems and, in particular embodiments, to systems and methods for multicore microprocessors.

BACKGROUND

[0003] Current hardware technology has developed System-on-Chip (SOC) devices with multiple cores. Currently, SOCs with 12 to 64 hardware threads exist, and there are plans for chips with 50+ cores (Intel Unveils New Product Plans for High-Performance Computing, http://www.intel.com/pressroom/archive/releases/2010/20100531comp.htm). In addition, multiprocessing computing systems have been developed that use from two to thousands of processors (Multiprocessing Systems, IEEE Transactions on Computers, Vol. C-25, No. 12, December 1976, http://63.84.220.100/csdl/trans/tc/1976/12/01674594.pdf). To take advantage of these multiprocessor or multithread systems, operating systems divide applications into discrete tasks that can be processed separately.

[0004] However, there are trade-offs between the number of threads used and the overhead accompanying multiple threads. When tasks on separate threads are completed, the results usually must be coordinated with tasks running on other threads. This creates difficult timing issues because tasks often require different amounts of time to accomplish. Also, subsequent tasks may require results from current tasks. This creates difficult synchronization issues. Another issue is power consumption. If all hardware threads are running at full capacity, the chip's power delivery systems may be over-taxed and may generate more heat than can be safely dissipated. One of the key challenges for parallel runtime systems is how to use hardware threads efficiently and to provide the best performance.

SUMMARY OF THE INVENTION

[0005] A method for operating a multithread processing system is provided, including assigning, by a controller, a subset of a plurality of tasks to a plurality of threads during a time N, collecting, by the controller, data during the time N concerning the operation of the plurality of threads, analyzing, by the controller, the data to determine at least one condition concerning the operation of the plurality of threads during the time N, and adjusting, by the controller, a number of the plurality of threads available in time N + 1 in accordance with the at least one condition.

[0006] A multithread processing system controller is provided. The multithread processing system controller includes a storage system storing programming and a processor coupled to the storage system and executing the programming. The programming configures the multithread processing system controller to assign a subset of the plurality of tasks to a plurality of threads during a time N, collect data during the time N concerning the operation of the plurality of threads, analyze the data to determine at least one condition concerning the operation of the plurality of threads during the time N, and adjust a number of the plurality of threads available in time N + 1 in accordance with the at least one condition.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] For a more complete understanding of the present invention, and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawing, in which:

[0008] Figure 1 illustrates the operation of an operating system for use in a multithreaded processing system;

[0009] Figure 2 illustrates the operation of one embodiment;

[0010] Figure 3 is a flow diagram of a throttling decision process according to an embodiment;

[0011] Figure 4 is a flow diagram of the decision process of the process shown in Figure 3; and

[0012] Figure 5 is an illustration showing a throttling operation performed by the embodiment of Figure 2.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

[0013] The making and using of the presently preferred embodiments are discussed in detail below. It should be appreciated, however, that the present invention provides many applicable inventive concepts that can be embodied in a wide variety of specific contexts. The specific embodiments discussed are merely illustrative of specific ways to make and use the invention, and do not limit the scope of the invention.

[0014] Figure 1 is a block diagram showing the basic operation of a multithread processing system including multiple processors. An example of such a system is a smart phone having a multicore processor. Another example is a virtualization environment wherein multiple virtual machines (VMs) can be instantiated and used for processing at least part of a processing job or application. However, other systems and devices are contemplated and are within the scope of the description and claims. One or more applications are initiated by calls from operating system 12. Operating system 12 is designed to exploit multithreaded systems (a thread is a discrete processing unit, such as a single processor in a multiprocessor system or a single core in a multicore system). To do this, operating system 12 breaks the work to be done into relatively discrete tasks 14. These tasks may be performed for an application or the operating system itself. The goal is to process the tasks as efficiently as possible.

[0015] To achieve this, the operating system creates tasks that require as little data from a concurrently running thread as possible. It is also desirable that no two threads require the same data from memory at the same time. This and many other characteristics of the operating threads can create problems such as stalling while one thread waits for the completion of another and excess power usage when all threads are concurrently running power-intensive processes.

[0016] Figure 2 is a block diagram of an embodiment that is implemented in software running within the operating system of the multithreaded system. Tasks 14 are provided to task container 20. Memory 22 is connected to task container 20. Task container 20 receives and stores task and caches the memory necessary for that task. The task then awaits instructions from controller 24 that assign the task to one of threads 26-1 to 26-N. In some embodiments, the threads 26-1 to 26-N comprise separate processors; in some embodiments, the threads 26-1 to 26-N comprise cores in a multicore processor; and some embodiments are a combination of both.

The term thread is used herein to indicate a discrete processing entity. However, each thread may not be absolutely discrete. Each thread can access memory and utilize other resources. The use of the term discrete herein only refers to the fact that each thread can process tasks relatively autonomously from the other threads. The threads can access independent resources or can access shared resources.

[0017] Controller 24 includes a data collection module 28. Data collection module 28 collects statistical data related to the tasks that are running on threads 26-1 to 26-N. For example, data collection module 28 may collect the number of object, memory or register locks issued when tasks are running; the waiting time of synchronization performed within a task group; memory related features like the cache miss rate, and others. This data may be collected in nearly real time and data is collected to determine if it is desirable or necessary to throttle the processors in a subsequent processing cycle. Throttling comprises disabling or de-activating one or more threads to promote more orderly processing, to promote lower power consumption, or to otherwise improve performance of the system. The data collected by the data collection module 28 can reflect the real time resource and system status and comprises a precise input for the throttling decision.

[0018] In an example embodiment, the multithreaded system includes an assignment module assigning a subset of a plurality of tasks to a plurality of threads during a time N, a collection module collecting data during the time N concerning the operation of the plurality of threads, an analysis module analyzing the data to determine at least one condition concerning the operation of the plurality of threads during the time N, and an adjustment module adjusting a number of the plurality of threads available in time N + 1 in accordance with the at least one condition. In some embodiments, the multithreaded system may include other or additional modules for performing any one of or combination of steps described in the embodiments.

[0019] Figure 3 is a flow diagram of a throttling decision process according to an embodiment. The threads that are loaded are run in step 101. Data on the operation of the threads is collected in step 103. The data is then analyzed in step 105. This analysis is based on experience and previous experimental results, that is, the analysis is heuristic. If the results of the analyses pass a pre-set threshold, then the decision module in step 107 will output a positive step value k, which indicates that k more hardware threads will be enabled for next processing time interval. If the

decision module does not meet the pre-set threshold, then this module will output a negative step value k, which means that the number of hardware threads will be reduced by k in next processing time interval. In step 109, a throttling command based on the k determined in step 107 is issued. In other embodiments k may be increased when the heuristic analysis is below a threshold and vice versa. In yet other embodiments, the integer amount of k may be dynamically determined based on the heuristic analysis.

[0020] Throttling is the temporary reduction of the number of threads available for processing. Experiments show, in certain circumstances, that too many threads can hinder the operation of the overall processing system. For example, if the data collection module 28 detects a large overall number of cache misses, temporarily reducing the number of operating threads may actually speed the overall processing throughput. Other examples of relevant data are the time necessary to sync tasks from different threads (one thread needs the results of the other), the number of object locks (where only one thread can access the locked object), and the number of cache misses. In other circumstances, too many threads operating in a particular mode may consume too much power. This is particularly important for portable devices, such as smart phones.

[0021] Decision process 105 is further illustrated in Figure 4. In step 201, it is assumed that the threads are operating in time slot N. In step 203, the data from data collection module 28 is gathered. In step 205, the collected data is analyzed using a heuristic based process using methods such as linear programming or weighted sums of input data. In step 207, the result of this analysis is compared with one or more pre-set thresholds. For example, the preset threshold may be determined heuristically. In step 209, the decision is yes if the result of the data analysis is greater than the pre-set threshold and no if less than the pre-set threshold. In one embodiment a yes output would trigger a positive value for k, meaning that the number of operating threads is reduced by k. An output of no would trigger a negative value for k and the number of threads would increase by k. In other heuristics, a result above a threshold may trigger a negative k value, while below a positive. An additional type of heuristic provides a negative k if the result is lower than a first threshold, a zero k if the result is between the first threshold and a second threshold, or a positive k if the result is above the second threshold. The throttling value of k is applied to the threads during time N + 1. The operation of this embodiment is not limited to reducing or

increasing the number of threads by k. This method is chosen for its simplicity and operational speed. In addition, this embodiment is not limited in the type of heuristic used. Different system configurations may respond differently, and thus different heuristic analysis can determine that different data is applicable to the analysis. Any method of increasing or reducing the number of operating threads in response to the decision analysis is considered within the scope of this embodiment.

[0022] Figure 5 illustrates the operation of the embodiment of Figure 2 when the data analysis threshold in step 207 is greater than the threshold. In Figure 5, it is assumed that no threads were throttled in time N and that k = 3. In this case, for example, controller 24 would not assign any tasks to threads 26-1, 26-2 and 26-3 during time N + 1, as shown in Figure 5. In some embodiments there may be other methods of temporarily shutting down threads. For example, in devices where power conservation is more important than speed, the throttled threads may have power removed. However, this would reduce the overall speed of the device due to the power-up time required when the power is reapplied.

[0023] Although the description has been described in detail, it should be understood that various changes, substitutions and alterations can be made without departing from the spirit and scope of this disclosure as defined by the appended claims. Moreover, the scope of the disclosure is not intended to be limited to the particular embodiments described herein, as one of ordinary skill in the art will readily appreciate from this disclosure that processes, machines, manufacture, compositions of matter, means, methods, or steps, presently existing or later to be developed, may perform substantially the same function or achieve substantially the same result as the corresponding embodiments described herein. Accordingly, the appended claims are intended to include within their scope such processes, machines, manufacture, compositions of matter, means, methods, or steps.

WHAT IS CLAIMED IS:

1. A method for operating a multithread processing system comprising:

assigning, by a controller, a subset of a plurality of tasks to a plurality of threads during a time N;

collecting, by the controller, data during the time N concerning the operation of the plurality of threads;

analyzing, by the controller, the data to determine at least one condition concerning the operation of the plurality of threads during the time N; and

adjusting, by the controller, a number of the plurality of threads available in time N + 1 in accordance with the at least one condition.

2. The method of claim 1, wherein the data is a number of object locks issued.

3. The method of any of claims 1 to 2, wherein the data is a number of cache misses.

4. The method of any of claims 1 to 3, wherein the data is a task synchronization waiting time.

5. The method of any of claims 1 to 4, wherein the multithread system comprises a multicore processor.

6. The method of any of claims 1 to 5, wherein the multithread system comprises a multiprocessor system.

7. The method of any of claims 1 to 6, wherein the adjusting of the number of the plurality of threads available in time N + 1 is a fixed integer that may be positive or negative.

8. A multithread processing system controller comprising:

a storage system storing programming; and

a processor coupled to the storage system and executing the programming, wherein the programming configures the multithread processing system controller to;

assign a subset of the plurality of tasks to a plurality of threads during a time N;

collect data during the time N concerning the operation of the plurality of threads;

analyze the data to determine at least one condition concerning the operation of the plurality of threads during the time N; and

adjust a number of the plurality of threads available in time N + 1 in accordance with the at least one condition.

9. The multithread processing system of claim 8, wherein the data is a number of object locks issued.

10. The multithread processing system of any of claims 8 to 9, wherein the data is a number of cache misses.

11. The multithread processing system of any of claims 8 to 10, wherein the data is a task synchronization waiting time.

12. The multithread processing system of any of claims 8 to 11, wherein the multithread system comprises a multicore processor.

13. The multithread processing system of any of claims 8 to 12, wherein the multithread system comprises a multiprocessor system.

14. The multithread processing system of any of claims 8 to 13, wherein the adjusting of the number of the plurality of threads available in time N + 1 is a fixed integer that may be positive or negative.

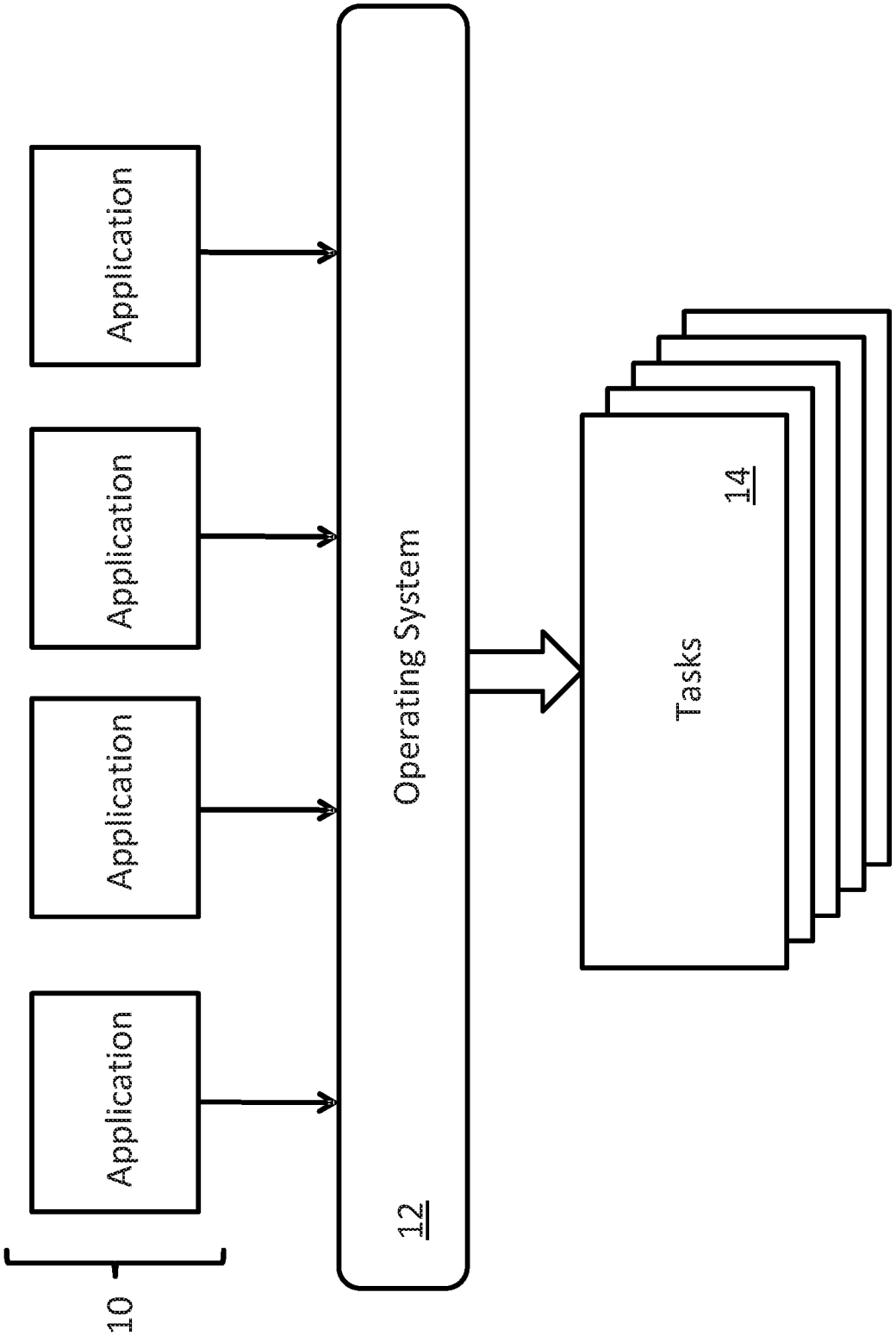
Application
Application
Application
Application
10
Operating System
12
Tasks
14

Figure 1

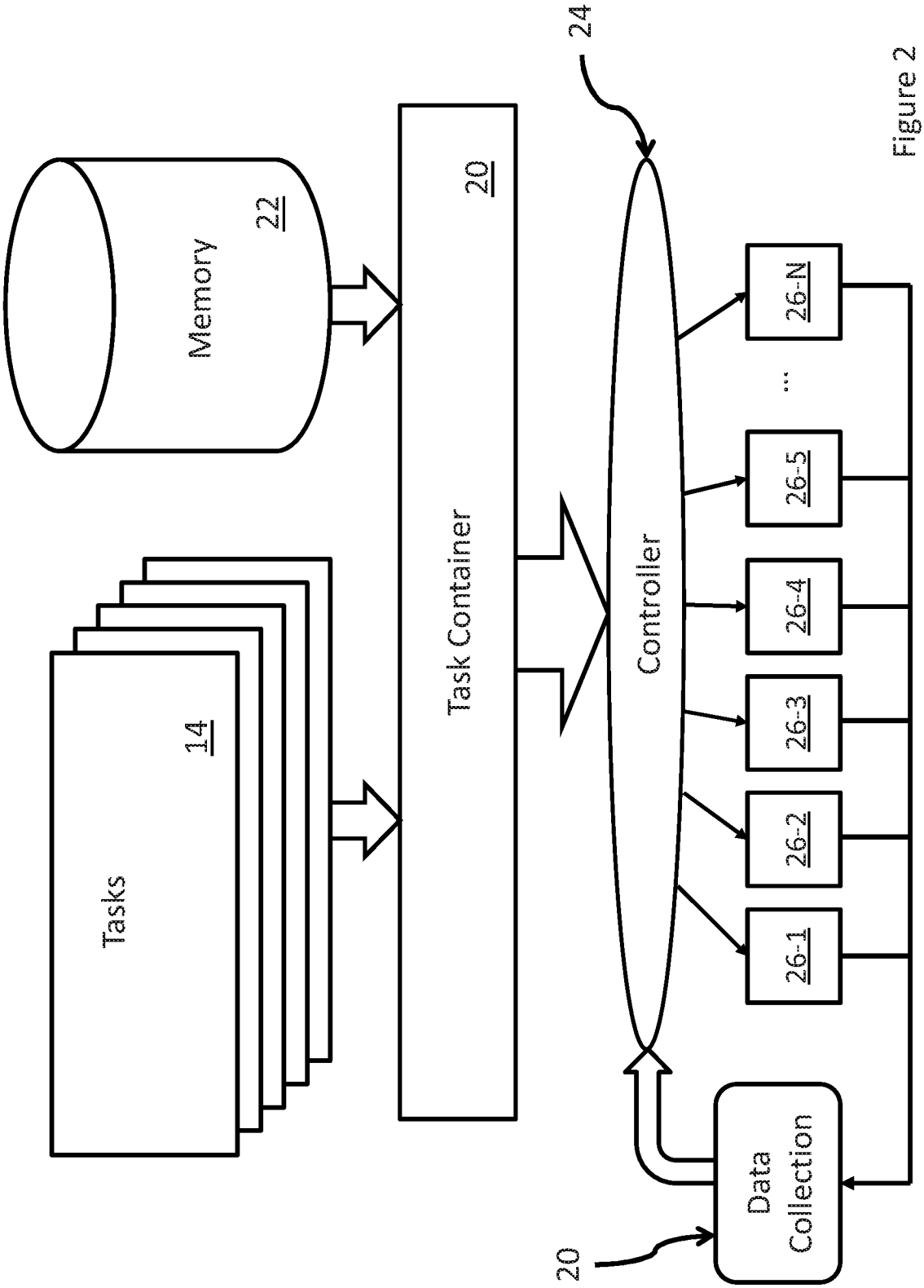
Memory
22
Tasks
14
Task Container
20
24
Controller
26-N
26-5
26-4
26-3
26-2
26-1
Data Collection
20

Figure 2

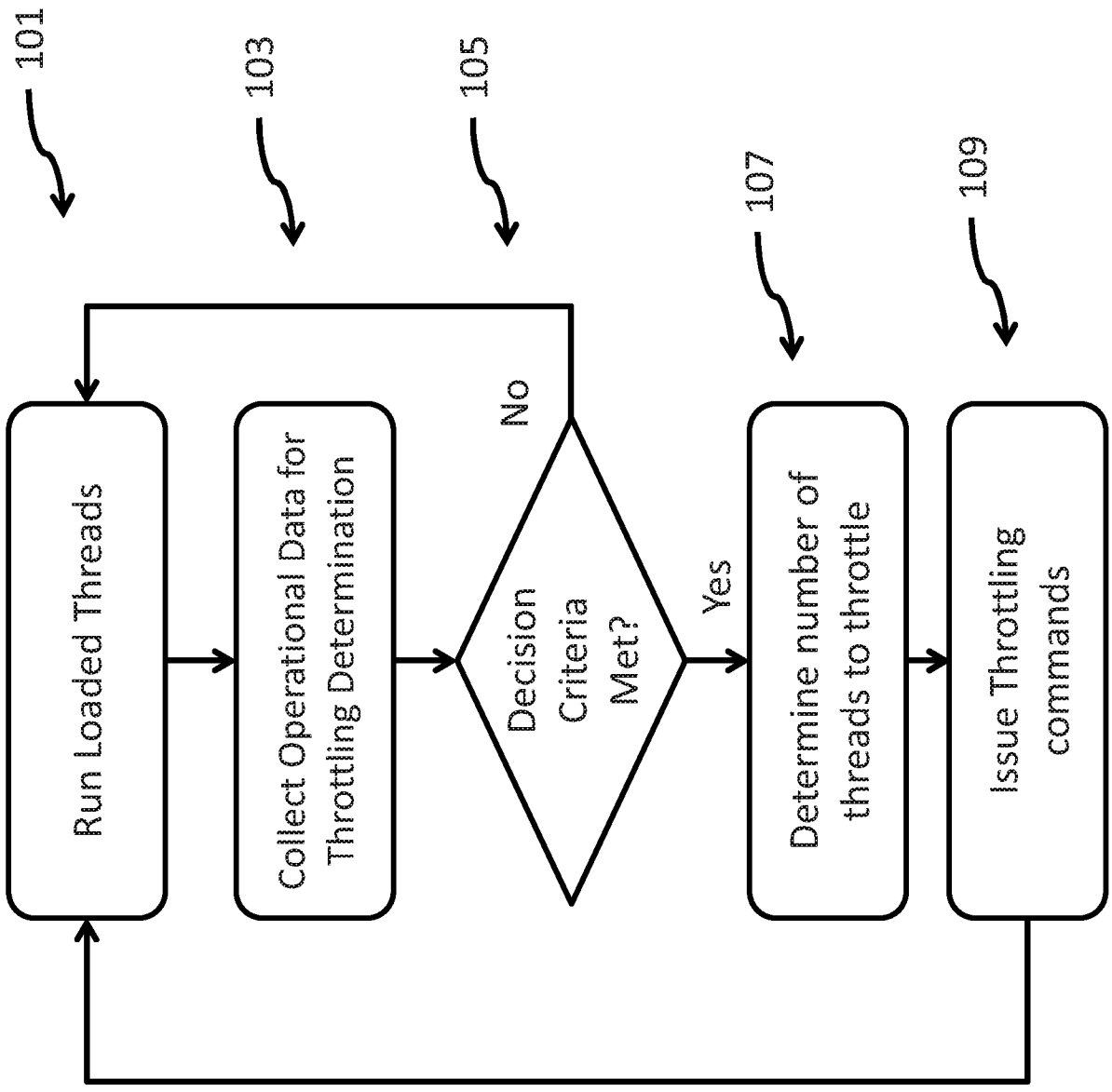
101
103
105
107
109
Run Loaded Threads
Collect Operational Data for Throttling Determination
Decision Criteria Met?
No
Yes
Determine number of threads to throttle
Issue Throttling commands

Figure 3

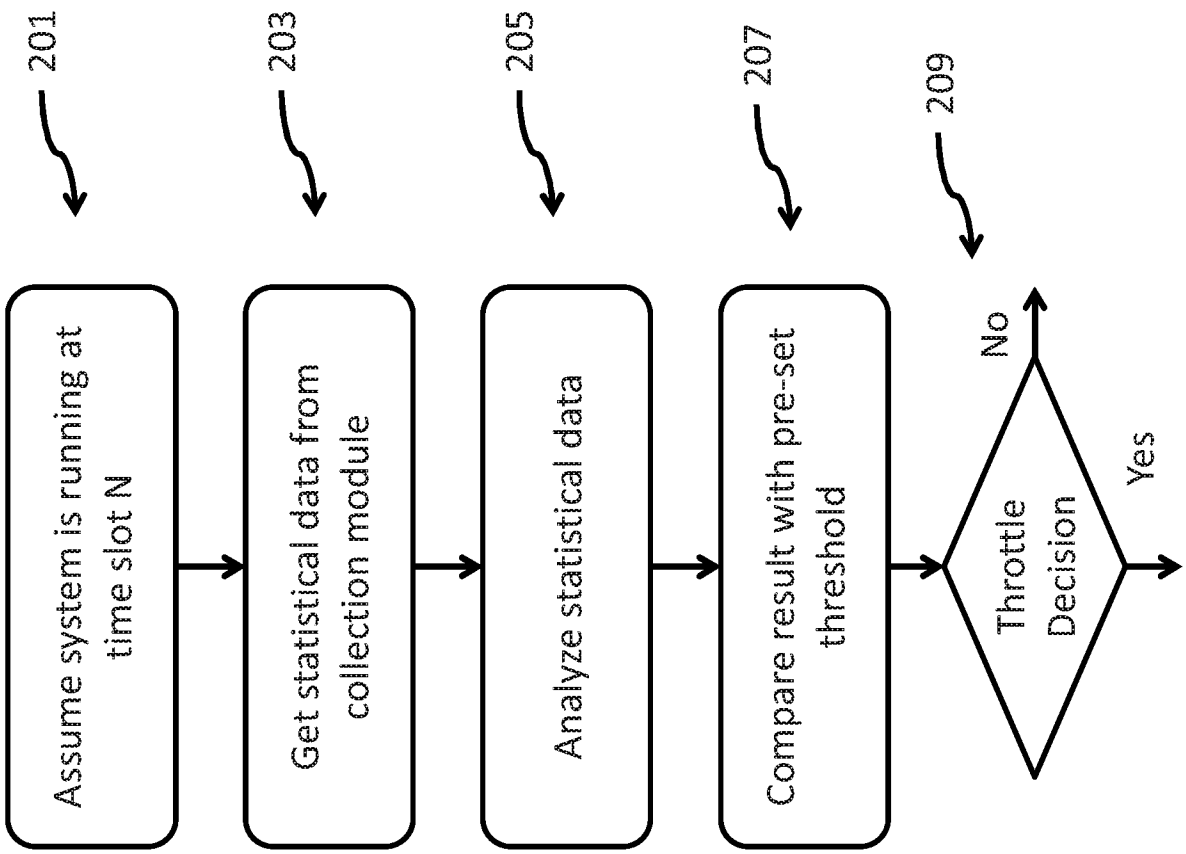
201
Assume system is running at time slot N
203
Get statistical data from collection module
205
Analyze statistical data
207
Compare result with pre-set threshold
209
Throttle Decision
No
Yes

Figure 4

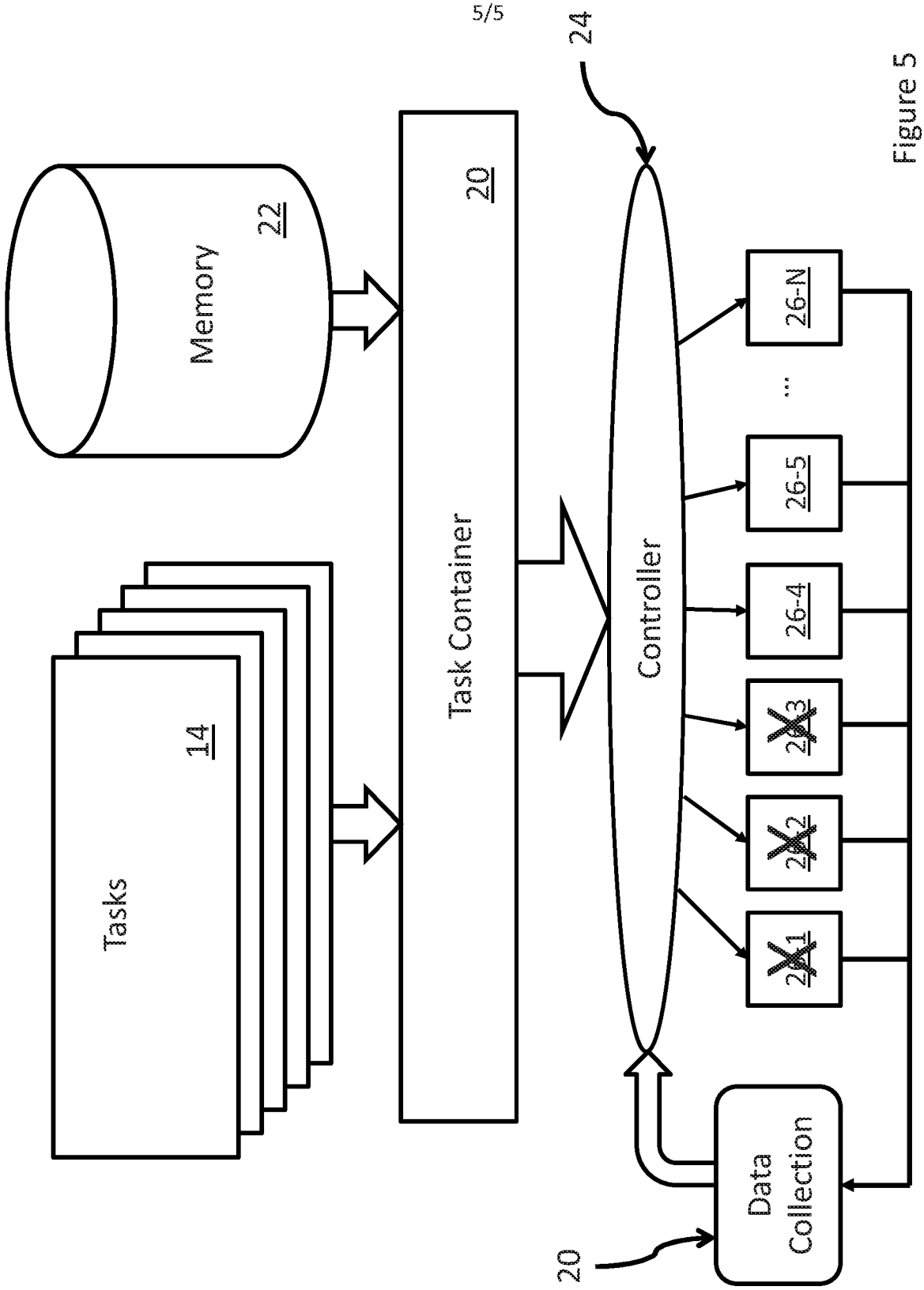
Memory
22
Tasks
14
20
Task Container
24
Controller
26-N
26-5
26-4
26-3
26-2
26-1
Data Collection
20

Figure 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2016/091893

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/46(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

SIPOABS, CNABS, DWPI, CNKI: multi, thread, task, assign, time, plurality, analyze, adjust, lock, cache, miss, synchronization, processor

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2015096031 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 02 July 2015 (2015-07-02) description, page 5, line 1 to page 9, line 25	1-14
A	CN 102629220 A (BEIJING DIGITAL CHINA SI-TECH INFORMATION TECHNOLOGY CO., LTD.) 08 August 2012 (2012-08-08) the whole document	1-14
A	US 9069564 B1 (NETLOGIC MICROSYSTEMS, INC.) 30 June 2015 (2015-06-30) the whole document	1-14

☐ Further documents are listed in the continuation of Box C.

☑ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 October 2016	**02 November 2016**

Name and mailing address of the ISA/CN	Authorized officer
STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088** **China**	**ZHANG,Yan**
Facsimile No. **(86-10)62019451**	Telephone No. **(86-10)62089421**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2016/091893

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
WO 2015096031 A1	02 July 2015	CN 105378652 A	02 March 2016
CN 102629220 A	08 August 2012	None	
US 9069564 B1	30 June 2015	None	