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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 August 2007 (30.08.2007)

PCT

(10) International Publication Number
WO 2007/098424 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/US2007/062356

(22) International Filing Date:
16 February 2007 (16.02.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/774,938 17 February 2006 (17.02.2006) US

(71) Applicant *(for all designated States except US)*: **QUALCOMM INCORPORATED** [US/US]; Attn: International Ip Administration, 5775 Morehouse Drive, San Diego, California 92121 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **JACOBS, Paul, E.** [US/US]; 9075 La Jolla Shores Lane, La Jolla, California 92037 (US). **SPRIGG, Stephen, A.** [US/US]; 14496 Southern Hills Lane, Poway, California 92064 (US).

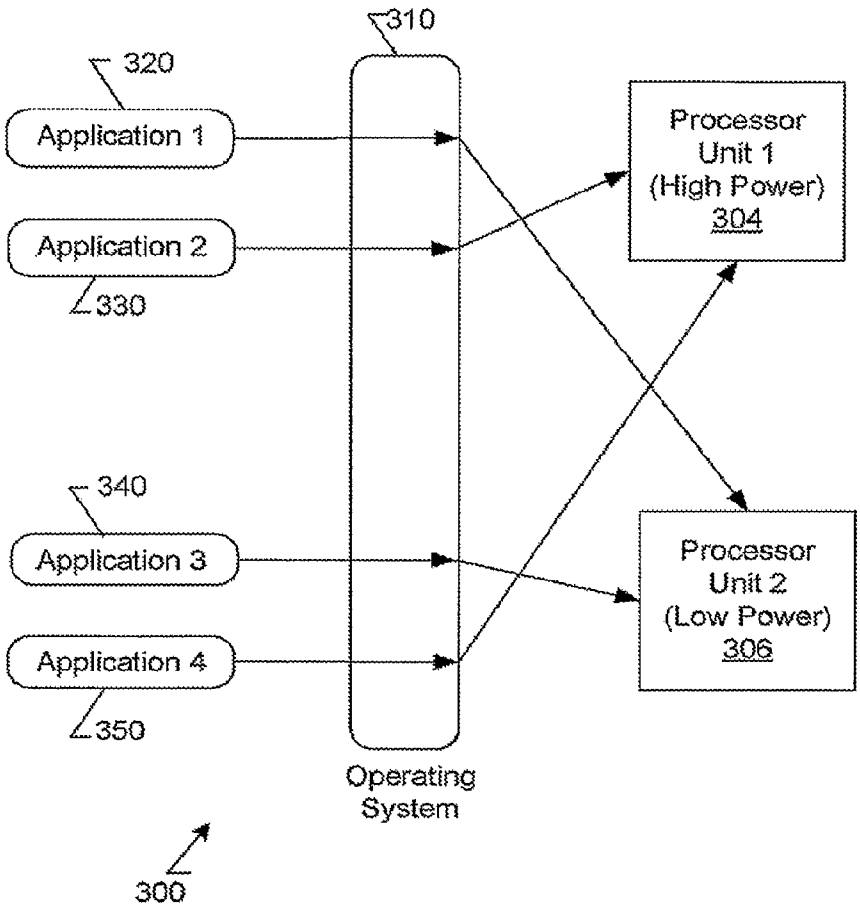
(74) Agent: **OGROD, Gregory, D.**; Attn: International Ip Administration, 5775 Morehouse Drive, San Diego, California 92121 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR MULTI-PROCESSOR APPLICATION SUPPORT



(57) Abstract: Described are methods and mechanisms for providing application support in a multi-processor system including receiving a request to execute an application, identifying a property specifying which processor from a plurality of processors to utilize to execute the application that is associated with the application, scheduling the application for execution on the specified processor based on the identified property, loading the application responsive to the scheduling of the application, and executing the application utilizing the specified processor.

WO 2007/098424 A2

ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, 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:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD FOR MULTI-PROCESSOR APPLICATION SUPPORT

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to Provisional Application No. 60/774,938, filed February 17, 2006.

BACKGROUND OF THE INVENTION

[0002] The invention relates generally to the field of power management for computing devices, and more particularly to power management on a multi-processor mobile device.

[0003] Consumers today make use of mobile devices, such as handheld cellular telephones, to perform very many different things. Conventional mobile devices can execute various types of software applications, and often include the capability of adding or installing new software products. Each of these software products requires a different level of system power to execute based, in part, on the amount of computation requirements the software product places on the processor that executes it. For instance, native user interface ("UI") applications often require much less computational effort than do multi-media applications or high-speed games.

[0004] Some mobile devices attempt to address this issue by using a two-processor design that includes a low-power processor and a high-power processor. Using this design, functions of the mobile device are segregated so that certain functions always use the low-power processor (e.g., code to operate the modem) and other functions always use the high-power processor (e.g., any installed applications). While this design ensures a high level of performance for the user, it also results in shorter battery life because the high-power processor is powered on to handle applications regardless of the level of computational effort needed.

[0005] An adequate system for supporting a multi-processor platform in a mobile device has eluded those skilled in the art, until now.

SUMMARY OF THE INVENTION

[0006] The invention is directed to a system and method for executing an application in a multi-processor system. Briefly stated, a multi-processor (e.g., dual processor) application specific integrated circuit ("ASIC") is provided that includes at least a low-power processor and a high-power processor. The ASIC is controlled by an operating system that is configured to support the execution of applications. When an application is loaded, the operating system evaluates a property associated with the application to determine on which of the multi-processors to execute the application. The application is then scheduled to execute on the appropriate processor.

[0007] In one aspect, a method for executing an application in a multi-processor system includes receiving a request to execute an application and identifying a property associated with the application. The property specifies which processor from a plurality of processors to utilize to execute the application. The method further includes scheduling the application for execution on the specified processor based on the identified property. The method additionally includes executing the application on the specified processor.

[0008] In another aspect, a mobile device includes a first processor, a second processor, at least one memory storage device in communication with the processors, and at least one computer-readable memory device which is readable by the processors. The computer-readable memory includes a series of computer-executable steps configured to cause the processors to receive a request at an operating system to execute an application; identify a property associated with the application, the property specifying which processor from a

plurality of processors to utilize to execute the application; schedule the application for execution on the specified processor based on the identified property; and execute the application utilizing the specified processor.

[0009] In yet another aspect, a computer readable medium storing a computer program for executing an application in a multi-processor system includes computer-readable code to receive a request to execute an application; computer-readable code to identify a property associated with the application, the property specifying which processor from a plurality of processors to utilize to execute the application; computer-readable code to schedule the application for execution on the specified processor based on the identified property; and computer-readable code to cause the application to be executed using the specified processor.

[0010] In another aspect, a system for executing an application in a multi-processor system includes means for receiving a request to execute an application; means for identifying a property associated with the application, the property specifying which processor from a plurality of processors to utilize to execute the application; means for scheduling the application for execution on the specified processor based on the identified property; and means for executing the application utilizing the specified processor.

[0011] In another aspect, a method for processing an application in a multi-processor environment includes receiving a request to process an application, identifying a property associated with the application, the property specifying which processor from a plurality of processors to utilize to process the application and processing the application. In one aspect, the application may be content data.

[0012] In another aspect, a mobile device includes a first processor, a second processor, at least one memory storage device in communication with the processors, and at least one computer-readable memory device which is readable by the processors. The computer-readable memory includes a series of computer-executable steps configured to cause the processors to receive a

request at an operating system to process an application; identify a property associated with the application, the property specifying which processor from a plurality of processors to utilize to process the application; and process the application utilizing the specified processor. In one embodiment, the application may be content data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Figure 1 is a functional block diagram generally illustrating a mobile device in which implementations of the invention are particularly applicable.

[0014] Figure 2 is a functional block diagram generally illustrating a system for providing multi-processor application support in accordance with an embodiment of the invention.

[0015] Figure 3 is a conceptual illustration of an operating system providing multi-processor application support in accordance with an embodiment of the invention.

[0016] Figure 4 is an operational flow diagram generally illustrating a process for providing multi-processor application support.

DETAILED DESCRIPTION

[0017] What follows is a detailed description of various techniques and mechanisms for power management in a mobile device. Very generally stated, the present invention is directed to determining which of a plurality of processors on which to execute an application based on a property of the application. Note that throughout this description, the term "application" is used for convenience and not intended to be limiting. For example, it will be recognized by those skilled in the art that "application" as used herein includes any function, task, content or other data sent to a processor for processing.

[0018] Figure 1 is a functional block diagram generally illustrating a sample mobile device 101 in which implementations of the invention are particularly

applicable. The mobile device 101 may be any handheld computing device, such as a cellular telephone, a personal digital assistant, a portable music player, a global positioning satellite (GPS) device, and the like. Although described here in the context of a handheld computing device, it should be appreciated that implementations of the invention may have equal applicability in other areas, such as laptop, desktop, or perhaps even server computing devices.

[0019] In this example, the mobile device 101 includes a multi-processor device 102, a memory 108, a storage medium 113, and a communication module 121 all coupled over a system bus 107. The multi-processor device 102 may be an Application Specific Integrated Circuit ("ASIC") that includes a first processor unit 104 and a second processor unit 106 encapsulated into a single unit. In another embodiment (not shown) each processor is implemented as a discrete unit.

[0020] In accordance with one embodiment of the present invention, one processor unit (e.g., processor unit 1 104) is more powerful than the other processor unit (e.g., processor unit 2 106). The more powerful processor is selected when more processing throughput is desired, such as may be more suitable for applications that are processor intensive. However, the less powerful processor consumes less power, and therefore results in more generous battery life and may be more acceptable for less process intensive applications (including tasks). Each of the processor units (processor unit 104 and 106) is a microprocessor or a special-purpose processor such as a digital signal processor (DSP), but may in the alternative be any conventional form of processor, controller, microcontroller, or state machine. In one example, the first processor unit 104 is implemented as a high-power microprocessor, and the second processor unit 106 is implemented as low-power DSP. Mobile device 101 may also include additional components known to those skilled in the art.

[0021] Multi-processor device 102 is coupled to the memory 108, which may be implemented as RAM holding software instructions that are executed by the

processor units 104 and 106. In this embodiment, the software instructions stored in the memory 108 include one or more applications 112 and an operating system 110. It is important to note that the memory 108 could be implemented as standalone RAM, or it could be integrated into the multi-processor device 102 with the processor units 104 and 106 to achieve improved efficiencies. The memory 108 may be composed of firmware or flash memory, such as a SmartMedia card. In another embodiment, operating system 110 includes software instructions enabling operating system 110 to determine which processor unit to use for execution of one or more applications from applications 112.

[0022] The storage medium 113 may be implemented as any nonvolatile memory, such as ROM memory, flash memory, or a magnetic disk drive, just to name a few. The storage medium 113 may also be implemented as any combination of those or other technologies, such as a magnetic disk drive with cache (RAM) memory, or the like. In this particular embodiment, the storage medium 113 is used to store data during periods when the mobile device 101 may be powered off or without power.

[0023] The communication module 121 enables bidirectional communication between the mobile device 101 and one or more other computing devices. Communications module 121 may include components to enable RF or other wireless communications, such as a cellular telephone network, Bluetooth connection, wireless local area network, or perhaps a wireless wide area network. Alternatively, communications module 121 may include components to enable land line or hard wired network communications, such as an Ethernet connection, universal serial bus connection, IEEE 1394 (Firewire) connection, or the like. These are intended as non-exhaustive lists and many other alternatives are possible.

[0024] Figure 2 is a functional block diagram illustrating in slightly greater detail the system memory 108 in which implementations of the invention are particularly applicable. As mentioned above, the system memory 108 includes an operating system 210 and one or more applications, such as first application

240 and second application 250. Each of those components will be described here in the context of the invention.

[0025] To illustrate the principles of the invention, the first application 240 requires greater processing power to perform acceptably, such as a multi-media application or a high-speed game. The second application 250 requires less processing power to perform acceptably, such as a native user interface module or the like. Each of the applications also has associated meta information such as a module information file (a "MIF" file), which includes salient details about the application such as its icon, title, and an enumeration of the privileges it requires in order to operate. Accordingly, the first application 240 as an associated MIF file 241, and the second application 250 has an associated MIF file 251.

[0026] Each respective MIF file further includes an identifier or property that either directly identifies on which of plural processors to execute the application, or includes identifying information that can be used to determine on which processor to execute the application (the "proc property"). For example, the first application 240 has a MIF file 241 including a proc property 242 that indicates the first application 240 requires greater processing power. Similarly the second application 250 has a MIF file 251 including a proc property 252 that indicates the second application requires less processing power and may function adequately on a low power processor. It should be noted that the actual form of the data in the property can take many forms. For example, it is envisioned that classes of processors may develop that generally categorize processors by their computational performance, by power consumption, or both. In this manner, the proc property may simply identify a minimum required processor rather than directly identifying a specific processor on which to execute the application.

[0027] The operating system (O/S) 210 is configured to organize and control the hardware and software of the mobile device. In this particular embodiment, the operating system 210 includes a scheduler 232 and a loader 235. The scheduler 232 manages the processes that may be executing on the mobile

device, and schedules processor time for each process or thread. In addition, the scheduler 232 is configured to determine which processor on which to execute a particular application by referring to the proc property of the application. To that end, the scheduler 232 is configured to read, directly or indirectly, meta information from the MIF file associated with the application to determine which processor on which to execute the application. The loader 235 is configured to load an application into a process and begin executing the application on a particular processor under control of the scheduler 232. In one embodiment, the scheduler 232 and the loader 235 operate in a kernel mode or protected mode of execution.

[0028] In operation, operating system 210 receives an instruction to execute the first application 240, such as via a system call from a shell. This instruction may have been initiated by a user selecting and activating an icon in a user interface, or the like. In this particular implementation, the operating system 210 then queries the local storage device to determine the location of the application 240. The operating system 210 reads the MIF file 241 for the application 240 to identify the execution environment for the application 240, such as which processor to be used in a multi-processor environment when executing the associated application. The operating system 210 may extract the proc property 242 from the MIF file 241 and pass it to the scheduler 232, along with an instruction to execute the application 240.

[0029] The scheduler then schedules the application to be loaded in the system memory 180 and executed on a particular processor, in accordance with the proc property of the application. The loader then loads the application into the identified memory locations. The ability of the scheduler 232 and operating system 210 to match an application with a processor increases overall system power utilization efficiency.

[0030] In other embodiments, when scheduler 232 reads properties (e.g. properties 242 and 252) from applications (e.g. first application 240 and second application 250, respectively), scheduler 232 may not match an application to a processor. For example, in a single processor configuration the scheduler 232

could simply ignore the proc property, and schedule the execution of the application in the conventional manner. Similarly, in a multi-processor configuration when the processor identified by the property is unavailable, scheduler 232 schedules the application based on other criterion, such as the most efficient use of the system processors. In yet another example, in a multi-processor configuration when an application does not identify a preferred processor, the scheduler 232 could schedule the application in the conventional manner, such as the most efficient use of the system processors.

[0031] Figure 3 is a graphical illustration of an operating system 310 scheduling the execution of each of four applications on each of two processors based on a property of the applications, in accordance with the invention. The exemplary application loading system 300 shown in Figure 3 may be implemented on mobile device 101, described above. Application loading system 300 is used to increase the overall efficiency of a computing system within which application loading system 300 operates. It does so by identifying and reading a property (described in Figure 2, above) associated with the application that instructs the operating system 310 to execute the application on a specified processor.

[0032] In Figure 3, application loading system 300 includes first processor 304, second processor 306, operating system 310, first application 320, second application 330, third application 340, and fourth application 350. The first processor 304, which may be a processor core on an ASIC, is a relatively-high performance and power processing unit. In contrast, the second processor 306, which may be another processor on the same ASIC as the first processor 304, is a relatively-low performance processing unit that consumes less power than the first processor 304.

[0033] In this example, first application 320 is a native user interface (U/I) application, second application 330 is a multi-media application such as for viewing .mpeg files or playing .wav files, third application 340 is a native cellular transmission and/or reception application, and fourth application 350 is a high-speed entertainment application such as a game. Accordingly, it has been

determined in advance that the first application 320 and the third application 340 will achieve acceptable levels of performance using a lower power processor. In contrast, it has also been determined that the second application 330 and the fourth application 350 require a higher power processor to achieve acceptable performance levels. Thus, the first application 320 and the third application 340 both include a property that identifies the second processor 306 as the preferred processor on which to execute those applications. Similarly, the second application 330 and the fourth application 350 both include a property that identifies the first processor 304 as the preferred processor one which to execute those applications.

[0034] In addition, if the processor specified by the property within an application is unavailable, the operating system 310 schedules the execution of the application based on other criterion. For example, to achieve adequate performance during a period when the low power processor is being highly utilized, the operating system 310 may schedule an otherwise low-power application to execute on the higher-power processor. Moreover, when the processor is not specified by the property within an application, operating system 310 may schedule the application based on other criterion, such as conventional load balancing or power considerations.

[0035] Figure 4 is an operational flow diagram generally illustrating a method 400 for providing support for an application to direct on which processor in a multi-processor system to execute. In one embodiment, method 400 is implemented with components of the exemplary operating environments of Figures 1-3. Preferably, one or more steps of method 400 are embodied in a computer readable medium containing computer readable code such that a series of steps are implemented when the computer readable code is executed on a computing device. In some implementations, certain steps of method 400 are combined, performed simultaneously or in a different order, without deviating from the objective of method 400.

[0036] At step 410, a request for execution of an application within a multi-processor system is received at an operating system. In one embodiment, the

operating system receives an instruction to execute an application, such as via a system call from the shell. In an example and referring to Figure 1 above, operating system 110 receives an instruction to execute an application within applications 112, such as a system call from the shell via media control component 111.

[0037] At step 420, a property associated with the application is identified. The property provides information that is used to determine on which processor within the multi-processor system to execute the application. In one embodiment, a property associated with the application is identified by a component of the operating system. In an example and referring to Figure 2 above, operating system 210 identifies first proc property 242 associated with first application 240 and passes the contents of the identified property to scheduler 232.

[0038] At step 430, the application is scheduled for execution on the specified processor based on the identified property. In one embodiment, the scheduler determines whether the identified processor is currently operating at an acceptable utilization to support the execution of the application. In an example and referring to Figure 1 and 2 above, scheduler 232 (within operating system 210 and 110) creates a process in which to execute the application 240, and then schedules that process for execution on the first processor 104.

[0039] At step 440, the application is loaded responsive to the scheduling of the application. In one embodiment, the loader loads the application into memory responsive to the scheduler scheduling the application to be loaded. In an example and referring to Figure 1 and 2 above, loader 235 loads application 240 into memory 108 responsive to scheduler 232 scheduling application 240 to be loaded.

[0040] At step 450, the application is executed using the specified processor. In one embodiment, the scheduler sets the stack pointer of the processor that was previously identified to the memory location containing portions of the application.

[0041] Advantageously, the system and techniques described above enable a mobile device having two processor cores on a single ASIC, where one processor core has a lower power consumption requirement than the other processor core, and where both processor cores share resources, such as RAM and operating system components. In addition, applications installed on the mobile device may be scheduled for execution on the processor that would result in an appropriate power and performance balance, rather than on a hard rule governing which applications run on which processors. These system and techniques result in improved battery life without necessarily sacrificing application usability.

[0042] While the present invention has been described with reference to particular embodiments and implementations, it should be understood that these are illustrative only, and that the scope of the invention is not limited to these embodiments. Many variations, modifications, additions and improvements to the embodiments described above are possible. It is contemplated that these variations, modifications, additions and improvements fall within the scope of the invention as detailed within the following claims.

CLAIMS

We/I Claim:

1. A method for executing an application in a multi-processor system, comprising:

receiving a request to execute the application;

identifying a property associated with the application, the property specifying which processor from a plurality of processors to utilize to execute the application;

scheduling the application for execution on the specified processor based on the identified property; and

executing the application utilizing the specified processor.

2. The method of claim 1, wherein the plurality of processors include a higher throughput processor and a lower throughput processor.

3. The method of claim 2, further comprising the step of assigning the property associated with an application based on the application's throughput processing need.

4. A method for processing an application in a multi-processor system, comprising:

receiving a request to process the application;

identifying a property associated with the application, the property specifying which processor from a plurality of processors to utilize to process the application; and,

processing the application utilizing the specified processor.

5. The method of claim 4, wherein the plurality of processors include a higher throughput processor and a lower throughput processor.

6. The method of claim 4, wherein the application is content data.

7. A mobile device comprising:

a communication bus;

a first processor;

a second processor;

at least one memory storage device in communication with the processors; and

at least one computer readable memory device which is readable by the processors, the computer readable memory device including a series of computer-executable steps configured to cause the processors to:

receive a request at an operating system to execute an application;

identify a property associated with the application, the property identifying which processor from a plurality of processors on which to execute the application;

schedule the application for execution on the specified processor based on the identified property; and

execute the application on the specified processor.

8. A mobile device comprising:

a communication bus;

a first processor;

a second processor;

at least one memory storage device in communication with the processors; and

at least one computer readable memory device which is readable by the processors, the computer readable memory device including a series of computer-executable steps configured to cause the processors to:

receive a request at an operating system to process an application;

identify a property associated with the application, the property identifying which processor from a plurality of processors on which to process the application; and

process the application on the specified processor.

9. The mobile device of claim 8, wherein the first processor has a different processing throughput than the second processor.

10. The mobile device of claim 8, wherein the application is content data.

11. A computer readable medium storing a computer program to execute an application in a multi-processor system, comprising:

computer readable code to receive a request to execute the application;

computer readable code to identify a property associated with the application, the property specifying which processor from a plurality of processors to use to execute the application;

computer readable code to schedule the application for execution on the specified processor based on the identified property;

computer readable code to load the application responsive to the scheduling of the application; and

computer readable code to execute the application utilizing the specified processor responsive to the loading of the application.

12. A computer readable medium storing a computer program to process an application in a multi-processor system, comprising:

computer readable code to receive a request to process the application;

computer readable code to identify a property associated with the application, the property specifying which processor from a plurality of processors to use to execute the application;

computer readable code to schedule the application for processing on the specified processor based on the identified property.

13. The computer readable medium of claim 12, wherein the application is content data.

14. A device for executing an application in a multi-processor system, comprising:

means for receiving a request to execute the application;

means for identifying a property associated with the application, the property specifying which processor from a plurality of processors to utilize to execute the application;

means for scheduling the application for execution on the specified processor based on the identified property; and

means for executing the application on the specified processor.

15. A device for processing an application in a multi-processor system, comprising:

means for receiving a request to process the application;

means for identifying a property associated with the application, the property specifying which processor from a plurality of processors to utilize to execute the application;

means for processing the application on the specified processor.

16. The device of claim 15, wherein the application is content data.

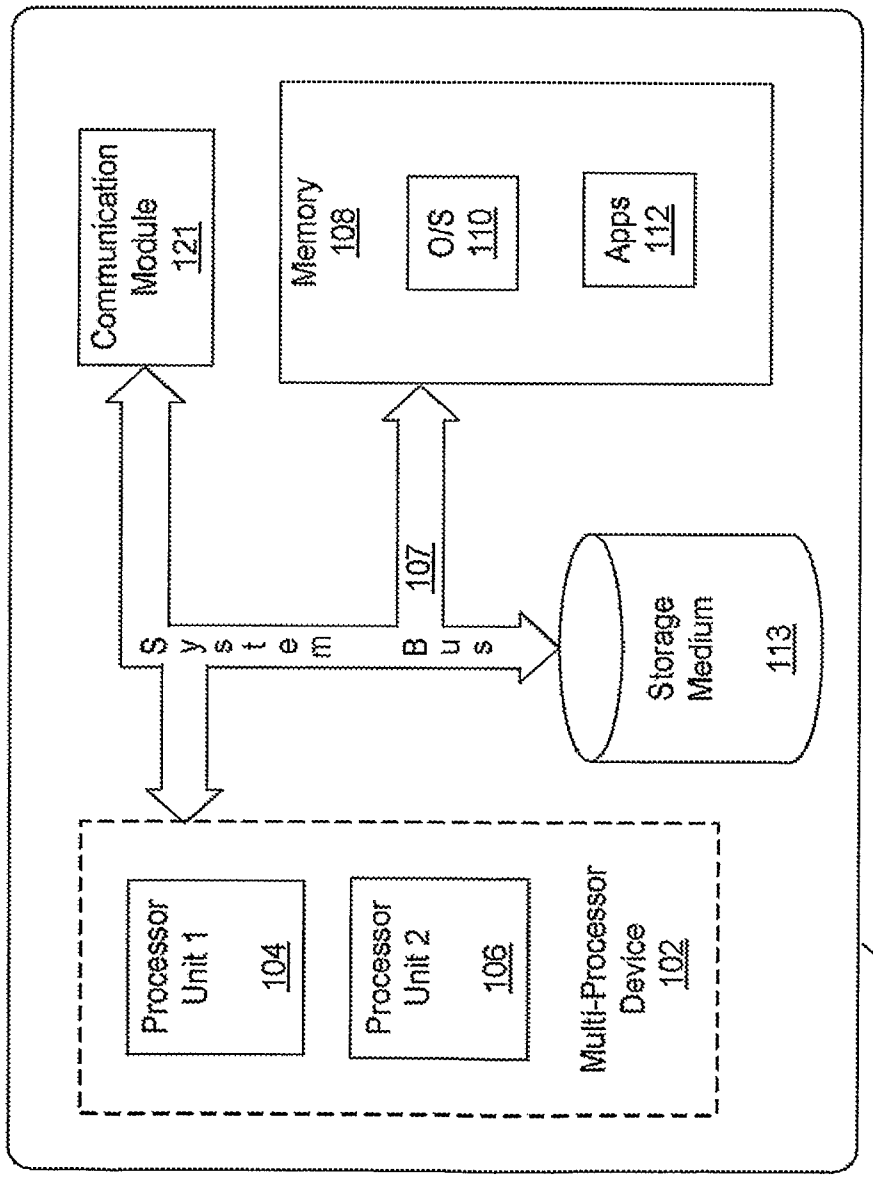

Mobile Device
101
Multi-Processor Device 102
Processor Unit 1 104
Processor Unit 2 106
System Bus 107
Storage Medium 113
Memory 108
O/S 110
Apps 112
Communication Module 121

Fig. 1

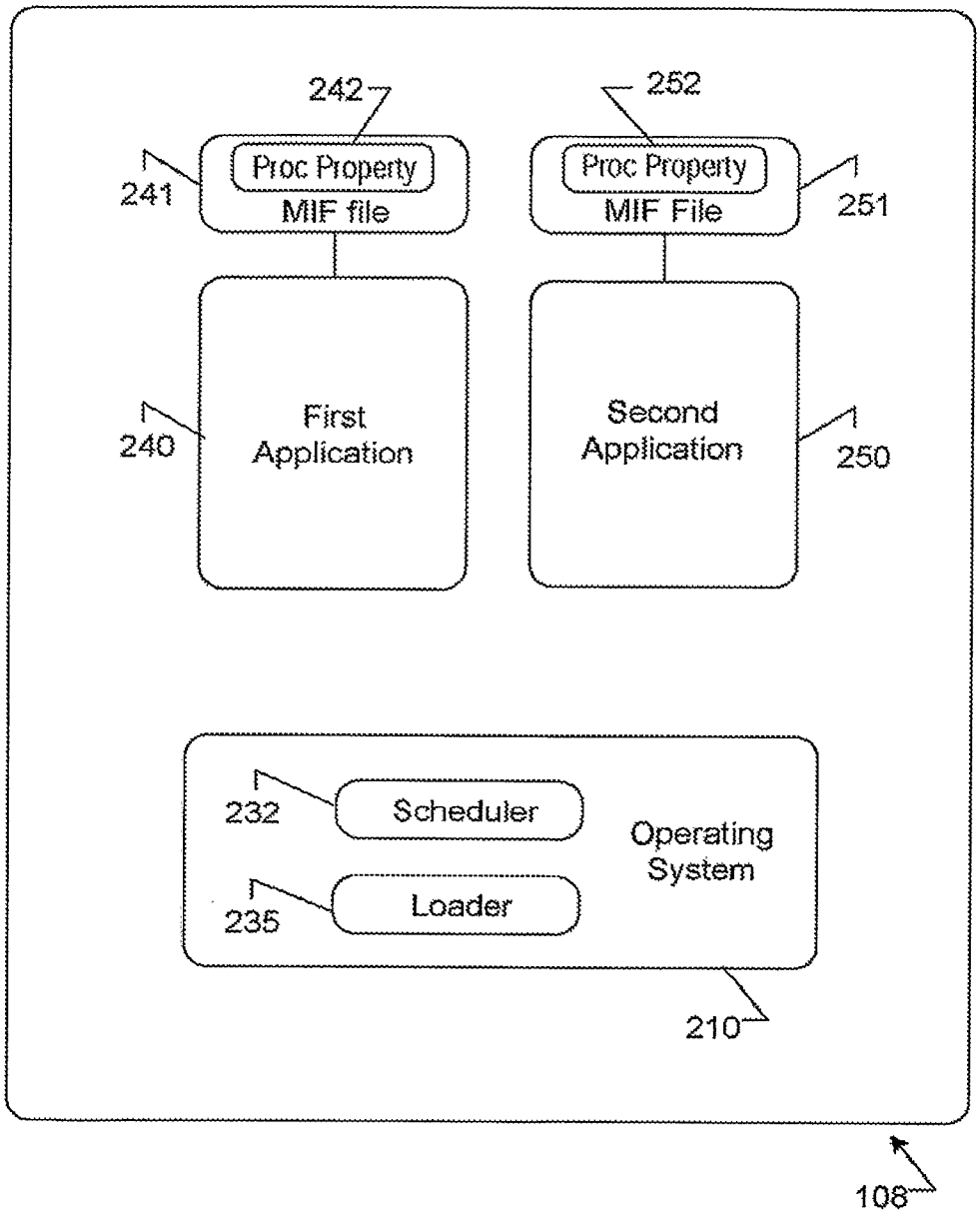

242
252
Proc Property
MIF file
Proc Property
MIF File
241
251
First
Application
Second
Application
240
250
232
Scheduler
Operating
System
235
Loader
210
108

Fig. 2

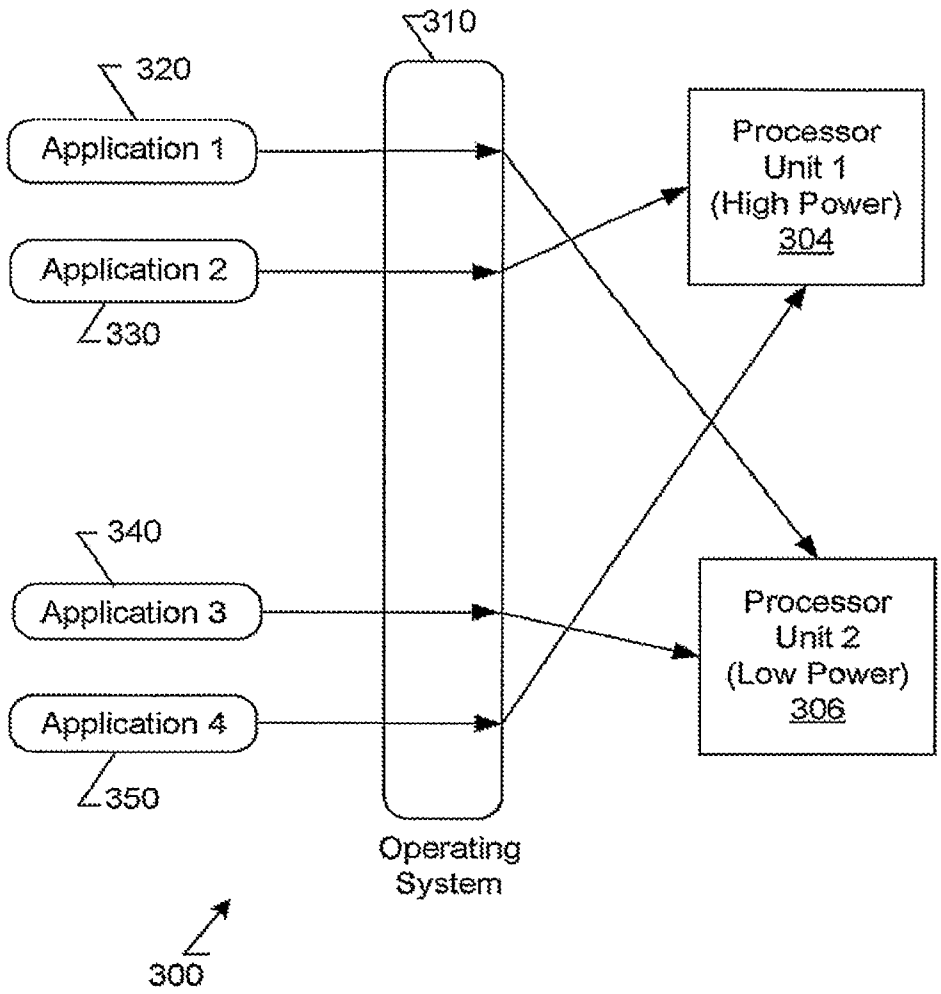

310
320
Application 1
Application 2
330
340
Application 3
Application 4
350
Operating System
Processor Unit 1 (High Power) 304
Processor Unit 2 (Low Power) 306
300

Fig. 3

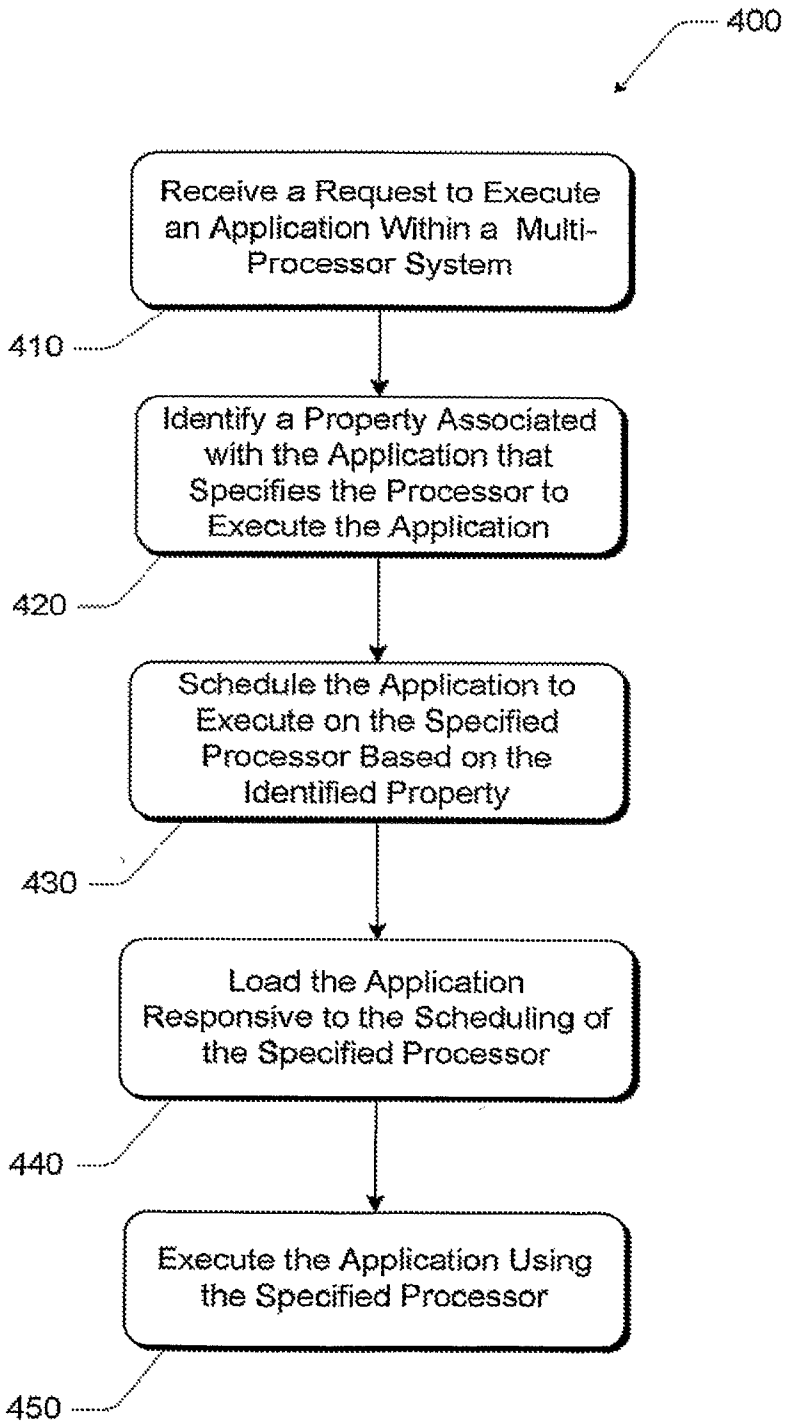

400
Receive a Request to Execute an Application Within a Multi-Processor System
410
Identify a Property Associated with the Application that Specifies the Processor to Execute the Application
420
Schedule the Application to Execute on the Specified Processor Based on the Identified Property
430
Load the Application Responsive to the Scheduling of the Specified Processor
440
Execute the Application Using the Specified Processor
450

Fig. 4