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

(19) World Intellectual Property Organization
International Bureau

PCT

(43) International Publication Date
4 October 2007 (04.10.2007)

(10) International Publication Number
WO 2007/109894 A1

(51) International Patent Classification:
H04L 29/02 (2006.01) *G06F 15/80* (2006.01)

(21) International Application Number: PCT/CA2007/000494

(22) International Filing Date: 28 March 2007 (28.03.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/743,848 28 March 2006 (28.03.2006) US
60/743,959 30 March 2006 (30.03.2006) US

(71) Applicant *(for all designated States except US)*: **RADISYS CANADA, INC.** [—/CA]; 4190 Still Creek Drive, Suite 300, Burnaby, British Columbia V5C 6C6 (CA).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **SALEEM, Adnan** [CA/CA]; 11296-163 Street, Surrey, British Columbia V4W 4P8 (CA). **CHUBBS, Alvin** [CA/CA]; 722 West 24th Avenue, Vancouver, British Columbia V5Z 2B8 (CA). **GUNN, Neil** [CA/CA]; 2587 Lund Avenue, Coquitlam, British Columbia V3K 6J8 (CA). **DAVIDSON, James** [CA/CA]; 15677-109th Avenue, Surrey, British Columbia V4N 4T5 (CA).

(74) Agents: **MITCHELL, Richard, J.** et al.; c/o Marks & Clerk, P.O. Box 957, Station B., Ottawa, Ontario K1P 5S7 (CA).

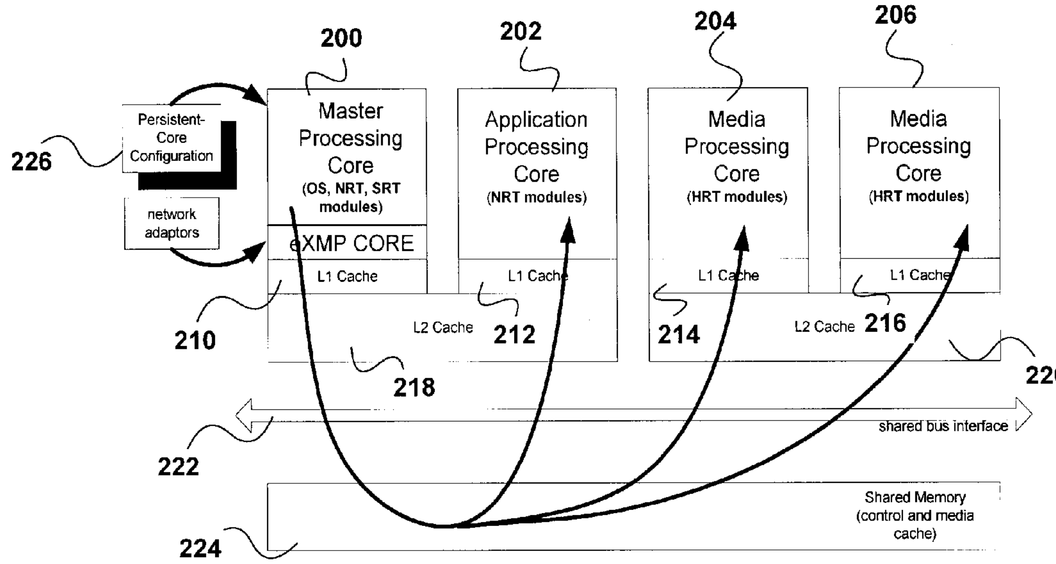
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, 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, 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, MT, 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).

Published:
— *with international search report*

[Continued on next page]

(54) **Title:** MULTIMEDIA PROCESSING IN PARALLEL MULTI-CORE COMPUTATION ARCHITECTURES



(57) **Abstract:** A media server for processing data packets comprising a plurality of processing cores (200, 202, 204, 206) for implementing media server functions, wherein said processing cores implement said media functions in modules categorized by real-time response requirements Modules may be categorized into hard real-time, soft real-time and near real time response requirements Hard real-time systems are used when it is imperative that an event is reacted to withm a strict deadline Soft real-time systems are typically those used where there is some issue of concurrent access and the need to keep a number of connected systems up to date with changing situations Near real-time could relate to a non real time operation and could also relate to an operation where there is some dependence on timely completion One of said processing cores may be a master core (200) providing configuration control and other services Shared memory may be used (224)

WO 2007/109894 A1

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

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.

MULTIMEDIA PROCESSING IN PARALLEL MULTI-CORE COMPUTATION ARCHITECTURES

FIELD OF THE INVENTION

[0001] This invention relates to the field of multimedia communications, and in particular to a media server, where the media server is intended for use with VOIP telephony networks, packet cable networks, PSTN telephony networks, wireless telephony networks, and all combinations thereof.

BACKGROUND OF THE INVENTION

[0002] Media servers are employed in telephony networks, and perform a variety of basic and enhanced services which include conferencing, audio and video interactive voice response (IVR), transcoding, audio and video announcements, and other advanced speech services. They may also be employed in networks which provide video conferencing services, as well as typical data exchange services of the sort which occurs over the Internet, over virtual private networks, within wide area networks and local area networks, and the like. In all cases, data exchange and processing performed by the media server is based on packet processing with fixed maximum processing time requirements.

[0003] Advances in hardware architectures, enabling multi-core and multi-processor computation, presents new challenges for software architectures and parallel processing algorithms. Incremental additions of parallel processing cores do not necessarily translate into equivalent linear increases in the amount of processing capacity. A given algorithm or processing task typically consists of some sub-sections which may be executed in parallel while other sub-sections must be executed in sequence. The relative proportion of serial and parallel execution sections governs the aggregate processing capacity of a system consisting of multiple parallel processing units. The relationship between the number of parallel processing units (N), proportion of parallelizable instructions (P), and the maximum speedup of the system is defined by Amdahl's Law.

[Amdahl's Law]

$$\text{MAXIMUM SPEEDUP} \le S + P / (S + P / N) \quad \text{or}$$

$$\text{MAXIMUM SPEEDUP} \le \frac{1}{S + (1 - S) / N}.$$

S = percentage of serial execution code

P = percentage of parallel execution code

N = Number of Processors

1 = S + P

[0004] Regardless of the total number of available processors (N), if all code sections require serial execution (S=1) then maximum speedup factor remains at 1. Theoretically, if all processing could be parallelized (P=1), then the maximum speedup is N, equivalent to the number of parallel processors available.

[0005] Amdahl's Law describes the general rules for taking advantage of multiple parallel processing cores for applications requiring large scale numerical data processing. However, the complex issues of real-time processing, deterministic response times, and load balancing across multiple cores are not addressed. These issues are essential for telecommunication applications.

SUMMARY OF THE INVENTION

[0006] Embodiments of the invention enable real-time sensitive media processing functions to be completed with deterministic response times, as required, utilizing multi-core and multi-processor hardware architectures.

[0007] Accordingly a first aspect of the invention provides a media server for processing data packets, comprising a plurality of processing cores for implementing media server functions, wherein said processing cores implement said media functions in modules categorized by real-time response requirements.

[0008] A second aspect of the invention provides a method of operating a media server for processing data packets, comprising providing modules categorized by real-time response requirements for performing said media functions; and implementing modules on a plurality of processing cores in accordance with said real-time response requirements.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The invention will now be described in more detail, by way of example only, with reference to the accompanying drawings, in which:-

[0010] Figure 1 is a block diagram of media server real-time processing modules;

[0011] Figure 2 is a block diagram of an asymmetric multi-core media server;

[0012] Figure 3 is a block diagram of a dynamic asymmetric multi-core media server; and

[0013] Figure 4 is a block diagram of a module symmetric multi-core media server.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0014] As illustrated in Figure 1, media server functions are comprised of multiple modules and layers 100, each having distinctly different processing requirements. The Media processing requests from network entities to the media server originate from control protocols, namely SIP, MGCP, Megaco (H.248), VoiceXML, and MSML.

[0015] The Session Control layer 102 manages the control protocol requests by allocating and initiating media processing performed by other modules in the media server. The media processing layer 106 performs packet processing in compliance with the RTP protocol, for all inbound and outbound media packets. Additionally, a media streaming module 109 provides functions for fetching and submitting media content from/to internal and external media storage systems, such as HTTP and NFS.

[0016] The underlying Operating System and system services, hardware device drivers, and operational configuration and management functions, including Operations, Administration, Maintenance and Processing (OAMP) module 107 provide services directly or indirectly, to all other modules within the media server.

[0017] The various modules 100 of the media server have distinctly different processing requirements from a perspective of real-time response. Although each module requires a deterministic response time, the real-time response requirements vary and are categorized into hard real-time (HRT), soft real-time (SRT), and near real-time (NRT) as shown in Figure 1. In this context, NRT is the least real-time constraining of the three classifications (hard realtime, soft realtime, and near realtime). NRT could relate to a non real time operation, but it could also relate to an operation where there is some dependence on its timely completion.

[0018] Generally, in computing terms, system is said to be real-time if the correctness of an operation depends not only upon the logical correctness of the operation but also upon the time at which it is performed. In a hard or immediate real-time system, the completion of an operation after its deadline is considered useless - ultimately, this may lead to a critical failure of the complete system. A soft real-time system on the other hand will tolerate such lateness, and may respond with decreased service quality (e.g., dropping frames while displaying a video). Hard real-time systems are typically found interacting at a low level with physical hardware, in embedded systems.

[0019] Hard real-time systems are used when it is imperative that an event is reacted to within a strict deadline. Usually such strong guarantees are required of systems for which not reacting in a certain window of time would cause great loss in some manner, such as physically damaging the surroundings or threatening human lives; although the strict definition is simply that missing the deadline constitutes complete failure of the system. In the context of multitasking systems the scheduling policy is normally priority driven pre-emptive schedulers.

[0020] Soft real-time systems are typically those used where there is some issue of concurrent access and the need to keep a number of connected systems up to date with changing situations. These can typically operate to a latency of seconds.

[0021] Three different multimedia processing embodiments utilizing multi-core hardware platforms are disclosed herein.

ASYMMETRIC MULTIMEDIA PROCESSING

[0022] Figure 2 shows an example of a media server based on a four processing core system. This embodiment has a dual-core dual-processor media server.

[0023] The asymmetric multimedia processing model for the media server, as shown in Figure 2, has four processing core modules 200, 202, 204, 206. The HRT modules 204, 206 are separate from the SRT and NRT modules 200, 202 for execution on independent processing cores. The number of cores dedicated to HRT modules and to SRT and NRT modules is a media server configurable option.

[0024] One of the set of processing cores 200 is identified as the master processing core and this provides the operating services, configuration control, and loading of the other available cores within the media server. The Master core also contains the eXMP Core Module 200, which controls system resource allocation and scheduling across all available processing cores. Each module is associated with its own L1 cache 210, 212, 214, 216, and pairs of modules are associated with respective L2 caches 218, 220.

[0025] The communication of control and media packets between the cores is based on a shared memory scheme or other IP based inter-core communication interfaces. In this example, the modules communicate over a shared bus interface 222 and have a shared memory 224.

[0026] The number of available processing cores and media servers consisting of multiple processing cores and adjunct DSP processors are managed by the eXMP Core Module 200. The allocation of cores or DSPs is based on pre-set system configuration and cannot be altered during run-time. However, a re-configuration allows the number of cores dedicated for HRT modules and the number of cores dedicated for SRT and NRT modules to be changed.

[0027] The allocation of available cores for HRT, SRT, and NRT functions can be modified using the OAMP module 107 of the media server shown in Figure 1. Updates to the allocation scheme are stored in persistent storage 226. Upon media server start-up, the pre-configured processing modules are instantiated on the available cores as identified by the configuration data.

[0028] The eXMP Core module 200 provides the following services to the multi-core media server.

1. Identifies the Master Processing Core at system start-up.

2. Loads the available cores with the appropriate media processing functions according to the assigned core allocation scheme from the persistent configuration data.
3. Provides a control and media communication interface between the available cores using shared memory or IP based interfaces.
4. Provides a direct and efficient network packet flow from the network adaptors to the appropriate media processing core.
5. Provides a layer above the OS and services such that the HRT processing cores are not unnecessarily interrupted, hence effectively separating the OS, NRT, and SRT functions from the HRT cores.
6. Manages the allocation of HRT media processing resources across the available HRT cores to provide load balancing.

[0029] Some of the Media Processing objects constrained to the HRT assigned cores are:

1. RTP Input
2. RTP Output
3. Audio and Video Decoder
4. Audio and Video Encoder
5. Audio Mixer
6. Video scaling and screen splitting
7. Audio Gain and Automatic Gain Control
8. In-band and Out-of-band tone detection including DTMF
9. In-band and Out-of-band tone generation including DTMF

10. Voice Activity Detector
11. Silence suppression
12. Echo cancellation
13. Video and audio announcement streaming
14. Hot-Word voice recognition
15. Voice Activated Video Switching
16. Audio and Video Recording

[0030] Each of the above media processing objects and variations of the objects has a quantifiable instruction count and a specific real-time response requirement. These processing objects remain idle until invoked, at which point the processing starts with a fixed maximum amount of latency and a fixed maximum amount of processing until completion. The processing requirement of these objects is unaffected by any other concurrent operation performed by the media server as only the media processing objects run on the cores assigned to HRT tasks.

[0031] The Session Control module and Media Processing Control and Management module collectively break higher level protocol service requests into a set of time-sequenced tasks for media processing objects.

[0032] These higher level protocol service requests include:

1. Create RTP stream to an end terminal
2. Play audio or video announcements
3. Start / stop audio or video recordings

4. Detect DTMF digits
5. Create a conference
6. Join users to a conference
7. Apply gain or DTMF clamping

[0033] The Media Processing Control and Management software models the processing time used by each media processing object for each assigned task. In this model, each DSP core is assigned a specific number of available CPU units. Each task required by a media processing object uses up a specific number of units.

[0034] The Media Processing Control and Management software uses the modelled load to determine which DSP core to assign new services to.

[0035] The Media Processing Control and Management software rejects requests for additional services when there are insufficient CPU units available to fulfill the request.

[0036] The Media Processing Control and Management software receives continuous feedback on the current number of CPU units used by each core. This feedback is used to dynamically adjust the level at which incoming requests are rejected.

[0037] The media processing objects use only a minimal set of OS services so that the modelled CPU load is not affected by OS service processing time.

[0038] The media processing software is constructed as a single OS thread so that the modelled CPU load is not affected by OS scheduler processing time.

DYNAMIC ASYMMETRIC MULTIMEDIA PROCESSING

[0039] The dynamically asymmetric multimedia processing design model, illustrated in Figure 3, separates the HRT modules 302, 306 from the SRT and NRT modules 300, 304, as in the case of asymmetric processing model described with reference to Figure 2. However, this embodiment enables the dynamic detection of system utilization and allocation of media processing cores accordingly. Mixing of HRT functions with NRT or SRT functions is not permitted in this embodiment in order to ensure that deterministic real-time responses required by the HRT functions can be guaranteed.

[0040] In this embodiment, the eXMP Core module 308 monitors the amount of core processor utilization and load levels during run-time. Based on this information, the eXMP Core module 308 can adjust the allocation or reservation of cores for HRT, SRT, and NRT functions. As with the previous embodiment, the Master Processor Core 300 remains static, while other available core functions may be dynamically adjusted.

[0041] A dual-core system is not possible in this embodiment because the mixing of the HRT modules with the NRT and SRT is not possible. However, for a media server which consists of four or more cores, the dynamic load balancing of HRT, SRT, and NRT modules is accomplished by the eXMP Core module 308, without the need for system reconfiguration.

[0042] During system start-up and initialization, the eXMP Core module performs core allocation according to the persistent configuration data stored in memory 326.

[0043] The frequency of adjustments to core process assignment takes into account the system cost in terms of processing latency. If such a penalty does not allow a re-adjustment, then the core process allocation scheme is not adjusted. This mechanism prevents thrashing, which can lead to lower overall system performance.

[0044] In order to instantiate a functional process on a given core as quickly as possible, the allowable processes are invoked on the core during system start-up. Each such process is set in the idle state until allocated and assigned by the eXMP Core module, at which time it is available for receiving events to perform the required actions.

[0045] A run-time model of current processing load on each core is managed by the eXMP Core module 308. Each core manages and updates its performance utilization information within a shareable section of memory. The eXMP Core module 308 periodically reads this information to determine if any adjustments are required to the current process assignments to the available cores. The scheduling and assignment ensures that NRT and SRT functions are prevented from execution on a given core concurrently with HRT functions.

The eXMP Core module provides the following services to the media server:

1. Initiate system start-up core allocation based on media server configuration, as to assignment of HRT, SRT, and NRT functions.
2. Loads each processing core with allowable HRT, SRT, and NRT functions and sets the associated states to Idle and ready state. This mechanism prevents un-necessary

process setup and switching times when re-assignment is required. At time of re-assignment, the associated process state is changed from Idle to Active or vice-versa.

3. Implements algorithms based on several factors which model resource utilization of each processing core. The factors contributing to processor core resource utilization are amount of CPU consumption, length of queued requests, and the average latency or response times.
4. Based on the core processor utilization model, re-adjusts or re-assigns a processing core for HRT or non-HRT functions.
5. Implements mechanisms to prevent thrashing a core processor from toggling between HRT functions and non-HRT functions.

MODULAR SYMMETRIC MULTIMEDIA PROCESSING

[0046] In a third embodiment, shown in Figure 4, based on a modular symmetric multimedia processing design, the execution of HRT, SRT, and NRT modules concurrently on the same core or sets of cores in a multi-core media server is permitted. Symmetric processing in this embodimentel is intentionally designed for coarse-grained parallel execution.

[0047] The HRT modules 404, 406 along with SRT and NRT modules 400, 402 are scheduled for execution on the same processing core, based on the current load level of the core, resource utilization level of the core, and the processing real-time deadline associated with the task to be scheduled.

[0048] Measurements of load levels on the processing cores are managed by each core and written to a common shareable memory location. The Master Processing Core 400 executes a load scheduling algorithm as part of the eXMP Core module 408. The eXMP Core scheduler 408 prevents the operating system scheduler from scheduling tasks directly to the cores, which may lead to undeterministic behavior.

[0049] An execution priority scheme allows the eXMP Core module 408, which includes a core scheduler, to determine the target core for execution as well as the priority order for processing within the core. This mechanism allows a given core to process queued requests in the order of processing priority based on the real-time requirements of the request.

[0050] The scheduling of worker threads on the available cores is under the control of the eXMP Core scheduler 408; however, threads on the Main Processor Core 400 are managed and processed by the Operating System scheduler.

[0051] The eXMP Core scheduler 408 provides scheduling of media processing tasks (HRT) without much involvement from the standard operating system services and TCP/IP stack. The media data packets are assembled and scheduled for processing by the core as determined by the eXMP Core scheduler. Media and control data copying is minimized by the use of shared memory regions which are accessible by the Main Processing Core 400 as well as the individual core which is scheduled for processing the content of the queued requests. The resource utilization measurements used by the eXMP Core scheduler 408 are specific to media processing objects with quantifiable and deterministic response time requirements.

[0052] The eXMP uScheduler 428 manages the scheduling requirements on a specific core. It enables concurrent execution of threads on the same processing core, under strict real-time controls. A standard operating system typically does not provide hard real-time controls required by media processing functions. The eXMP uScheduler 428 provides higher priority and service time for HRT module request over any other currently executing NRT and SRT modules on the same core.

Load Balancing

[0053] The eXMP cores 428 monitor and measure the CPU utilization, queue lengths, and response times periodically. The measured results are periodically updated in a common shared memory area, which is made accessible to the eXMP Core module 408 on the Master Processing Core 400. The eXMP core 408 schedules new requests for the individual cores based on the processing measurements. The requests objects are queued for processing by the individual cores in a shared memory region. An algorithm for load balancing, within the real-time response constraints, is implemented and managed within the eXMP Core module providing a centralized load balancing mechanism.

Cache Coherency

[0054] In order to avoid or minimize cache incoherency, the HRT, SRT, and NRT modules are designed with minimum contention for same regions of memory data structures. Data access locking via spin locks and other mutual exclusion mechanisms are minimized in the core processing modules as well as in the eXMP Core scheduling modules.

Redundancy

[0055] The eXMP Core includes a heart-beat protocol which allows detection of core failures due to hardware or software fatal errors. Since this design implements a shared memory mechanism, the transactional and operational states of modules executing on core are preserved over a core failure. The eXMP Core module 408 utilizes this mechanism in switching the processing from the failed core to a redundant core by re-mapping the shared memory, containing the relevant state information, to be utilized by the redundant core.

Claims:

1. A media server for processing data packets, comprising:

a plurality of processing cores for implementing media server functions, wherein said processing cores implement said media functions in modules categorized by real-time response requirements.

2. A media server as claimed in claim 1, wherein said modules are categorized into hard real-time (HRT), soft real-time (SRT), and near real time (NRT) response requirements.

3. A media server as claimed in claim 2, wherein subsets of said processing cores being dedicated to said HRT, SRT and NRT modules.

4. A media server as claimed in claim 3, wherein the number of processing cores in each of said subsets of processing cores is configurable.

5. A media server as claimed in claim 4, wherein one of said processing cores serves as a master core for providing operating services, configuration control and loading of available cores within the media server.

6. A media server as claimed in claim 5, wherein said master core includes a control core module for allocating and scheduling system resources across said processing cores.

7. A media server as claimed in claim 5, wherein the HRT modules are separated from the SRT and NRT modules for independent processing.

8. A media server as claimed in claim 8, wherein the control core module is configured to monitor the amount of core processor utilization and load levels during runtime to dynamically adjust the allocation of processing cores for said HRT, SRT, and NRT functions.

9. A media server as claimed in claim 5, wherein said master core is configured to schedule said HRT, SRT and NRT modules for execution concurrently on the same core or sets of cores.

10. A media server as claimed in claim 9, wherein said master core module is configured to implement an execution priority scheme to determine the target core for execution.

11. A method of operating a media server for processing data packets, comprising:

providing modules categorized by real-time response requirements for performing said media functions; and

implementing modules on a plurality of processing cores in accordance with said real-time response requirements.

12. A method claimed in claim 11, wherein said modules are categorized into hard real-time (HRT), soft real-time (SRT), and near real time (NRT) response requirements.

13. A method as claimed in claim 13, wherein subsets of said processing cores are dedicated to said HRT, SRT and NRT modules.

14. A method claimed in claim 13, wherein the number of processing cores in each of said subsets of processing cores is configurable.

15. A method as claimed in claim 4, wherein one of said processing cores serves as a master core for providing operating services, configuration control and loading of available cores within the media server.

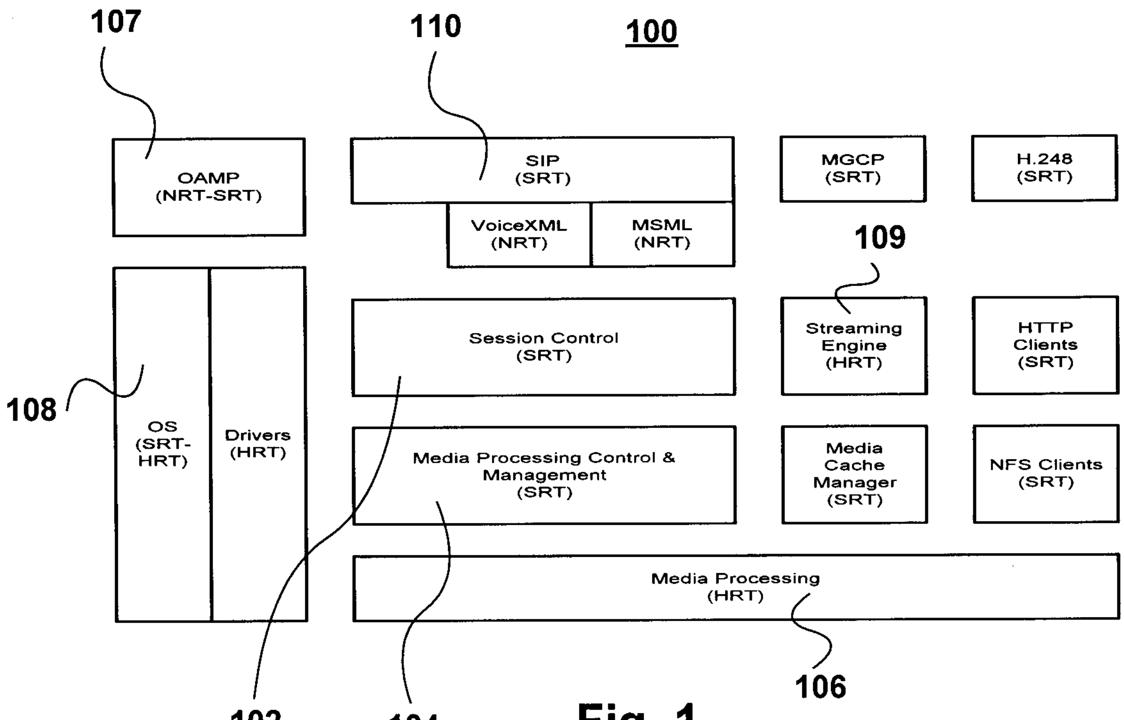
16. A method as claimed in claim 15, wherein a control core module in said master core allocates and schedules system resources across said processing cores.

17. A method as claimed in claim 15, wherein the HRT modules are separated from the SRT and NRT modules for independent processing.

18. A method as claimed in claim 8, wherein the control core module monitors the amount of core processor utilization and load levels during runtime to dynamically adjust the allocation of processing cores for said HRT, SRT, and NRT functions.

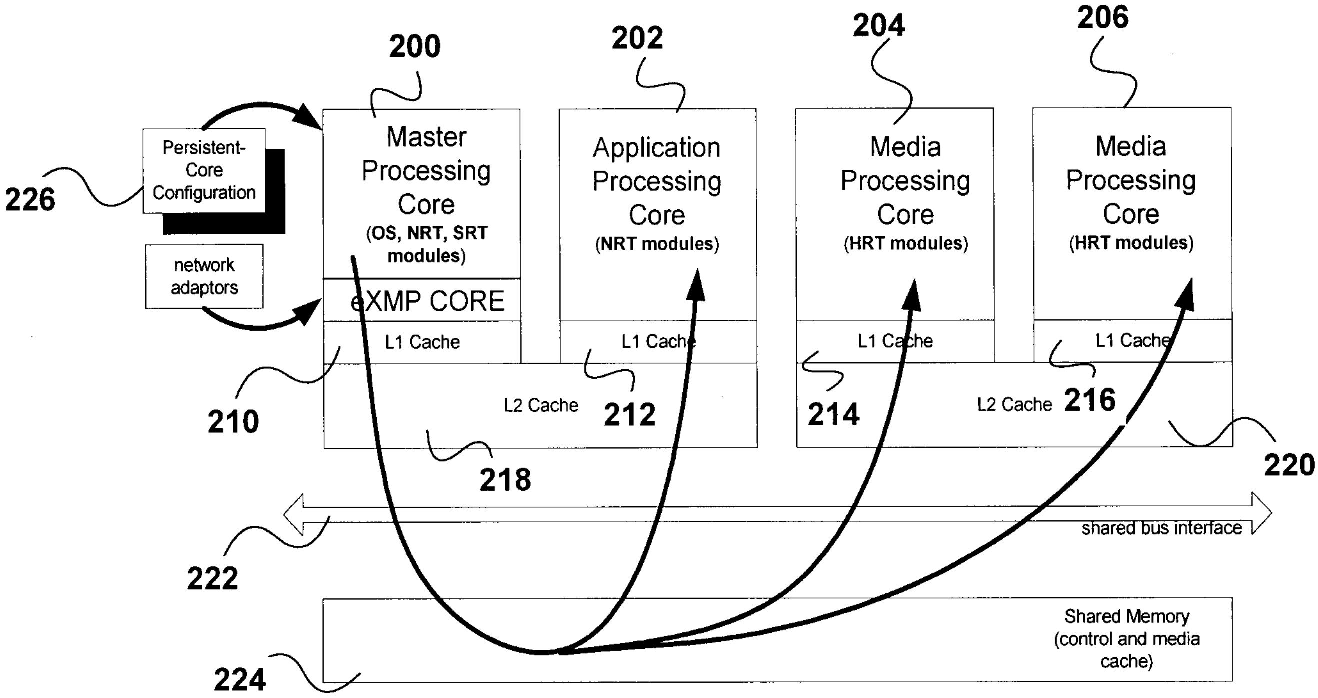
19. A method as claimed in claim 15, wherein said master core schedules said HRT, SRT and NRT modules for execution concurrently on the same core or sets of cores.

20. A method as claimed in claim 19, wherein said master core module is configured to implement an execution priority scheme to determine the target core for execution.



100
107
110
OAMP (NRT-SRT)
SIP (SRT)
VoiceXML (NRT)
MSML (NRT)
MGCP (SRT)
H.248 (SRT)
109
108
OS (SRT-HRT)
Drivers (HRT)
Session Control (SRT)
Streaming Engine (HRT)
HTTP Clients (SRT)
Media Processing Control & Management (SRT)
Media Cache Manager (SRT)
NFS Clients (SRT)
Media Processing (HRT)
106
102
104

Fig. 1



200
202
204
206
226
Persistent-Core Configuration
network adaptors
Master Processing Core (OS, NRT, SRT modules)
eXMP CORE
Application Processing Core (NRT modules)
Media Processing Core (HRT modules)
Media Processing Core (HRT modules)
L1 Cache
L1 Cache
L1 Cache
L1 Cache
L2 Cache
L2 Cache
210
212
214
216
218
220
shared bus interface
222
Shared Memory (control and media cache)
224

Fig. 2

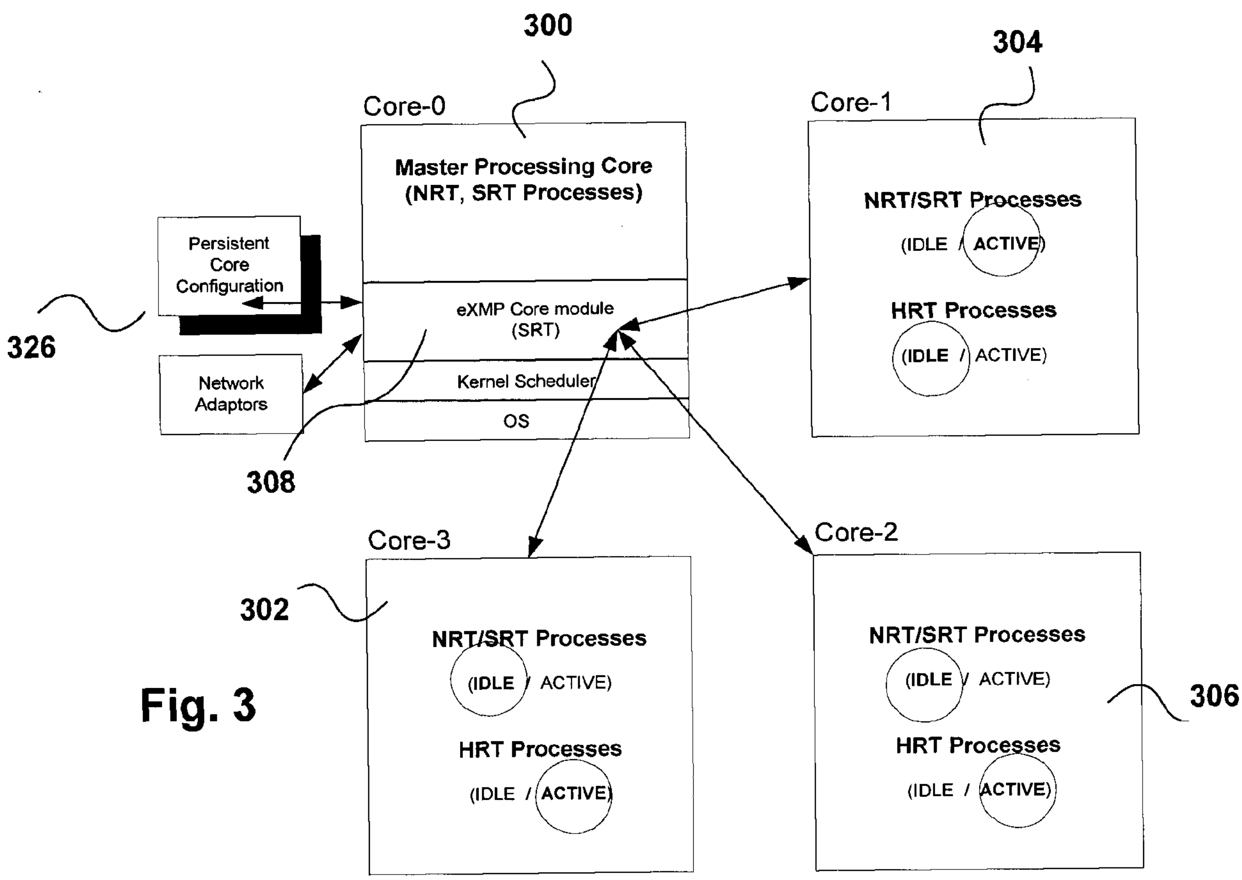

300
304
Core-0
Master Processing Core
(NRT, SRT Processes)
Core-1
NRT/SRT Processes
(IDLE / ACTIVE)
HRT Processes
(IDLE / ACTIVE)
Persistent Core Configuration
eXMP Core module
(SRT)
326
Network Adaptors
Kernel Scheduler
OS
308
Core-3
Core-2
302
NRT/SRT Processes
(IDLE / ACTIVE)
HRT Processes
(IDLE / ACTIVE)
NRT/SRT Processes
(IDLE / ACTIVE)
HRT Processes
(IDLE / ACTIVE)
306

Fig. 3

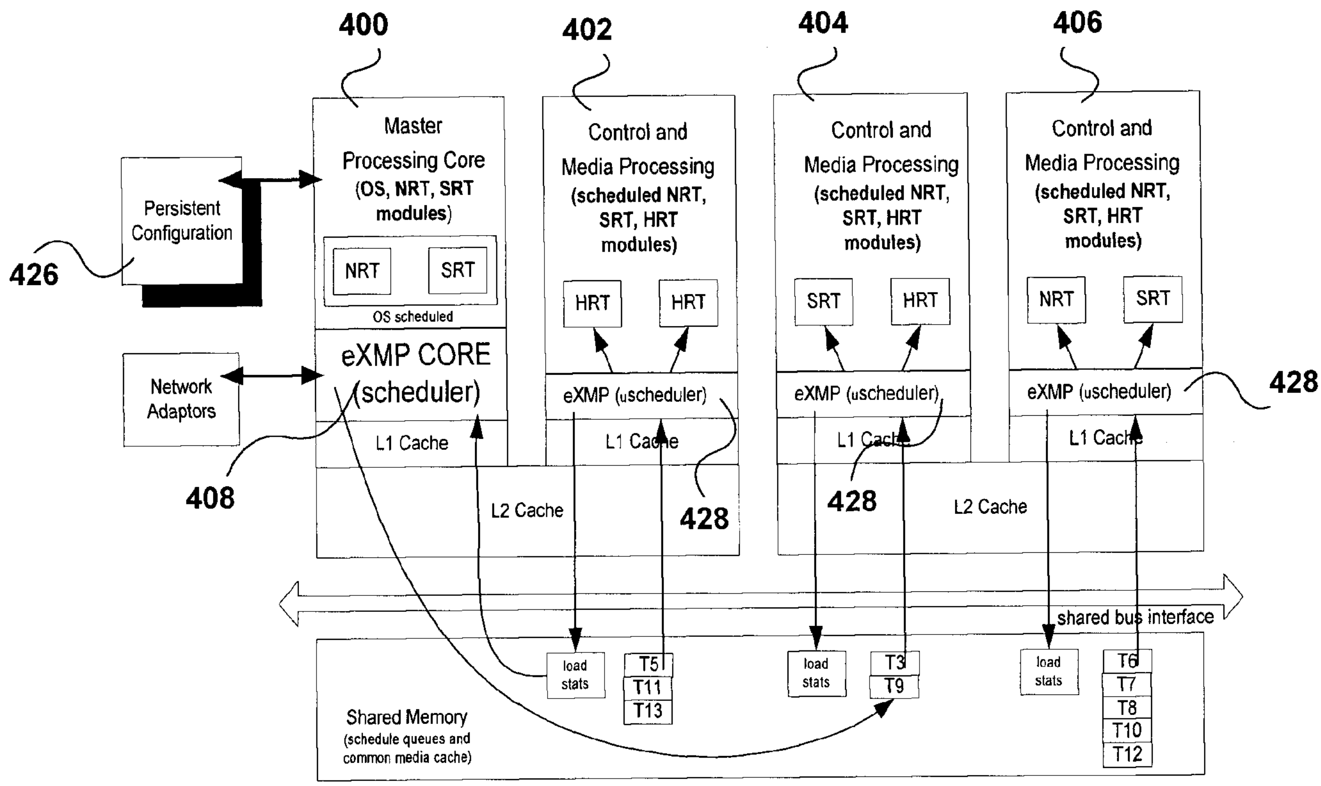

400
402
404
406
Master Processing Core
(OS, NRT, SRT modules)
NRT
SRT
OS scheduled
Control and Media Processing
(scheduled NRT, SRT, HRT modules)
HRT
HRT
SRT
HRT
NRT
SRT
Persistent Configuration
426
Network Adaptors
eXMP CORE
(scheduler)
eXMP (uscheduler)
428
L1 Cache
L2 Cache
408
shared bus interface
load stats
T5
T11
T13
T3
T9
T6
T7
T8
T10
T12
Shared Memory
(schedule queues and common media cache)

Fig. 4

INTERNATIONAL SEARCH REPORT	International application No. PCT/CA2007/000494

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***H04L 29/02*** (2006.01) , ***G06F 15/80*** (2006.01)
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)
IPC: H04L, US Classification: 700/2

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

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
Delphion: media server, real-time, server, classification, category, parallel, parallel processing, processing core, plural processors, class, category, module, sort, segment, divide, pieces

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,748,468 Notenboom et al., 05 May 1998 (05-05-1998), abstract, column 1 line 10 to column 2 line 67, column 7 lines 29-54, column 8 lines 45-60.	1-5, 6, 8, 11-15, 16, 18
A	US 5,576,945 McCline et al., 19 November 1996 (19-11-1996), whole document.	1-20
A	US 5,799,149 Brenner et al., 25 August 1998 (25-08-1998), whole document	1-20

[ ] Further documents are listed in the continuation of Box C. [X] 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
26 July 2007 (26-07-2007)	27 July 2007 (27-07-2007)

Name and mailing address of the ISA/CA	Authorized officer
Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476	Stuart Ginn 819- 934-5147

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2007/000494

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US5748468	05-05-1998	NONE	
US5576945	19-11-1996	CA2211112 A1	01-08-1996
		JP11504142T T	06-04-1999
		WO9623253 A1	01-08-1996
US5799149	25-08-1998	NONE	