

Search query (2): [SP]: asynchronous task multiplexing and chaining the described technology is directed towards sharing asynchronous (async) tasks between task chains, including in a way that prevents cancellation of lower-level chain entity from cancelling a shared async task. a shared async task is wrapped in multiplexer code that maintains lower-level entity identities as a set of listeners of the shared async task, and when a listener cancels, only removes that listener from the set of listeners so that the shared async task does not cancel as long as one listener remains in the set. also described is optimization to share an async task, and wrapping tasks in cancel-checking code that prevents the task from running its work if the task is intended to be cancelled but is queued to run before the cancel request is queued to run.

Results: 10

1) Family number: 61843274 (US2016070595A)

© PatBase

Title: ASYNCHRONOUS TASK MULTIPLEXING AND CHAINING

Abstract: The described technology is directed towards sharing asynchronous (async) tasks between task chains, including in a way that prevents cancellation of lower-level chain entity from cancelling a shared async task. A shared async task is wrapped in multiplexer code that maintains lower-level entity identities as a set of listeners of the shared async task, and when a listener cancels, only removes that listener from the set of listeners so that the shared async task does not cancel as long as one listener remains in the set. Also described is optimization to share an async task, and wrapping tasks in cancel-checking code that prevents the task from running its work if the task is intended to be cancelled but is queued to run before the cancel request is queued to run.

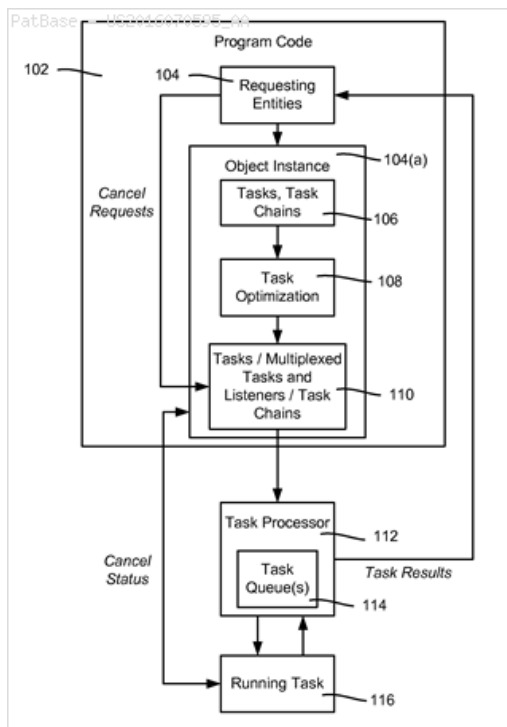
Classifications:

International (IPC 8-9): G06F9/46 G06F9/48

G06F9/54 (Advanced/Invention)

CPC: G06F9/4881 G06F9/542 G05B2219/25367 G06F9/54

US: 718/106



Family:

Publication number	Publication date	Application number	Application date
CA2960181 AA	20170303	CA20152960181	20150903
CO20170003242 A2	20170620	CO20170003242	20170403
EP3189435 A1	20170712	EP20150771777	20150903
MX2017002873 A1	20170530	MX20170002873	20150903
SG11201701779W A1	20170427	SG20171101779	20150903
US2016070595 AA	20160310	US20150843870	20150902
US2017109201 AA	20170420	US20160395481	20161230
US9658885 BB	20170523	US20150843870	20150902
WO16036983 A1	20160310	WO2015US48403	20150903

Priority:

US20140046116P 20140904 US20150843870 20150902 WO2015US48403 20150903
US20160395481 20161230

Probable Assignee: HOME BOX OFFICE INC

Assignee(s): (std): HOME BOX OFFICE INC

Assignee(s): PARKER J JORDAN C ; FINNIGAN PATRICK ; FURTWANGLER BRANDON C ; FURTWANGLER NATHAN J E ; FURTWANGLER TYLER R

Inventor(s): (std): BRANDON C FURTWANGLER ; FINNIGAN PATRICK ; FURTWANGLER BRANDON C ; FURTWANGLER NATHAN J E ; FURTWANGLER TYLER R ; J JORDAN C PARKER ; NATHAN J E FURTWANGLER ; PARKER J JORDAN C ; PATRIK FINNIGAN ; TYLER R FURTWANGLER

Inventor(s): PATRICK FINNIGAN

Agent(s): SMART AND BIGGAR; CAROLINA MERCEDES DAZA MONTALVO; GRUENECKER PATENT U RECHTSANWALTE PARTG MBB; AMIN TUROCY AND WATSON LLP; MICHALIK ALBERT S ET AL

Designated states: AE AG AL AM AO AT AU AZ BA BB BE BF BG BH BJ BN BR BW BY BZ CA CF CG CH CI CL CM CN CO CR CU CY CZ DE DK DM DO DZ EC EE EG ES FI FR GA GB GD GE GH GM GN GQ GR GT GW HN HR HU ID IE IL IN IR IS IT JP KE KG KM KN KP KR KZ LA LC LI LK LR LS LT LU LV LY MA MC MD ME MG MK ML MN MR MT MW MX MY MZ NA NE NG NI NL NO NZ OM PA PE PG PH PL PT QA RO RS RU RW SA SC SD SE SG

2) Family number: 12974385 (US6490611B)

© PatBase

Title: User level scheduling of inter-communicating real-time tasks

Abstract: In a multi-tasking system a writer task generates real-time and non real-time messages having multiple priority levels in an ascending order arranged in a plurality of priority bands. The real-time messages have predetermined timing deadlines. A plurality of queues indexing the messages by pointers, there is one queue for each priority band. A dispatcher moves pointers from a lower priority queue to a higher priority queue in response to a time-out signal dependent on the timing deadline. A data push agent transmits the messages indexed by the pointers according to the multiple priority levels.

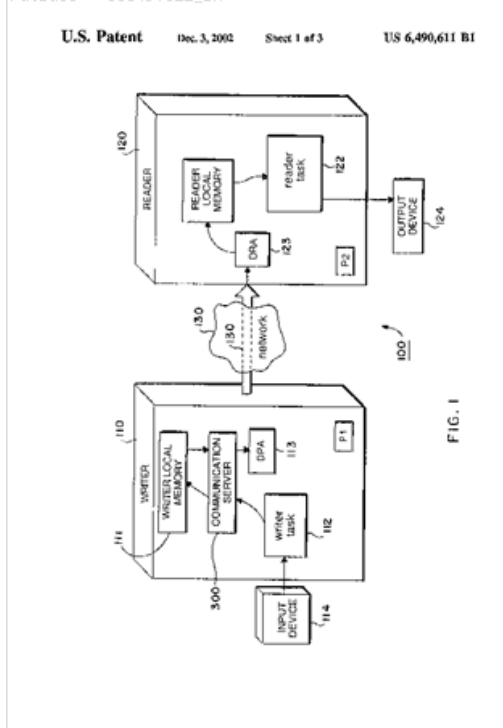
Classifications:**International (IPC 8-9):** G06F13/00 G06F9/46

G06F9/48 (Advanced/Invention);

G06F13/00 G06F9/46 (Core/Invention)

International (IPC 1-7): G06F13/00 G06F9/00 G06F9/46 G06F9/48 G06F9/54**CPC:** G06F9/4887 G06F9/544**European:** G06F9/46C4S1 G06F9/48C4S1 G06F9/54F**US:** 709/103 718/100 718/103 718/107

PatBase - US6490611_BA

**Family:**

Publication number	Publication date	Application number	Application date
EP1024429 A2	20000802	EP19990122335	19991109
EP1024429 A3	20021218	EP19990122335	19991109
JP2000284980 A2	20001013	JP19990361211	19991220
US6490611 BA	20021203	US19990239583	19990128

Priority:

US19990239583 19990128

Probable Assignee: MITSUBISHI ELECTRIC RESEARCH LABORATORIES INC

Assignee(s): (std): MITSUBISHI ELECTRIC CORP ; MITSUBISHI ELECTRIC INF TECH ; MITSUBISHI ELECTRIC INF TECHNO ; MITSUBISHI EREKUTORITSUKU INFU ; MITSUBISHI ELECTRIC RES LAB IN ; MITSUBISHI ELECTRIC RES LAB

Assignee(s): MITSUBISHI ELECTRIC RESEARCH LABORATORIES INC ; MITSUBISHI ELECTRIC INFORMATION TECHNOLOGY CENTER ; MITSUBISHI ELECTRIC INF TECHNOL CENTER AMERICA INC ; MITSUBISHI DENKI KABUSHIKI KAISHA ; MITSUBISHI DENKI KK

Inventor(s): (std): GOMEZ OSCAR J GONZALEZ ; GONZALEZ GOMEZ OSCAR J ; SHEN CHIA

Designated states: AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**3) Family number: 30259319** (US2004025163A)

© PatBase

Title: JOB MANAGEMENT IN PRESENCE OF IMPLICIT DEPENDENCY

Abstract: Methods, systems, and computer program products for job management in the presence of implicit dependency. In a situation where static dependency exists between a first computer job and a second computer job as an initial start condition for the second job, where the start of the second job depends on the completion of the first job, a scheduler determines whether the first job has created a child job, in which case the initial start condition is changed into an adjusted start condition by adding an implicit dependency, whereby the start of the second job is made further to depend at least on the completion of the child job.

Classifications:**International (IPC 8-9):** G06F9/00 G06F9/46 (Advanced/Invention);

G06F9/00 G06F9/46 (Core/Invention)

International (IPC 1-7): G06F9/00**CPC:** G06F2209/506 G06F9/5038**European:** G06F9/50A6E**US:** 718/102 718/106



US2010318995 AA	20101216	US20090481770	20090610
US8959517 BB	20150217	US20090481770	20090610

Priority:

US20090481770 20090610

Probable Assignee: MICROSOFT TECHNOLOGY LICENSING LLC

Assignee(s): (std): CALLAHAN DAVID ; GUSTAFSSON NIKLAS ; MICROSOFT CORP ; MESSMER WILLIAM R ; RINGSETH PAUL F

Assignee(s): MICROSOFT TECHNOLOGY LICENSING LLC

Inventor(s): (std): CALLAHAN DAVID ; GUSTAFSSON NIKLAS ; MESSMER WILLIAM R ; RINGSETH PAUL F

Agent(s): MICROSOFT CORP; RAGHU; KEVIN; MICKEY

5) Family number: 49222747 (US7979870B)

© PatBase

Title: METHOD AND SYSTEM FOR LOCATING OBJECTS IN A DISTRIBUTED COMPUTING ENVIRONMENT

Abstract: Method and system for locating objects in a computing grid. Requests for one or more objects are received from a client or other requesting entity. A persistent object locator is distributed across the grid and searches the grid for a reference to or handle on the requested object. The persistent object locator includes an internal object locator element (IOL) that resides locally, e.g., on the client, and an external object locator element (EOL) that resides on one or more other servers that are external relative to the client. The EOL is initiated if the IOL cannot locate the requested object locally. The EOL searches the grid for the requested object and if the object is available, delivers the reference to or handle on the requested object to the requesting entity. Objects can also be registered to the POL for future use. References to objects can also be deleted from the POL in the event of a communication failure or if a server or process terminates as expected.

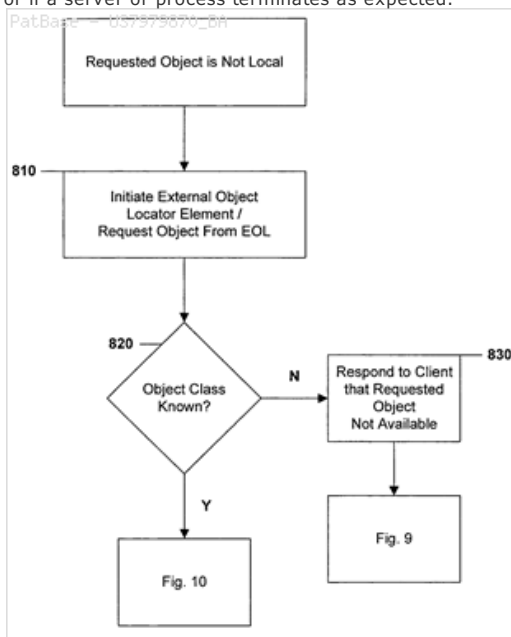
Classifications:

International (IPC 8-9): G06F13/00 G06F15/16 G06F15/173 G06F9/46 (Advanced/Invention)

CPC: G06F9/524 G06F11/3409 G06F11/3476 G06F9/5072 H04L29/08648 H04L41/0233 G06F9/4843 G06F9/485 G06F9/4881 G06F11/0793 G06Q10/10 G06Q50/22 G06F11/0709

European: G06F9/52D

US: 709/201 709/202 709/203 709/224 709/225 718/100 718/106 719/330



Family:

Publication number	Publication date	Application number	Application date
US7979870 BA	20110712	US20050297609	20051208
US8108878 BA	20120131	US20050297604	20051208
US8244854 BA	20120814	US20050297268	20051208
US8250131 BA	20120821	US20050297720	20051208
US8806490 BA	20140812	US20050297562	20051208

Priority:

US20040634453P 20041208	US20040634455P 20041208	US20040634496P 20041208
US20040634481P 20041208	US20040634628P 20041208	US20040634687P 20041208
US20040634717P 20041208	US20040634718P 20041208	US20040634719P 20041208
US20040634720P 20041208	US20040634694P 20041208	US20050297604 20051208
US20050297268 20051208	US20050297562 20051208	US20050297609 20051208
US20050297720 20051208		

Probable Assignee: CADENCE DESIGN SYSTEMS INC

Assignee(s): (std): CADENCE DESIGN SYSTEMS INC ; PULSIPHER DARREN W

Inventor(s): (std): PULSIPHER DARREN W

Agent(s): VISTA IP LAW GROUP LLP

6) Family number: 49411413 (US2011202924A)

© PatBase

Title: ASYNCHRONOUS TASK EXECUTION

Abstract: Techniques for asynchronous task execution are described. In an implementation, tasks may be initiated and executed asynchronously, thereby allowing a plurality of calls to be made in parallel. Each task may be associated with a respective timeout that triggers an end to execution of the task. If a timeout for a low priority task expires without completing both the low priority task and a relatively higher priority task, then the low priority task may use the relatively higher priority task to extend execution time of the low priority task in order to allow additional time to perform the low

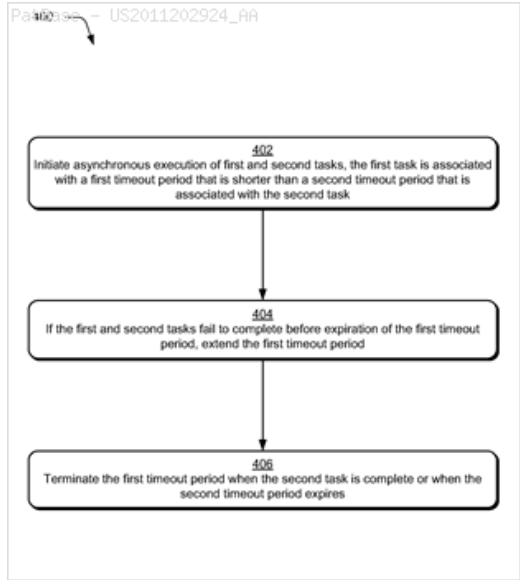
priority task.

Classifications:

International (IPC 8-9): G06F9/46 (Advanced/Invention)

CPC: G06F9/485 G06F9/4887

US: 718/102 718/103 718/103P 718/105 718/107



Family:

Publication number	Publication date	Application number	Application date
US2011202924 AA	20110818	US20100706931	20100217
US8713571 BB	20140429	US20100706931	20100217

Priority:

US20100706931 20100217

Probable Assignee: MICROSOFT TECHNOLOGY LICENSING LLC

Assignee(s): (std): BANGUERO EDGAR A ; CREVIER DANIEL W ; HOLTON ERIC J ; MICROSOFT CORP ; PERRIN BRIAN M ; MIRANDA STEINER JOSE EMMANUEL ; STEINBOK JEFFREY E

Assignee(s): MICROSOFT TECHNOLOGY LICENSING LLC

Inventor(s): (std): BANGUERO EDGAR A ; CREVIER DANIEL W ; HOLTON ERIC J ; MIRANDA STEINER JOSE EMMANUEL ; PERRIN BRIAN M ; STEINBOK JEFFREY E

Agent(s): BRYAN; MICKY; ANDREW

7) Family number: 49664756 (US2011239217A)

© PatBase

Title: PERFORMING A WAIT OPERATION TO WAIT FOR ONE OR MORE TASKS TO COMPLETE

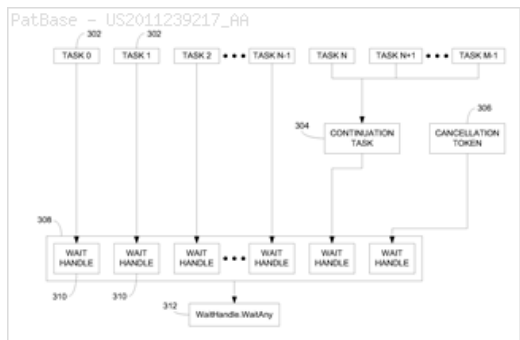
Abstract: A method of performing a wait operation includes creating a first plurality of tasks and a continuation task. The continuation task represents a second plurality of tasks. The continuation task and each of the tasks in the first plurality have an associated wait handle. The wait handles for the first plurality of tasks and the continuation task are stored in an array. A wait operation is performed on the array, thereby waiting for at least one of the tasks in the first and second pluralities to complete.

Classifications:

International (IPC 8-9): G06F9/46 G06F9/52 (Advanced/Invention)

CPC: G06F9/522

US: 1/1 718/100 718/100P



Family:

Publication number	Publication date	Application number	Application date
US2011239217 AA	20110929	US20100732552	20100326
US9229788 BB	20160105	US20100732552	20100326

Priority:

US20100732552 20100326

Probable Assignee: MICROSOFT TECHNOLOGY LICENSING LLC

Assignee(s): (std): DUFFY JOHN ; HOAG JOSEPH E ; LIDDELL MIKE ; MICROSOFT CORP ; MICROSOFT TECHNOLOGY LICENSING LLC ; SHIH DANNY S ; TOUB STEPHEN H ; YILDIZ HUSEYIN S

Inventor(s): (std): DUFFY JOHN ; HOAG JOSEPH E ; LIDDELL MIKE ; SHIH DANNY S ; TOUB STEPHEN H ; YILDIZ HUSEYIN S

Agent(s): KEVIN; MICKY; RAGHU

Title: TASK SCHEDULING BASED ON DEPENDENCIES AND RESOURCES

Abstract: An example system identifies a set of tasks as being designated for execution, and the set of tasks includes a first task and a second task. The example system accesses task dependency data that corresponds to the second task and indicates that the first task is to be executed prior to the second task. The example system, based on the task dependency data, generates a task dependency model of the set of tasks. The dependency model indicates that the first task is to be executed prior to the second task. The example system schedules an execution of the first task, which is scheduled to use a particular data processing resource. The scheduling is based on the dependency model.

Classifications:

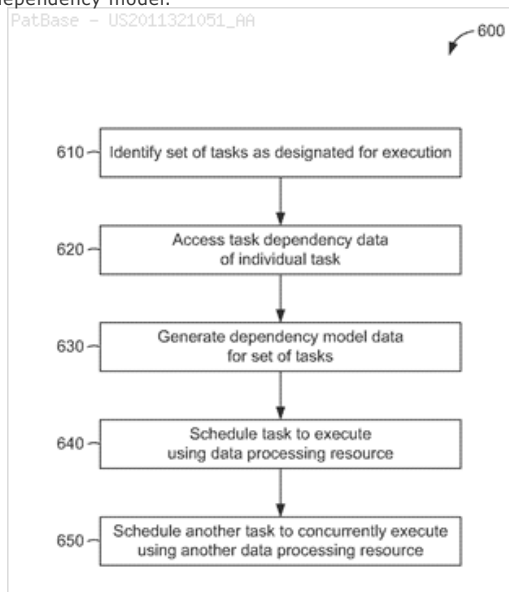
International (IPC 8-9): G06F9/46 G06F9/48 G06F9/50

H04L29/08 (Advanced/Invention)

CPC: H04L29/08963 G06F9/5038 G06F9/4881

European: G06F9/48C4S

US: 1/1 718/102 718/102P 718/103 718/106

**Family:**

Publication number	Publication date	Application number	Application date
US2011321051 AA	20111229	US20110166395	20110622
US2015026691 AA	20150122	US20140509936	20141008
US8887163 BB	20141111	US20110166395	20110622
US9223628 BB	20151229	US20140509936	20141008

Priority:

US20100358837P 20100625 US20110166395 20110622 US20140509936 20141008

Probable Assignee: EBAY INC

Assignee(s): (std): EBAY INC ; RASTOGI PERV

Inventor(s): (std): RASTOGI PERV

Agent(s): SCHWEGMAN LUNDBERG AND WOESSNER PA

Title: NUMA AWARE SYSTEM TASK MANAGEMENT

Abstract: Task management in a Non-Uniform Memory Access (NUMA) architecture having multiple processor cores is aware of the NUMA topology in task management. As a result memory access penalties are reduced. Each processor is assigned to a zone allocated to a memory controller. The zone assignment is based on a cost function. In a default mode a thread of execution attempts to perform work in a queue of the same zone as the processor to minimize memory access penalties. Additional work stealing rules may be invoked if there is no work for a thread to perform from its default zone queue.

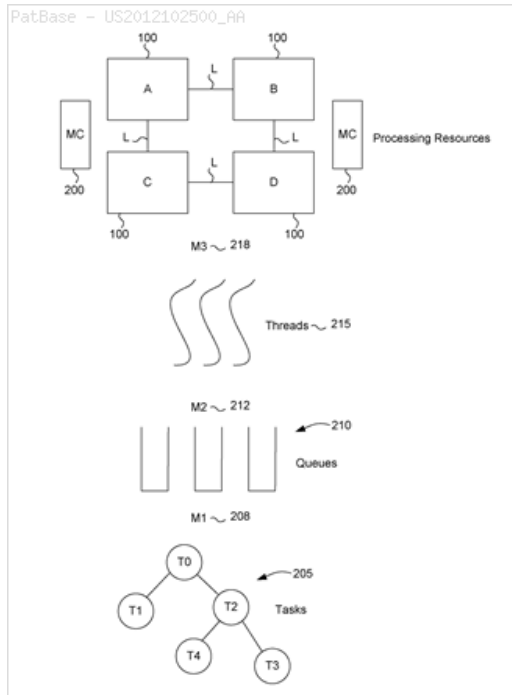
Classifications:

International (IPC 8-9): G06F13/14 G06F15/80 G06F3/00 G06F5/00

G06F9/06 G06F9/38 G06F9/46 G06F9/50 (Advanced/Invention)

CPC: G06F9/5061 G06F2209/505

US: 1/1 718/100 718/102 718/102S 718/104 718/104P 718/105 718/105P



Family:

Publication number	Publication date	Application number	Application date
KR20120042619 A	20120503	KR20110066116	20110704
KR20120042654 A	20120503	KR20110103240	20111010
US2012102500 AA	20120426	US20110077612	20110331
US2012102501 AA	20120426	US20110077567	20110331
US8640140 BB	20140128	US20110077567	20110331
US9152468 BB	20151006	US20110077612	20110331

Priority:

US20100406537P 20101025 US20110077567 20110331 US20110077612 20110331

Probable Assignee: SAMSUNG ELECTRONICS CO LTD

Assignee(s): (std): SAMSUNG ELECTRONICS CO LTD ; TIAN CHEN ; WADDINGTON DANIEL

Assignee(s): SAMSUNG ELECTRONICS CO

Inventor(s): (std): CHEN TIAN ; DANIEL WADDINGTON ; TIAN CHEN ; WADDINGTON DANIEL

Agent(s): MICHAEL; SHERMAN AND ZARRABIAN LLP; KENNETH L; HEMAVATHY; SHERMAN IP LLP

10) Family number: 51053823 (US2012110581A)

© PatBase

Title: TASK CANCELLATION GRACE PERIODS

Abstract: A command to perform a task can be received and the task can be started. A command to cancel the task can also be received. The task can be provided with a warning signal and a predetermined grace period of time before cancelling the task, which can allow the task to prepare for cancellation, such as by shutting down cleanly. If the task has not shut down within the grace period, then the task can be cancelled after the grace period expires.

Classifications:

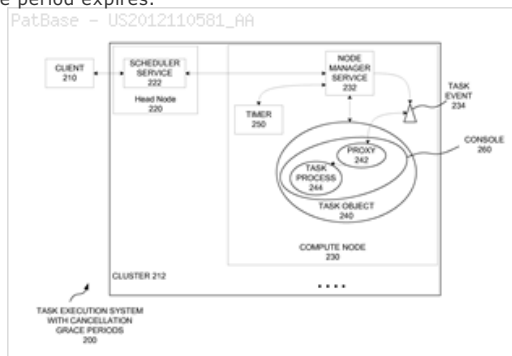
International (IPC 8-9): G06F15/16 G06F9/44

G06F9/46 (Advanced/Invention)

CPC: G06F9/4418 G06F9/485

European: G06F9/48C 4P

US: 718/100 718/100P



Family:

Publication number	Publication date	Application number	Application date
CN102467373 A	20120523	CN201010536241	20101028
US2012110581 AA	20120503	US20110101156	20110505

Priority:

CN201010536241 20101028

Probable Assignee: MICROSOFT TECHNOLOGY LICENSING LLC

Assignee(s): (std): CHAKRAVORTY SAYANTAN ; MICROSOFT CORP ; SU JUN ; WATSON COLIN

Assignee(s): MICROSOFT TECHNOLOGY LICENSING LLC

Inventor(s): (std): CHAKRAVARTY S ; CHAKRAVORTY SAYANTAN ; SU JUN ; WATSON C ; WATSON COLIN
Inventor(s): C WATSON ; CWATSON ; S CHAKRAVARTY ; SCHAKRAVARTY