Resource-type weighting is used in evaluating the use-rights associated with hardware resources.
PARTITION P2 WORKLOAD WL PROCESSOR H1 PROCESSOR H2 PROCESSOR H3 PROCESSOR H4

MANAGEMENT STATION MW1
GLOBAL MANAGER GM1
POLICIES MP1
WEIGHTING TABLE WT1
TYPE-A = 2.0 RTU
TYPE B = 1.0 RTU
RIGHTS TRACKER 23
6.0 RTU

TRANSFER USE RIGHTS BETWEEN UNEQUAL RESOURCES M24
ACTIVATE DIFFERENT QUANTITY OF 2ND RESOURCES M25

FIG. 2
RESOURCE-TYPE WEIGHTING OF USE RIGHTS

[0001] This application claims priority from copending U.S. patent application Ser. No. 11/742,656, filed 2007 May 1, of which this application is a divisional.

BACKGROUND OF THE INVENTION

[0002] Purchasers of computer systems who buy only enough hardware to meet current average demand often face costly interruptions when upgrading hardware to meet increased demands. In a “limited-right-to-use” purchase model, a purchaser purchases a computer system with reserve computing power, but pays only for a subset of the installed hardware and software (e.g., an operating system) until the rest is required, e.g., by future increases in demand. Refinements of this business model 1) allow a user to make temporary use of the reserve capacity to handle spikes in demand, 2) allow use rights to migrate among resources to facilitate a reallocation of resources to workloads. IntravVEN and interserver migration of use rights are addressed in copending patent application Ser. Nos. 11/590,334 and 11/590,584.

[0003] Herein, related art is described to facilitate understanding of the invention. Related art labeled “prior art” is admitted prior art; related art not labeled “prior art” is not admitted prior art.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The figures depict implementations/embodiments of the invention and not the invention itself.

[0005] FIG. 1 is a combination schematic diagram and flow chart in accordance with embodiments of the invention.

[0006] FIG. 2 is a combination schematic diagram and flow chart in accordance with other embodiments of the invention.

DETAILED DESCRIPTION

[0007] A server system API comprises host resources HR1, and partitions P11 and P12 for running user workloads W11 and W12, as shown in FIG. 1. While server system API is a single housing with two or more partitions, the invention applies to systems with partitions distributed among separate standalone servers. While the illustrated partitions have particular hardware resources, the invention applies quite generally to diverse sets of hardware resources.

[0008] Partition P11 includes processors PA1-PA4, memory modules MB1-MB4, and communications devices (e.g., network interface cards or host-bus adapters for storage-arrays networks) CC1 and CC2. At the time represented in FIG. 1, processors PA1 and PA2, memory modules MB1 and MB2, and communications device C21 are active (A), while the remaining hardware resources are not active (N). Partition P21 includes processors PD1-PD4, memory modules ME1-ME4, and communications devices CF1 and CF2. Processors PD1-PD3, memory modules ME1-ME3, and communications device C21 are active.

[0009] Host resources 11 run a number of platform management programs including a request handler 13, a resource controller 15, a use rights calculator 17, and a license tracking and accounting program (license tracker) 18. License tracker 18 stores data regarding existing license rights. Included with request handler 13 is a table 19 of weighting coefficients for different types of hardware resources, for example, type A processors (e.g., processors PA1-PA4) have a 1.0 weighting coefficient, while type D processors (e.g., processors PD1-PD4) have a weighting coefficient of 2.0. This means, for example, that the license rights consumed when a D-type processor is activated are double those consumed when an A-type processor is activated. In addition, license tracker 18 monitors hardware usage and charge usage according to licensing terms (which can include prepaid temporary license rights and pay-as-you-go terms).

[0010] Request handler 13 handles manual and automated requests for changing resource levels on a permanent or temporary basis. For example, a user can request that an additional processor be permanently activated for partition P11. Request handler 13 determines whether or not a processor is available for partition P11 and determines whether the necessary use rights were prepaid or, if not, whether they could be charged or purchased. Request handler 13 can communicate with a vendor system to purchase additional use rights as specified by a configuration file for request handler 13.

[0011] Request handler 13 calls use rights calculator 17 to determine whether available rights are sufficient for meeting a request to activate resources. Use rights calculator 17 uses the weighting coefficients table to make this determination. The calculator can determine from license tracker 18 the total amount of rights available and the total amount currently used and then determine whether the difference is sufficient to allow the resource request to be met. If the rights are not sufficient, the request is so notified. If the rights are sufficient, request handler 13 commands resource controller 15 to activate resources as required. The change in activation is recognized by license tracker 18, which handles any necessary accounting changes.

[0012] A method ME1 executed by server system API is flowcharted in FIG. 1. Method segment M11 involves a resource allocation action such as an implementation of a change in resource allocation or a request for the same. Method segment M12 involves quantitatively evaluating license status as a weighted sum of actual or planned active resources. If method segment M11 involved implementation of a change in resource activations, then method segment M12 can involve detecting the new configuration and reflecting it in accounting at method segment M13. For example, prepaid rights can be debited, or a charge account can be charged. If method segment M11 involves only an unimplemented request, method segment M12 provides that the request can be met at method segment M14.

[0013] In the latter case, if the determination is that the request cannot be met, the requestor is notified of the insufficiency at method segment M15. Such a notice can allow the requestor to revoke the request or to take action (e.g., purchase additional use rights) to remedy the insufficiency. If the determination is that the request can be met, the new allocation can be implemented at method segment M16. Method ME1 can be iterated whenever resource requirements change.

[0014] Resource activations can be permanent or temporary. Temporary resource activations can be terminated at a predetermined time or under predetermined conditions, e.g., as specified in the request, or can be terminated by a separate request. A request to inactivate a temporarily activated resource can be implemented immediately at method segment M11, with the appropriate impact on accounting being recorded at method segment M13.

[0015] There are several useful scenarios for the invention. A user with prepaid rights can use them flexibly to purchase rights to different classes of devices (e.g., processors versus...
storage versus communications) and of different types (e.g., high performance versus low performance) within a class. The ability to pool rights across devices of the same class but different performance is particularly applicable for systems having components of different generations of technology.

The invention also has applicability to scenarios involving workload management, as it allows resources of one class or type to be exchanged for resources of another class or type. For example, processing power can be migrated from one workload to another even though the workloads run on respective stand-alone systems of different generations. In this case, a global workload manager can take the weighting coefficients into account when assigning resources to workloads.

While migration of use rights among different resources is straightforward, migration of use rights between resources with disparate capabilities can disrupt an intended correspondence between pricing and utility. The present invention applies use-rights weightings to hardware resources to provide for flexible migration of use rights while maintaining a relatively constant relationship between utility and pricing.

In the example of FIG. 2, a server system AP2 includes servers S1 and S2 and a management workstation MS1. Server S1 includes system-level hardware on which workload management software WM1 is run, partitions (e.g., partition P1, on which workloads (e.g., workload WL1) are run, and reserved hardware R1 which is inactive due to use rights limits. In an alternative embodiment, workload management programs are run within the same partitions and using the same resources as the workloads being managed. At time T1, processors G1 and G2 are assigned to partition P1, while processors G3 and G4 are inactive and assigned to reserve R1. Similarly, server S2 includes system-level hardware on which workload management software WM2 is run, partitions for running workloads (e.g., workload WL2 on partition P2), and reserved hardware R2. At time T1, partition P2 includes processors H1 and H2, while processors H3 and H4 are in reserve R2.

While two standalone systems are shown in FIG. 1, the invention provides for any number of standalone systems. For example, the invention applies to a multi-partition monolithic system having two or more hard or virtual partitions or virtual machines. In addition, greater numbers of standalone systems of varying capacities can be used.

Management workstation MW1 runs software, stored on computer-readable storage media 21, for management system AP2 as a whole. In particular, global workload manager GM1 cooperates with server-specific workload managers WM1 and WM2 to track utilization, performance, and load, and reallocate hardware resources to workloads as a function of those parameters according to management policies MP1.

Global workload manager GM1 implements a reallocation plan through local workload managers WM1 and WM2, which are software agents. Workload managers contribute to reallocation planning by gathering and communicating utilization (e.g., percent of available processing power), performance (e.g., time to handle a request), and load data (e.g., quantity of requests received per second) to global workload manager GM1. In addition, workload managers WM1 and WM2 can trigger reallocation by communicating a request for additional resources (or an offer to relinquish resources) according to local management policies.

Global workload manager GM1 implements a reallocation plan through local workload managers WM1 and WM2, which are software agents. Workload managers contribute to reallocation planning by gathering and communicating utilization (e.g., percent of available processing power), performance (e.g., time to handle a request), and load data (e.g., quantity of requests received per second) to global workload manager GM1. In addition, workload managers WM1 and WM2 can trigger reallocation by communicating a request for additional resources (or an offer to relinquish resources) according to local management policies.

At method segment M22, the available hardware resources are weighted as set forth in weighting table WT1. Processes H1-H4 are rated at 1.0 use rights each, while processors G1-G4 are rated at 2.0 use rights each. In this case, processors G1-G4 are more advanced and more highly rated than processors H1-H4. Thus, one G-type processor can be exchanged for two H-type processors without affecting the total use rights available. For example, to shift processing power from partition P1 to partition P2, processor G2 can be activated and processors H3 and H4 can be added to partition P2. Once the resource equivalencies are established, resource allocation to workloads can be determined at method segment M23. The criteria are determined by the management policies. Once an allocation is determined it can be implemented at method segments M23-M25, resulting in the configuration shown for time T3 in FIG. 1.

Method segments M23-M25 can be ordered to avoid exceeding license rights at any instant. Where this is not a concern, method segments can be performed in order. At method segment MS3, hardware resources that are planned for deactivation or deactivated so that use rights associated therewith are available for transfer.

Depending on the selection of weightings, it can be difficult to perform an “even-up” migration, e.g., one in which the amount of use rights released exactly matches the newly consumed use rights. For example, if resource J has a 1.0 weighting and resource K has a 1.1 weighting, then releasing 11 instances of resource permits 10 instances of resource K to be added with no “left-over” use rights. However, smaller migrations of resource J to resource K will leave some remainder. For example, releasing five instances of resource J will permit four instances of resource K to be added, with 0.6 use rights units left unutilized.

Depending on the business model, the unused rights might lead to a cost reduction, or might not. If there is no cost reduction, management policies may bias against allocations that use paid-for use rights inefficiently. On the other hand, the presence of unused use rights may make some reallocation more attractive than they otherwise might be. For example, if in the example with type J and type K use rights, if there were 0.1 unused use rights, then a release of only one type J resource would permit a full type K resource to be added.

The resources can be processors, storage (including memory and disk-based storage), and communications devices (including I/O devices and network interfaces). For
each of the classes of resources, there can be subclasses, e.g., floating-point versus general-purpose processors, solid-state storage versus disk storage, and computer network bandwidth versus storage-array network bandwidth. Units can be in terms of clock rates, numbers of cores, storage capacity, and transfer rates.

[0028] The invention provides for different partitioning technologies including hard partitions, virtual partitions, and virtual machines. Some technologies allow finer allocations of hardware resources, e.g., fractional processors can be assigned to a workload. This capability can also be used to minimize unused use rights.

[0029] Herein, “software agents” are computer programs, and a computer “workload” is a program or set of programs. Herein, a “computer program” or more simply a “program” is an ordered set of instructions tangibly embodied in a computer-readable storage media and interpretable and executable by a central processing unit. Herein, “program” does not encompass purely abstract ideas, natural phenomena, or laws of nature. A “program set” is a set of one or more programs. All programs described herein effect changes in state in computer-readable memory.

[0030] “Use rights”, as the phrase is used herein, refers to a user’s rights to use specific hardware and software resources. “Limited use rights” and “limited rights to use” and “right-to-use limitations” all refer to situations in which there is hardware and/or software installed within a computer system for which a user does not have use rights, but for which use rights may generally be obtained, either permanently or temporarily at some cost or under some conditions.

[0031] “Computer-readable media” refers to media that can be accessed by a computer and includes volatile and nonvolatile, removable and non-removable media. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. “Computer storage media” includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data.

[0032] “Computer storage media” encompasses, but is not limited to, random access memory (RAM), read-only memory (ROM), Electrically-Erasable Programmable Read-Only Memory (EEEPROM), flash memory or other memory technology, compact disc read-only memory (CDROM), digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer.

[0033] “Communication media” encompasses computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency (RF), infrared, and other wireless media. Combinations of any of the above should also be included within the scope of “computer-readable media”.

[0034] In system API, hardware resource reallocations are determined in accordance with management policies MP1, which are user configurable. Management policies MP1 determine what factors are to be considered in allocating hardware resources and what ranges of allocations are possible. For example, management policies can require a minimum hardware configuration for a partition even under idle conditions, a maximum hardware configuration under full utilization, and various intermediate configurations. Management policies MP1 can require that the current use rights limitations be met by any reallocation. Alternatively, management policies can specify conditions under which use rights can be automatically increased (e.g., by an automated licensing arrangement) or decreased. These and other variations upon and modifications to the illustrated embodiment are provided by the present invention, the scope of which is defined by the following claims.

What is claimed is:
1. Computer-readable media comprising:
   a weightings table for assigning use-right weightings to hardware types so that some hardware types are assigned greater weights than other hardware types.
2. Computer-readable media as recited in claim 1 further comprising means for assigning costs to temporary hardware activations as a function of said weightings.
3. Computer-readable media as recited in claim 1 further comprising a workload manager for reallocating hardware resources to workloads so that a quantity of hardware resources assigned to said workloads changes proportionally more than the quantity of use rights assigned to said workloads.
4. Computer-readable media as recited in claim 3 wherein said workload manager reallocates said hardware resources without changing the quantity of said use rights.
5. Computer-readable media as recited in claim 3 wherein said workload manager changes the quantity of use rights when reallocating said hardware resources.
6. Computer-readable media as recited in claim 3 wherein said workload manager changes the quantity of use rights when reallocating said hardware processors by an amount less than at least amount of use rights associated with a single processor.
7. Computer-readable media as recited in claim 3 wherein said workload manager manages hardware resources on separate first and second standalone computer systems.
8. Computer-readable media as recited in claim 7 wherein processors on said first computer system have a different nominal performance than the processors on said second standalone computer system.
9. A method comprising assigning use-rights weightings to hardware resources in a computing system.
10. A method as recited in claim 9 further comprising assigning a cost to a temporary activation of hardware at least in part as a function of said weightings.
11. A method as recited in claim 10 further comprising reallocating hardware resources to workloads while transferring use rights from a first quantity of resources of a first resource type to a different quantity resources of a second resource type.
12. A method as recited in claim 10 wherein said reallocating does not involve changing the amount of use rights.
13. A method as recited as recited in claim 12 wherein said reallocating involves changing the amount of use rights in accordance with management policies.
14. A method as recited in claim 13 wherein said reallo-
cating involves a change in use rights costs.

15. A method as recited in claim 13 wherein said reallo-
cating does not involve a change in use rights costs.

16. A method as recited in claim 11 wherein said hardware
resources include processors with different performance rat-
ings, said different performance ratings being associated with
different use-right weightings.

17. A method as recited in claim 11 wherein said reallo-
cating involves deactivating hardware resources in a stan-
dalone computing system and activating hardware resources
in a different standalone computing system.

18. A method as recited in claim 17 wherein the quantity of
processors deactivated is not the same as the number of pro-
cessors activated, while the quantity of use rights is
unchanged from before said reallocation.

19. An allocation method comprising:
performing at least one action of a set consisting of request-
and implementing an allocation of active computer
hardware resources of different types to server parti-
tions, said types including at least first and second
resource types;
evaluating a license status of said allocation of resources as
a weighted sum of said resources wherein weightings vary by resource type, said first resource type having a
different weighting than said second resource type;
if said performing involves requesting, implementing said allocation if and only if said evaluating indicates said
license status is sufficient to accommodate said allocation;
and
if said performing involves implementing prior to said
evaluation, charging an account according to the evalu-
ated license status.

20. A method as recited in claim 19 wherein said first type
includes processors sharing a first performance specification
and said second type includes processors sharing a second
performance specification different from said first perform-
ance specification.