View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by OpenSIUC

Southern Illinois University Carbondale
OpenSIUC

Articles

Department of Electrical and Computer Engineering

2008

Enabling Location-Based Services in Data Centers

Krishna Kant
Intel Corporation

Neha Udar
Intel Corporation

R. Viswanathan
Southern Illinois University Carbondale, viswa@engr.siu.edu

Follow this and additional works at: http://opensiuc.lib.siu.edu/ece_articles

Published in Kant, K., Udar, N., & Viswanathan, R. (2008). Enabling location-based services in data centers. IEEE Network, 20-25. doi: 10.1109/MNET.2008.4694170 ©2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

Recommended Citation

Kant, Krishna, Udar, Neha and Viswanathan, R.. "Enabling Location-Based Services in Data Centers." (Jan 2008).

This Article is brought to you for free and open access by the Department of Electrical and Computer Engineering at OpenSIUC. It has been accepted for inclusion in Articles by an authorized administrator of OpenSIUC. For more information, please contact opensiuc@lib.siu.edu.

Enabling Location-Based Services in Data Centers

Krishna Kant and Neha Udar, Intel Corp.
R. Viswanathan, Southern Illinois University

Abstract

In this article, we explore services and capabilities that can be enabled by the localization of various assets in a data center or IT environment. We also describe the underlying location estimation method and the protocol to enable localization. Finally, we present a management framework for these services and present a few case studies to assess benefits of location-based services in data centers.

Major data centers routinely have several tens of thousands of assets (servers, switches, storage bricks, etc.) that usually go into standard slots in a rack or a chassis that fits the rack. The racks are 78 in high, 23–25 in wide, and 26–30 in deep. The rows of racks are arranged in pairs so that the servers in successive odd-even-row pairs face one another. Figure 1 shows a typical row of a data center with the popular rack mount assets that come in 1U/2U/4U sizes (1U is about 1.8 in). The other increasingly common configuration involves blade servers that are inserted vertically into a chassis, and the chassis fits in the rack. A typical rack may take about six chassis, each with about 14 blade servers.

The ease with which assets can be inserted into and removed from their slots makes the assets quite mobile. There are a variety of reasons for moving assets around in a data center; these include replacement of obsolete/faulty assets, operating system (OS) and application software patching, physical space reorganization, logical group changes to handle evolving applications and services, and so on. This makes asset tracking a substantial problem in large data centers, and tracking solutions are beginning to emerge [1].

In our previous work [2, 3], we explored asset tracking by exploiting wireless universal serial bus (USB) radios embedded in servers. Wireless USB (WUSB) is an upcoming replacement for the wired USB and ultimately is expected to be ubiquitous. WUSB uses ultra-wide band (UWB) at its physical layer, which can provide much better localization than other technologies (e.g., a wireless local area network [WLAN] [4]) and much more cheaply than radio frequency identification (RFID) [1]. In [3], we show that a combination of careful power control and exploitation of the geometry can localize individual servers with high accuracy.

In this article, we exploit this localization capability of UWB to provide a variety of location-based services (LBSs) in the data centers. Unlike traditional LBSs, our focus here is not on arming humans with useful information, but on enabling middleware to do a better job of resource management. As a simple example, each rack in a data center has a certain capacity for power circuits that cannot be exceeded. Therefore, a knowledge of the rack membership of servers can enable abiding by this restriction. However, in a large data center, we require more than just locations — we require an efficient mechanism to exchange location and other attributes (e.g., server load) so that it is possible to make good provisioning/migration decisions. This is where LBS services come in. We still envision the middleware to make the final selection of servers based on the appropriate policies; the function of LBS is merely to identify a "good" set of assets.

The rest of the article is organized as follows. The next section describes asset localization technologies and discusses the WUSB-based approach briefly. We then discuss how LBS fits in the management framework for the servers. Within this section we illustrate how LBS can be exploited for power and thermal balance among servers. The final section concludes the discussion.

Localization in Data Centers

In this section we discuss localization technologies, WUSB localization protocol, and implementation issues.

Localization Technologies

In principle, the most straightforward way to track assets is to make the asset enclosures (chassis, racks, etc.) intelligent so that they can detect and identify the asset being inserted or removed from a slot. Unfortunately, most racks do not have this intelligence (chassis often do). Even so, the enclosures themselves still must be localized. Hence, we must look for other (perhaps wireless) solutions to the problem. Furthermore, a change to existing infrastructure or significant external infrastructure for asset management is expensive and may itself require management. Therefore, low cost and low impact solutions are required.

RFID-based localization appears to be a natural solution for data centers, but unfortunately it requires substantial infrastructure to achieve acceptable accuracy. In particular, [1] describes such a system where each server has an RFID tag and an RFID reader per rack. The reader has a directional antenna mounted on a motorized track, and each rack has a sensor controller that is aware of its position. This is expensive to implement. The achievable accuracy of an RFID system implemented by LANDMARC is less than 2 m [4]. Thus,

■ Figure 1. *Snapshot of row of a typical data center.*

an RFID solution is neither cost effective nor can it achieve the desired localization accuracy.

Localization is a very well-studied problem in wireless networks; however, our interest is in only those technologies that are accurate enough to locate individual racks/chassis and (preferably) individual servers. Note that the localization of 1U servers requires accuracies of the order of one inch. In the following, we survey some localization technologies and address their applicability to data centers.

WLAN-based localization is extensively explored in the literature [4] and can be implemented easily in software. Unfortunately, even with specialized techniques such as a multipath decomposition method [4], the root mean square error (RSME) in the best line-of-sight (LoS) case is only 1.1 meters.

Ultrasonic or surface acoustic wave (SAW) systems perform localization based on time of flight (ToF) of sound waves. Because of the very low speed of sound, SAW systems can measure distance with an accuracy of a few centimeters. Unfortunately, SAW systems require substantial infrastructure and uninterrupted sound channels between emitter and receivers.

In [2, 3], we explored a WUSB-based localization solution that assumes that each server comes fitted with a WUSB radio (as a replacement for, or in addition to, the wired USB interface) that has requisite time of arrival (ToA)-based measurement capabilities. This can provide an effective and inexpensive localization solution.

WUSB Standardization and Platform Issues

The IEEE standards group on personal area networks (PANs) is actively working on UWB-based communications under the WiMedia alliance and 802.15.4 task group. WUSB is a middleware layer that runs atop WiMedia medium access control (MAC). 802.15.4a focuses on low data rate (LDR) applications (≤ 0.25 Mb/s), and is set to serve the specific requirements of industrial, residential, and medical applications. The design of 802.15.4a specifically addresses localization capability and is ideally suited for LBS applications. Our suboptimal choice of WUSB/WiMedia is motivated by practical considerations: as stated above, we expect WUSB to be ubiquitous; therefore, using WiMedia does not require additional expense or complexity for data center owners. Of course, everything about the proposed techniques (with the exception of timing) applies to 802.15.4a as well.

WUSB uses the MAC protocol based on the WiMedia standard. It is a domain-dependent MAC with a master-slave architecture involving a Piconet controller (PNC) and up to 255 terminals (slaves). The PNC maintains global timing using a super frame (SF) structure. The SF consists of 256 slots, and each slot has a duration of 256 microseconds. An SF consists of a beacon period, contention-access period, and contention-free period. The beacon period is used for PNC to terminal broadcasts; a contention-access period is used by the terminals to communicate with others or to ask the PNC for reserved-channel time; and a contention-free period is dedicated for individual transmissions over agreed upon time slots.

Server localization is often a crucial functionality when the server is not operational (e.g., due to replacement, repair, or bypass). Consequently, the localization driver is best implemented in the baseboard management controller (BMC) of the server rather than in the OS of the main processor. BMC is the main controller that will stay operational, as long as the server is plugged in and provides for intelligent platform management [5]. However, providing BMC control over WUSB in a post-boot environment is a challenge that is not addressed here.

Location Estimation Methods

Localization involves determining the position of an unknown node in a two- or three-dimensional space using range estimates from a few reference nodes, (i.e., nodes with known locations) to an unknown node. The range estimate can be obtained using received signal strength (RSSI), a ToA technique, an angle of arrival (AoA) technique, or a hybrid method that is a combination of any of these methods. Here, we focus on the most widely used ToA method for UWB ranging. The ToA technique determines the distance by estimating the propagation delay between the transmitter and receiver. Then, the position of an unknown node is identified using the traditional methods such as the intersection of circles or the intersection of hyperbolas using the time difference of arrival between the two ToAs [6]. However, due to errors in range measurements, a statistical estimation technique such as maximum likelihood estimation (MLE) is required. MLE estimates distributional parameters by maximizing the probability that the measurements came from the assumed distribution.

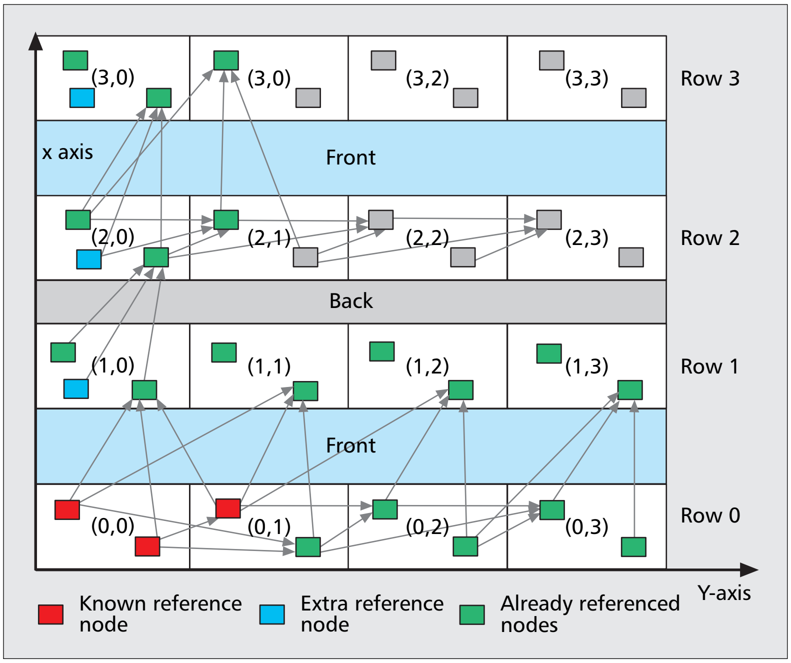
Because the server positions can take only a small number of discrete positions in a rack, the MLE problem can be transformed into a simpler maximum likelihood identification (MLI) problem [3]. MLI exploits the geometry of racks to accurately identify the position of the unknown server.

Figure 2 shows the rack configuration and an associated coordinate system (x, y, z), where x is the row offset, y is the rack offset within a row, and z is the server height in a rack. Consider rack(0,0) with N plugged-in servers. For determining the location of unknown server u, MLI uses three reference nodes, of which the first two are in rack(0,0) and the third one in rack(0,1). Each reference node i (where $i \in 1, 2, 3$) measures the distance to an unknown node u as r_{iu} using ToA. We assume that a range estimate r_{iu} is distributed as Gaussian with zero bias (that is, the expected value of the estimate equals true distance) and variance of $\sigma^2 = N_0/2$. The distance between each reference node and N-2 possible positions in the rack is known. Given the three range estimates and N-2 possible distances from each of the reference nodes, N-2 likelihood functions (LFs) are formed. Out of N-2 LFs, the minimum-valued LF identifies the position of an unknown server. In [3], it is shown that the performance of the MLI method far exceeds the performance of the traditional methods.

Localization Protocol

Asset localization in data centers involves the following two distinct phases:

- *Cold-start* phase that localizes *all* servers starting with a few reference servers with known locations



■ Figure 2. *Localization in a data center during the cold start phase.*

- *Steady-state* phase that tracks individual asset movements subsequently

The steady-state phase is relatively easy to handle and is not described here due to space constraints.

The cold-start phase starts with one of the known servers in the servers that are hard coded as PNC and all others in the listening mode. The role of the PNC is to form the Piconet with the servers from the current rack and a few servers from the adjacent and the opposite rack to enable rack-to-rack localization. One complication in cold-start localization is the avoidance of servers in racks that we are currently not interested in localizing. This, in turn, requires "macro-localization," that is, the determination of which rack the responding servers belong to, so that we can suppress the undesirable ones. This is handled by a combination of careful power control and by exploiting the geometry of the racks. Generally the localization proceeds row by row as explained below.

Row 0 Localization — We start with three known servers as shown in Fig. 2. During rack(0,0) localization, all the unknown servers in rack(0,0) and at least one server in the adjacent rack(0,1) and *two* servers in the opposite rack(1,0) are localized to enable localization in the subsequent racks as shown by red and green/black arrows in Fig. 2. (To avoid clutter, not all arrows are shown.) After the current rack localization is complete, the PNC in the current rack performs hand off to one of the localized servers (new PNC) in the rack(0,1). Thus, localization continues one rack at a time along with a few localizations in the adjacent and opposite rack until all servers in the last rack of row 0 are localized.

After the last rack localization, PNC in the last rack updates all the servers with the position of their neighbors and hands off to the selected PNC in the last but one rack in row 0. This hand off in the reverse direction continues until the rack(0,0) is reached. Now PNC in rack(0,0) is ready to hand off to the suitable known server in the rack(1,0) (odd numbered row).

Row 1 Localization — At the beginning of the row 1 localization, all the servers in row 0 are localized, and the PNC in rack(0,0) selects a known server as a new PNC in rack(1,0). In the beginning of row 1 localization, each rack in row1 has at least two known servers. But, there are no known servers in row 2. Also, given the alternating rows of front and back facing servers, communication across the back aisles is very challenging due to the heavily metallic nature of the racks as shown in Fig. 2. Therefore, only the racks located at the edge of the one row can communicate with the racks located at the edges of the next rows. During rack(1,0) localization all the servers in rack(1,0) and three servers in rack(2,0) (next even row) are localized. From rack(1,1) onwards, only the servers in the current rack are localized until the last rack in row 1 is localized. The localization in reverse direction continues as in row1 until the rack(1,0) is reached. The PNC in rack(1,0) hands off to the new PNC in rack(2,0). Location of unknown nodes in successive odd-even-row pairs continues similarly and is not discussed here.

Accuracy of Localization Protocol

The accuracy of the localization protocol depends on the variance and bias in range estimates. The variance comes from variations in channel parameters, and the bias is generally a result of partial or full occlusion of the receiver relative to the transmitter. Our previous work [2] measured variance and bias in the range estimates by direct measurements in a commercial data center. In our localization protocol, lack of line of sight and hence, substantial bias, is expected only when we hop across the back aisle. The normal technique for handling bias is simply to estimate it and remove it [7]. Thus, the assumption of no bias is still reasonable. We expect to address the question of bias estimation in future works as it requires much more intensive measurements than what we have currently.

In [3], an MLI method was proposed for localization and compared with the traditional method of hyperbolic positioning using Matlab simulation. It was shown that the performance of the MLI method far exceeds the traditional method. The probability of error in identifying a location of a node increases with the variance as expected and was found to be on the order of 10E-5 to 10E-2 for the variances between 0.15 to 0.5. It was further shown in [3] that by controlling the variance via multiple measurements, the rack-to-rack error propagation can be kept sufficiently small so that the protocol can handle large data centers.

Location-Based Services

After the servers in a data center are localized, interesting LBSs can be enabled in a data center. In this section, the need for enabling LBSs is discussed. The next subsection lists a variety of services that can exploit LBS. We then explain the management framework for enabling LBS. The last subsection illustrates the role of LBS in power and thermal balance in data centers.

The Need for LBS

Data centers show perennially low average-server utilization (in the range of 5–10 percent) but ever-increasing server count, power consumption, and associated infrastructure and management costs. The low utilization is attributable not only to unpredictable demands but, more importantly, to the need for isolation among various applications and activities. Virtualization has recently gained acceptance as a way to increase resource utilization in data centers while still maintaining a level of isolation between various applications and activities.

Aggressive virtualization leads to the notion of utility computing, whereby the entire data center can be viewed simply as a pool of resources (computes, storage, special functions, etc.) that can be allocated dynamically to applications based on the current needs.

Virtualization can be viewed as a mechanism to make the physical location of resources irrelevant because any resource can be assigned to any application in this model. Although this flexibility provides several advantages, a location-blind resource allocation can lead to anomalies, poor performance, and ultimately suboptimal resource usage. In other words, location-aware resource management can retain all the advantages of a virtualized data center while avoiding its pitfalls. We discuss these in the next few paragraphs.

The isolation requirement addressed above implies that each application executes on its own virtual cluster, defined as a set of virtual machines (or virtual nodes) connected by QoS-controlled virtual links [8]. However, the performance isolation between applications becomes increasingly difficult as more applications are mapped to common physical resources. Location awareness can be helpful in this regard. The increasing data center size and the utility-computing approach make it an increasingly attractive target of attacks by viruses, worms, focused traffic (distributed denial of service attacks), and so on. Confining a virtual cluster to a physical region offers advantages in terms of easier containment of attacks. In this context, the relevant physical region is really a network region, for example, a set of servers served by one or more switches or routers; however, the two are strongly related. For example, all blade servers in a chassis share a switch, and all chassis switches in a rack connect to the rack-level switch. Thus the location-based provisioning and migration can be beneficial from a security/isolation perspective. For essentially the same reasons, a location-aware allocation can yield better performance for latency-sensitive applications because the reduction in the number of switches on the communication paths also reduces the communication latency.

The continued increase in processing power and the reduction in physical size has increased the power densities in data centers to such an extent that both the power-in (i.e., power drawn) and power-out (i.e., power dissipated as heat) have become serious problems. For example, most racks in data centers today were designed for a maximum of 7-kWh consumption, but the actual consumption of a fully loaded rack easily can exceed 21 kWh. As a result, in older data centers, racks are often sparsely populated lest the power circuit capacity be exceeded resulting in a brownout. In addition, the power and cooling costs are becoming a substantial percentage of overall costs. Consequently, an intelligent control over both power consumption and cooling becomes essential. Power/thermal issues inherently are tied to the location of the active assets. For example, cooling can be made more effective and cheaper if the servers with high thermal dissipation are not bunched up [9].

The high velocity fans required for effective cooling in increasingly dense environments also makes noise an important issue in data centers. In addition, fans are usually the third or fourth largest consumers of power in a platform and may waste a significant fraction of that power as heat. Therefore, an intelligent control of the speed of adjacent fans not only can reduce noise but also can make cooling more effective.

Application of LBS

Because the feasible services depend on achievable localization accuracy, we introduce LBS at two levels of localization granularity:

- Coarse grain localization (CGL), defined as the ability to identify (with, say, 95 percent or better accuracy), data center racks, pods, or cubicles containing small clumps of IT equipment, storage towers, and mobile devices in the vicinity (e.g., people carrying laptops). The desired precision here is ± 0.5 meters.
- Medium grain localization (MGL), defined as the ability to identify (again, with 95 percent or better accuracy), individual plugged-in assets within a chassis (and by implication, the chassis itself) and individual mobile devices (e.g., laptops, blackberries). The desired precision here is ≈ ± 5 cm.

In the following, we list a variety of services that can exploit CGL and MGL. The list is not intended to be exhaustive but merely indicates the usefulness of LBS within a data center. Also, a real implementation of such services could include environment and usage-model-specific elements:

- Application allocation to minimize inter-process communication (IPC) or storage access delays among the virtual nodes
- Temporary inclusion of a mobile device in a logical group within its physical proximity (It is assumed that the device can communicate over a much wider physical range, so this service may use attributes beyond just the ability to communicate.)
- In an IT environment, direct a print job to the nearest working but idle printer
- Dynamic migration of virtual machines (VMs) among adjacent servers to balance per-server power-in (and especially, power-out)
- Roving query distribution to maximize power savings and balance out heat dissipation (This technique is the opposite of load balancing in that it allows idle servers to go into deep low-power states while keeping the active servers very active.)
- Logical grouping of assets based on their location to simplify inventory management, allocation, de-allocation, migration, and so on
- Trouble ticket management, that is, identification of the asset that requires replacement, fixing, software patching, and so on
- Physically segregated allocation of applications based on their trustworthiness, reliability, sensitivity, or other attributes
- Quick quarantine of all servers belonging to the same enclosure as the server that detects a denial of service (DoS) or virus attack
- Automated adjustment of air-flow direction flaps from shared fans to maximize cooling of hot spots and perhaps control fan noise (This situation is generally applicable to blade chassis that have shared fans rather than racks, which usually do not.)

A Management Framework for LBS

Currently, flexible management of virtualized data centers and creation of utility computing environments is driven by initiatives from major software vendors such Dynamic System Initiative (DSI) from Microsoft, adaptive enterprise from HP, on-demand computing from IBM, and N1 from Sun Microsystems. These initiatives are geared toward providing middleware solutions to the dynamic data center management problem, based on the information available from the OS and low-level management software running on the BMC [5]. Although the management software can implement LBS arbitrarily, based on the physical locations reported by the localization layer running in the BMC, a more structured approach is highly desirable. We envision the following three layers:

- Layer 1: application programming interfaces (APIs) to

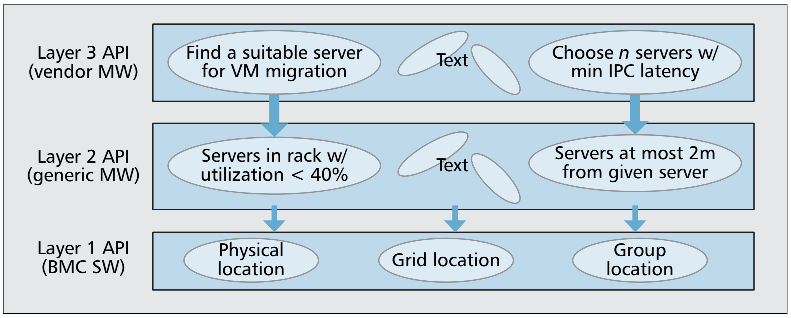


Figure 3. *Illustration of LBS application layers.*

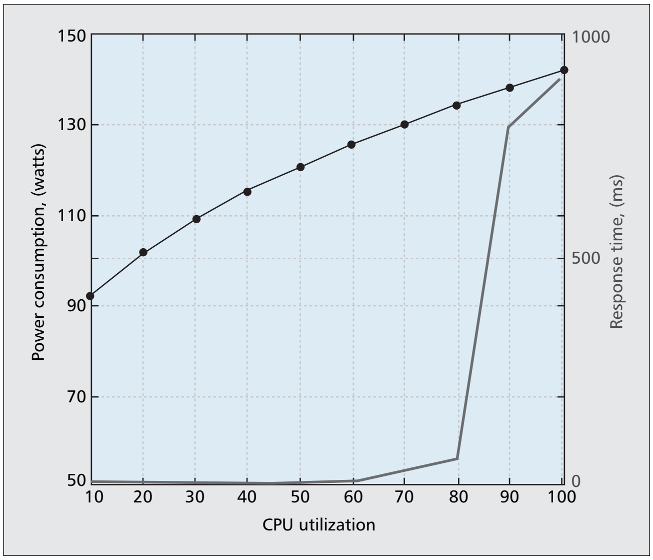


Figure 4. *CPU utilization vs power & response time.*

obtain asset location in various formats. At a minimum, three formats seem necessary:

–Physical 3-D location relative to the chosen origin
–Grid-based location (rack_row_no, rack_no, asset_no_in_rack)
–Group-level location such as location of the entire rack or chassis

- Layer 2: APIs to identify a group of assets satisfying constraints that relate to their location, static attributes (e.g., installed memory), and perhaps even the current utilization levels. For flexibility, the constraints may be expressed in a declarative fashion (see below).
- Layer 3: LBSs themselves, implemented as a part of the middleware. It is envisioned that an LBS will invoke layer 2 APIs to select promising candidates and then perform further selection based on its requirements.

Figure 3 shows an illustration of these layers and their interactions.

There is a strong trend in management software to use a standardized representation of the underlying management data and access it using Web services. In particular, the Distributed Management Task Force (DMTF) has developed a common information model (CIM) for describing computing and business entities that has been adopted widely (www.wbmsolutions.com/tutorials/CIM/cim-specification.html). For example, a CIM model of a network interface card (NIC) will have all relevant attributes of the NIC (e.g., speed, buffer size, transport segment offload [TSO], and whether it is enabled, etc.). CIM supports hierarchical models (nested classes, instances, inheritance, etc.) for describing a complex system in terms of its components. CIM models can be accessed through a Web services management (WSMAN) interface for querying and updating the attributes. The location can be part of a CIM model and can be accessed through WSMAN services.

Although CIM can adequately describe the configuration of servers and applications, a more powerful language such as the services modeling language (SML) (www.microsoft.com/business/dsi/serviceml.mspx) is required to specify service related constraints. Because it is XML-based, SML can describe schemas using XML document type definitions (DTDs). Furthermore, SML documents can refer to elements in other SML documents and thereby specify complex relationships through schematron (www.schematron.com). For example, it is possible to say something like "allocate the application to a server only if the server utilization is less than 40 percent." Thus, SML can allow for resource management based on declared constraints as opposed to those buried in the middleware code.

Exploiting LBS for Power/Thermal Balancing

In this section, we show that LBS can be used effectively to handle the issues of power and thermal balance in a data center. Consider a data center having a single row with two racks. Each rack has 12 slots and is partially filled with eight identical servers. Suppose that each rack has a maximum power draw capacity of 650 *W*. Let us consider running an application that demands 320 percent CPU utilization. In the following subsections, we analyze allocating this application in three different ways:

- Scenario 1: No localization; the server locations are unknown.
- Scenario 2: CGL: it is known that the server belongs to a particular rack, but the exact location in the rack is not known.
- Scenario 3: MGL: the exact location of the server in the rack is known.

Power-Load Balance

It is well known that the power consumption *P* relates to the CPU utilization *U* by a non-linear relationship. In [10], the authors performed detailed measurements on streaming media servers with several configurations to study the relation between CPU utilization and the power consumption, and they found that the power consumption can be expressed approximately as:

$$P = P_I + (P_F - P_I)U^{0.5} \tag{1}$$

where P_I = is the idle power, P_F is the power when CPU is fully loaded, and U is the CPU utilization.

Such a dependence is very much a function of the machine and workload characteristics, and there is no suggestion here that this is a general equation. However, it suffices to illustrate a few interesting points about power/thermal balance. We also make use of the power numbers reported in [10]: an idle power of P_I = 69 W and P_F = 145 W at full load. The authors also specify a low-power mode consumption of P_L = 35 W. This mode generally puts the CPU, memory, and disk in low-power modes.

Given the power consumption in the idle mode and the low-power mode, it is power efficient to distribute a higher load on fewer servers and force more servers in the low-power mode. The distribution of a higher load on fewer servers is limited by the response time of the server. As shown in Fig. 4, the response time takes off sharply beyond a 60-percent CPU utilization.

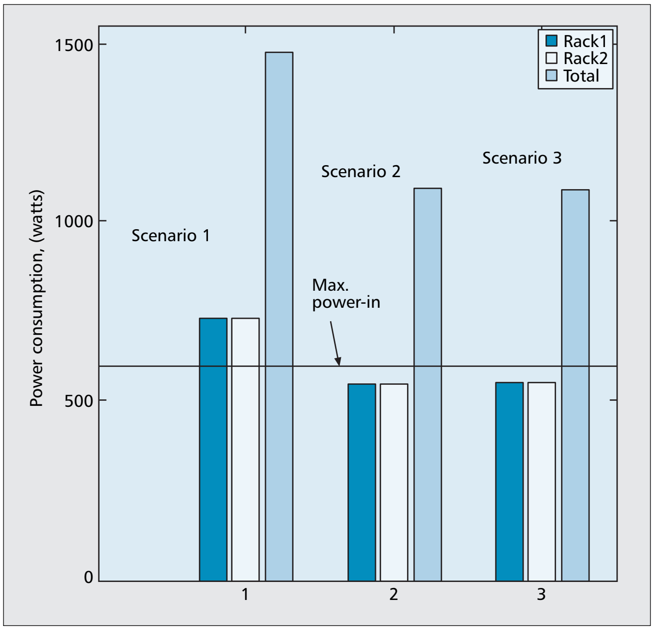


Figure 5. *Power consumption for various localization scenarios.*

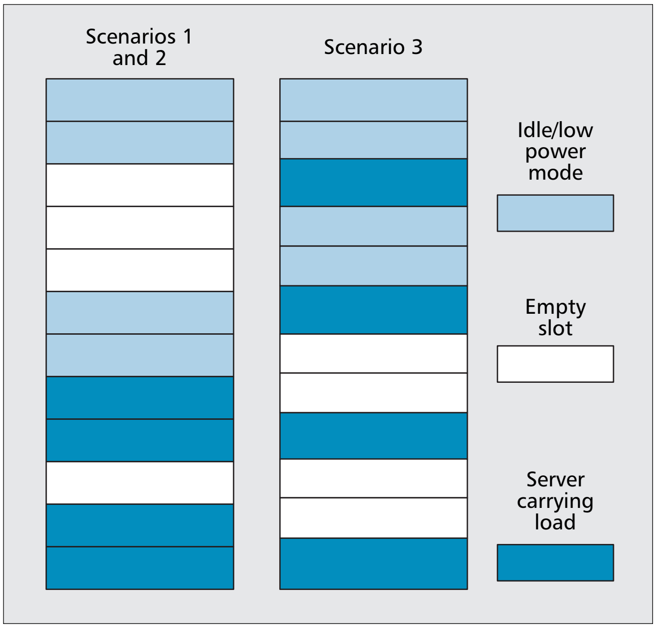


Figure 6. *Power/thermal balance scenarios.*

In Scenario 1, given that the server locations are unknown, a simple strategy is to distribute the load equally on all the available servers. In this case, each of the 16 servers carries a load of 20 percent to meet the total load demand of 320 percent. With equal-load sharing, each rack exceeds the maximum power-in for a rack as shown in Fig. 5. In Scenario 2, using CGL, it is known which servers belong to either of the two racks. Therefore, the total load is divided equally between the two racks. Further, within each rack, four out of eight servers share the 40-percent load, and the remaining servers are put in low-power mode. The non-uniform distribution of load among the available servers leads to power saving as shown in Fig. 5 and also meets the maximum power-in requirement of a rack. Scenario 3 is identical to Scenario 2 in terms of power because knowing the precise location of a server does not provide any additional advantage. Further power saving can be achieved if two servers in each rack carry a load of 60 percent, one server carries a 40-percent load, and the remaining five servers in each rack are in the low-power mode.

Thermal-Load Balance

Thermal power dissipated from the CPU is proportional to the power consumed, and a non-uniform distribution of thermal power places more demand on cooling the data center [9]. To illustrate the point, reconsider the situation of Scenarios 2 and 3 above, that is, eight servers sharing the entire load while the other eight are put in low-power mode. In Scenario 2, the lack of precise server location can result in loaded servers all being placed in physical proximities, but Scenario 3 can achieve better thermal balance by spreading out the loaded servers as shown in Fig. 6.

Conclusions

In this article we introduced the important topic of asset localization in data centers and discussed wireless, USB-based techniques for the same that do not require an external infrastructure. Further, a localization protocol for systematically localizing assets in a data center was described briefly. We also introduced the notion of location-based services and illustrated that localization can be used to obtain power/thermal balance in a data center.

References

[1] "HP Creates RFID Technology for Tracking Data Center Assets," *HP Press Release*, Palo Alto, CA, Oct. 17, 2006.

[2] N. Udar *et al.*, "Ultra Wideband Channel Characterization and Ranging in Data Centers," *ICUWB*, Sept. 2007, pp. 322–27.

[3] N. Udar, K. Kant, and R. Viswanathan, "Localization of Servers in a Data Center Using Ultra Wideband Radios," *IFIP Networking*, May 2008, pp. 756–67.

[4] H. Liu, H. Darabi, and P. Banerjee, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Trans. Syst., Man, and Cybernetics, Part C*, vol. 37, no. 6, Nov. 2007, pp. 1067–80.

[5] H. Zhuo *et al.*, "Remote Management with the Baseboard Management Controller in Eighth-Generation Dell PowerEdge Servers," in *Dell Power Solutions*, Oct. 2004. Retrieved, June 2005.

[6] Y. T. Chan, H. Y. C. Hang, and P. C. Ching, "Exact and Approximate Maximum Likelihood Localization Algorithms," *IEEE Trans. Vehic. Technol.*, vol. 55, no. 1, Jan. 2006, pp. 10–16.

[7] B. Alavi and K. Pahlavan, "Modeling of ToA-Based Distance Measurement Error Using UWB Indoor Radio Measurements," *IEEE Commun. Letters*, vol. 10, Apr. 2006, pp. 275–77.

[8] K. Kant, "Virtual Link: An Enabler of Enterprise Utility Computing," *Proc. ISPA*, Nov. 2006, pp. 104–14.

[9] R. K. Sharma et.al, "Balance of Power Dynamic Thermal Management for Internet Data Centers," *IEEE Internet Comp.*, Feb. 2005, pp. 42–49.

[10] L. Chia-Hung, B.Ying-Wen, and L. Ming-Bo, "Estimation by Software for the Power Consumption of Streaming-Media Servers," *IEEE Trans. Instr. and Meas.*, vol. 56, no. 5, Oct. 2007, pp. 1859–70.

Additional Reading

[1] WiMedia Draft MAC Standard 0.98, "Distributed Medium Access Control (MAC) for Wireless Networks," *WiMedia*, Aug. 2005.

Biographies

Krishna Kant (krishna.kant@intel.com) received his Ph.D. degree in computer science from the University of Texas at Dallas in 1981. He has been with Intel Corp. since 1997, where he has worked in a variety of research areas including traffic characterization, security/robustness in the Internet, data center networking, utility computing, and power control of computer systems. From 1991 to 1997, he was with Telcordia Technologies (formerly Bellcore) and worked on SS7 signaling and congestion control. Prior to this, he was an associate professor of computer science at Penn State University.

Neha Udar (neha.udar@intel.com) received her Ph.D. in August 2008 and is now with Intel Corp. Her dissertation deals with asset tracking in a data center using WUSB radios. Her research interests are wireless and personal communications, UWB, Zigbee networks, and WiMAX technology.

R. Viswanathan (viswa@engr.siu.edu) received his Ph.D. in electrical engineering from Southern Methodist University, Dallas, Texas in 1983. He is the interim dean and a professor in the College of Engineering, Southern Illinois University, Carbondale, Illinois. His research interests are wireless and personal communications, spread spectrum communications, detection and estimation theory, and wireless networks.