LOW-POWER COMPUTING

Safa Alsalman [a], Raihan ur Rasool [b], Rizwan Mian [c]
[a] King Faisal University, Alhsa, Saudi Arabia
[b] Victoria University, Melbourne, Australia
[c] Data++, Toronto, Canada
Corresponding email: safa.a.alsalman@outlook.com

Abstract

With the abundance of mobile electronics, demand for Low-Power Computing is pressing more than ever. Achieving a reduction in power consumption requires gigantic efforts from electrical and electronic engineers, computer scientists and software developers. During the past decade, various techniques and methodologies for designing low-power solutions have been proposed. These methods are aimed at small mobile devices as well as large datacenters. There are techniques that consider design paradigms and techniques including run-time issues. This paper summarizes the main approaches adopted by the IT community to promote Low-power computing.

Keywords: Computing, Energy-efficient, Low power, Power-efficiency.

Introduction

In the past two decades, technology has evolved rapidly affecting every aspect of our daily lives. The fact that we study, work, communicate and entertain ourselves using all different types of devices and gadgets is an evidence that technology is a revolutionary event in the human history. The technology is here to stay but with big responsibilities comes bigger challenges. As digital devices shrink in size and become more portable, power consumption and energy efficiency become a critical issue. On one end, circuits designed for portable devices must target increasing battery life. On the other end, the more complex high-end circuits available at data centers have to consider power costs, cooling requirements, and reliability issues. Low-power computing is the field dedicated to the design and manufacturing of low-power consumption circuits, programming power-aware software and applying techniques for power-efficiency [1][2]. Moore's law states that the "number of transistors that can be placed inexpensively on an integrated circuit (IC) will double approximately every two years". It sheds no light on the consequence of increased power consumption as a result. A major problem of doubling transistors is the unavoidable exponential increase in the power consumption as shown in Figure 1.

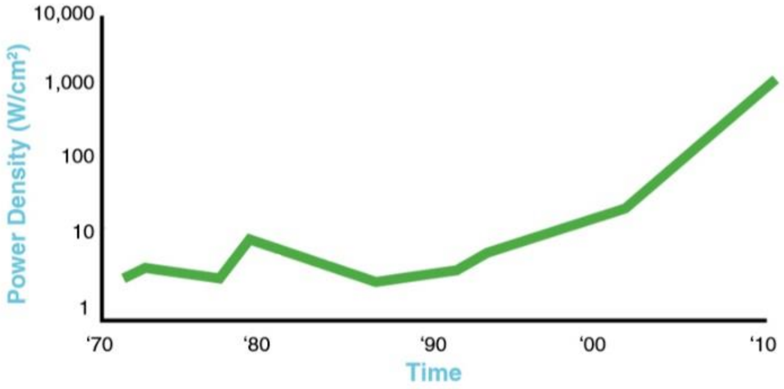


Figure 1: Power Density Exponential Growth over the Past Decades

Before 1990, power consumption was not an issue for majority of ICs. However, with the rapid growth of portable devices with high-performance processors, low-power hardware became an active topic. The first initiative to tackle the power consumption problem was the launch of the energy star program in 1992[3]. The *Energy Star* program is voluntary labelling program for promoting energy efficiency. The label is awarded to companies that have succeeded in minimizing the use of energy while maximizing efficiency. It applies to computer equipment as well as home appliances including air conditioners, fridges, television sets etc. One of the first and most remarkable results of this initiative is the sleep mode function of computer monitors that replaced the usage of screen savers [4].

Another important development in power-awareness computing was the establishment of the *Green Grid* organization in 2006. The Green Grid is a voluntary organization that develops standards for measuring datacenters' efficiency. Member companies share information about processes and technologies that can help datacenters improve performance against predefined standards. Members include Intel, IBM, Microsoft, HP, and Dell.

In 2007, the *Climate Savers Computing Initiative* (CSCI) was initiated as a non-profit group of consumers, businesses and organizations dedicated to promoting smart technologies that improve power efficiency and reduce the energy consumption of computers. In 2012, CSCI was merged with the Green Grid to realize their common goals of saving energy and reducing greenhouse gas emissions. Many companies are now considering new designs for their electronics to increase battery life and minimize their environmental impact in the light of the standards and strategies introduced by the Green Grid organization [5].

Why Low-Power Computing

Using lower-power computing practices suggests several rewards for individuals, business, and governments. These benefits include:

- **Cutting down costs:** Utility costs are a major concern in every business. Electricity prices are growing because of ever increasing power demand, and of course, natural resources are finite. While the focus on renewable energy resources could help with finite resources, adopting strategies that lower the power consumption helps in cost reduction as well [6].
- **Environmental Sustainability:** Low-power computing is a major topic in Green IT that is concerned with minimizing the environmental impact of the IT infrastructure while maximizing efficiency. As computer systems manage their power settings more carefully, the energy generated by burning fossil fuel for powering computers will be reduced resulting in a reduced CO2 footprint [7].
- **Extended Buttery Life:** In a mobile device, the battery means availability. Therefore, one of the most important attributes when purchasing a mobile device is considering its battery life. Adopting practices that lower the power consumption rate can result in a substantial increase of the battery life of a device [8]. As a result, companies are trying to increase the battery life as much as they can to get users' blessing.
- **Resource Utilization:** Lowering power consumption helps in efficient and effective resource utilization which means infrastructure such as datacenters and computers. This helps in reducing the wastage of resources by matching them with the actual demand [9].

- **Healthier life:** Electronic devices draw power as long as they are operating. Being near these devices endangers humans' lives because of heat and power dissipation. Developing low-power solutions for electronics reduces heat dissipation and therefore health hazardous possibilities.

Approaches to Low-Power Computing

The problem of reducing power consumption requires the collaboration of electronic engineers and computer scientists. Developing a low-power solution may be implemented in the early stages by designing low-power circuits and developing efficient algorithms. It can also mean implementing good programming practices to achieve lower consumption at systems' runtime such as managing the power dynamically [10]. Figure 2 is a taxonomy for the general approaches to reducing power consumption.

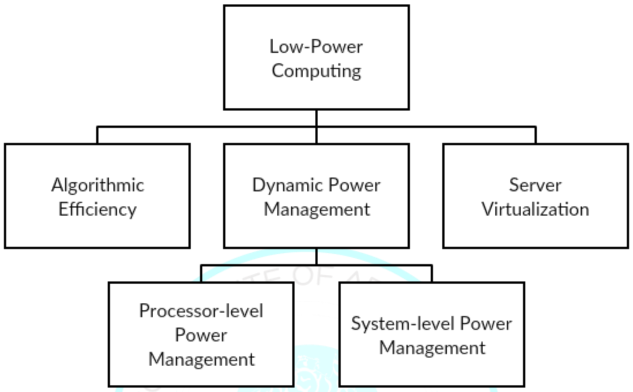


Figure 2: Taxonomy of approaches to Low-Power Computing

Algorithmic Efficiency

The property of an algorithm which relates to the amount of computational resources used by the algorithm is known as algorithm's efficiency. In the design and analysis of algorithms, time and space efficiency play a major role in defining whether the algorithm is worth implementation. However, performance is not the only parameter affected by the efficiency of algorithms. As the efficiency of algorithm increases, the number of steps involved in computation and storage requirement will reduce. Both of these will result in savings of electrical power and hence will contribute to low-power computing. For example, the use of protocols that are intelligent to make routing decisions in wireless networks [11] is an important research area in the field of Artificial Intelligence. Some strategies in this area include:

- Minimizing the energy consumed for each message.
- Minimizing the cost per packet.
- Minimizing the maximum energy drain of any node.

Dynamic Power Management

Low-Power techniques can be applied during early design stages (static) or they can be implemented during system's runtime (dynamic)[12][13]. In the IT world, electrical equipment and information systems are always designed for the peak performance consideration, although it is rarely reached. Taking a retailer website example, the maximum number of users interacting with the system is never reached unless it is the holiday season. Power consumption can be significantly reduced if the voltage and frequency of the processor are reconfigured depending on current requirements rather than the peak performance. Likewise, electrical components are usually sitting idle unless the user is actively interacting with the system.

Dynamic Power Management (DPM) is a design methodology that dynamically adapts an electronic system to deliver the demanded services by adjusting the performance according to the workload. DPM is focused around the principal of selectively minimizing system components' performance or turning them off when they are idle for a certain period of time. A control procedure called *policy* is used by power managers to control the power consumption. An example of a policy is enforcing the system to sleep after a certain period of idleness called the time-out interval. Dynamic power managers can be applied on different layers of the IT infrastructure such as devices (processors, monitors, hard disks, etc.), systems, and networks. In this paper, we discuss examples of power management techniques for processors and systems.

Processor-level Power Management

Power consumption can be reduced at this level by applying *Dynamic Voltage and Frequency Scaling* techniques to CPU voltage and clock rate [14]. In *Dynamic Voltage Scaling* (DVS), the voltage is increased (overvolting) or decreased (undervolting) based on recent circumstances. When the voltage is decreased, power consumption reduces as well. Undervolting is usually used in mobile devices where battery lifetime is vital. Moreover, Undervolting is used in large systems where power reduction affects heat dissipation and hence the requirements of cooling systems. Similarly, *Dynamic Frequency Scaling* (DFS) reduces the required power for CPU operation [15]. Since the frequency is typically a function of the voltage applied, the two techniques are usually applied over the same chip to achieve power-efficiency. Figure 3 shows the relationship between the frequency and voltage and the possible power savings when dynamic scaling is applied.

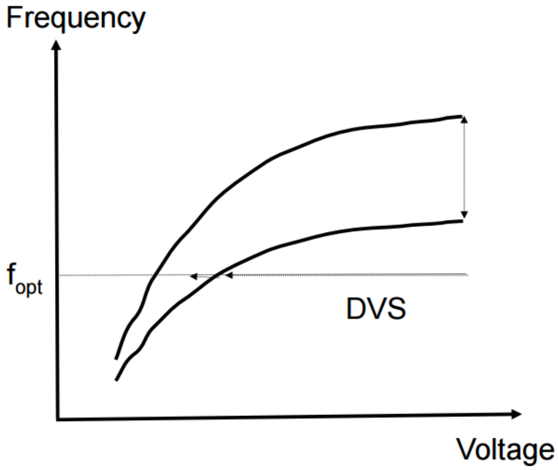


Figure 3: Voltage and Frequency relationship

The typical power consumption of a CMOS-based circuit is approximately calculated by the formula [1]:

$$P = \alpha . C . V^2 . f$$

Where α is a constant representing the CPU's switching factor, C is the effective capacitance, V is the operating voltage and f is the operating frequency. As a result of this formula, we can see that:

1. Reducing the frequency to one-half reduces the power by a factor of 2.
2. Reducing the voltage to one-half reduces the power by a factor of 4.
3. Reducing the voltage and the frequency to one-half reduces the power by a factor of 8.

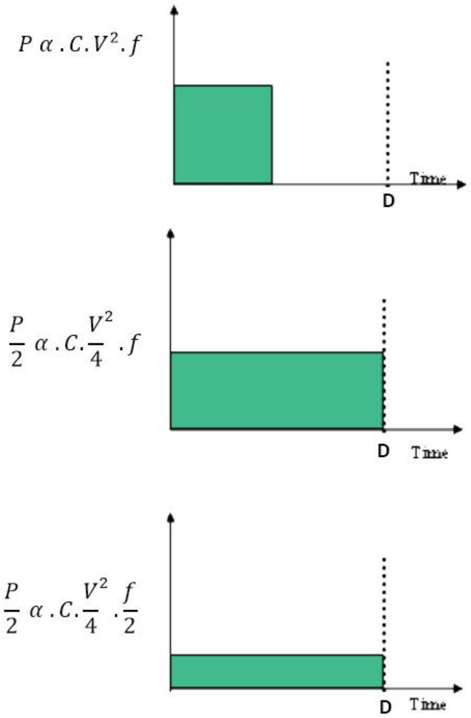


Figure 4: Effect of Frequency and Voltage Scaling on Power Reduction

However, reduction of processors' voltage results in performance degradation. Assuming that we have a task T that needs to be done before a deadline D, we can see the trade-off between the performance and power in Figure 4. Although there is a clear power saving when doing the task by decreasing the voltage or the frequency, the task will require more time to finish which may not be tolerable in some situations.

Some implementations of DFS include Intel's SpeedStep series and AMD's Cool'n'Queit which focuses on producing less heat so that the fan can operate slower and draw less power. AMD also introduced PowerNow! which is a technology used in mobile devices to improve power consumption. Other implementations include VIA Technologies' LongHaul and Transmita's LongRun.

System-level Power Management

In system-wide power management, the system or some of its components are turned off based on the actual computational workload [16]. Sometimes, however, the components are not turned-off but their performance is lowered if the workload is not at its peak point. System's workload is defined by the number of requests initiated by devices. For example, a hard disk request would be to read or write data. System components such as hard disk drives, monitors, and network equipment draw power constantly although most users frequently leave their systems idle. Hard disk drives and monitors are among the most power hungry devices in the computer systems. Shutting them down when idle results in significant power savings. Figure 5 shows an abstraction of a system-level dynamic power manager.

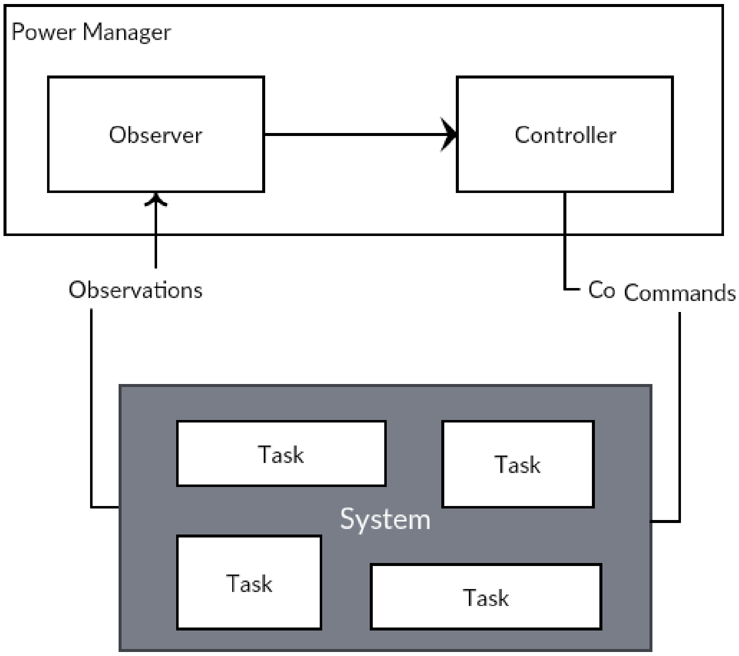


Figure 5: Abstraction Model of a System-level Dynamic Power Manager (DPM)

The system should observe the workload and decide if a certain device needs to be shut down if it has been idle for a specific period of time or it is expected to be idle for a long time. However, waking a device up after being shutdown requires some power and results in some delay. Referring to figure 6, a device needs some time t(s) to shut down and t(w) to wake up again.

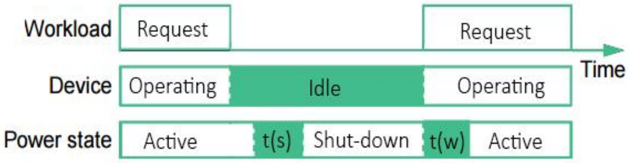


Figure 6: Dynamic Power Management Using Sleep Policy

DPM strategies for system-wide management are classified based on the predicator they use, the degree of control given to the power manager and the type of decision selected by the power manager. The *predicator* uses the pre-observed history to predict the next idle period. As for the *degree of control*, it can be reactive where the power manager cannot regulate the waking up process, or proactive in which the power manager have some control. The decision selected by the power manager can be deterministic i.e. whenever a certain situation occurs, the same decision will be taken by the power manager or the decision can be stochastic in which decisions may differ although the same situation occurred (based on the observed (current) state of the workload and of the system). Some implemented DPM policies involve [17]:

1. **Timeout-based shut down:** In this case, the system needs to observe the start of the current idle state to predict its length. The power manager can only shut the system down and has no control in the waking up process. The decision taken by the power manager is deterministic in this strategy.
2. **Pre-emptive wakeup:** The power manager observes the entire workload history in this case to predict the period of idleness. The power manager is proactive i.e. it is granted the privileges of

shutting down components as well as waking them up to avoid delays. The decision selected is also deterministic in this case.

3. **Stochastic control:** The stochastic policies assume an ideal predictor that can foresee the upcoming period of idleness. The power manager can be reactive in some situations and proactive in others. The decision is taken by the power manager based on the probability of a request reception and may vary depending on the accordingly but can be one of many possible deterministic policies such as:

- The system is never shut down resulting in no power saving but the performance does not degrade.
- The system is shut down whenever a period of idleness is detected. Consequently, the power is saved but there is some delay in performing tasks.

Server Virtualization

Datacenters use a huge number of servers to respond to users' requests. Each server is usually dedicated to some function such as file, database, email etc. As the number of servers increases to cope with data growth, power consumption increases as well [18]. However, most of the servers are underutilized most of the time. The resources provided by hardware in datacenters are much more than what a typical server needs to operate [19]. Consequently, to increase hardware utilization, it is better to move away from the physical model to a logical model [20]. To reduce energy consumption in datacenters, companies use virtualization to reduce the number of servers and the energy used to power them. In server virtualization, a single physical server is split into a number of virtual servers that may handle different functions [21], as shown in figure 7.

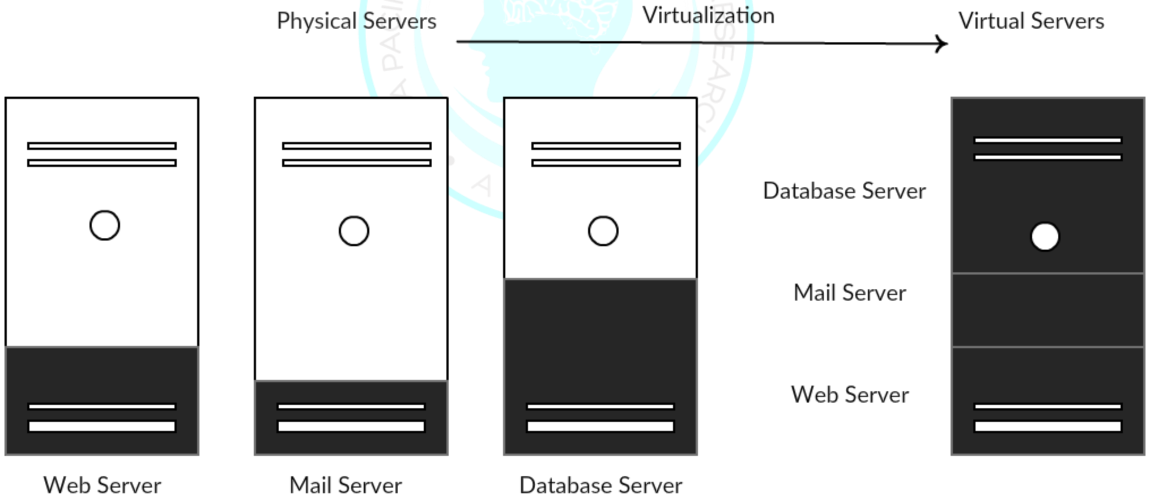


Figure 7: Server Virtualization as a technique to reduce power consumption in datacenters.

In virtualization, the OS is no longer bounded to the physical server it runs on. The virtualization layer is called hypervisor [22] and many virtual machines can be loaded over the virtualization layer [23] as illustrated in Figure 8.

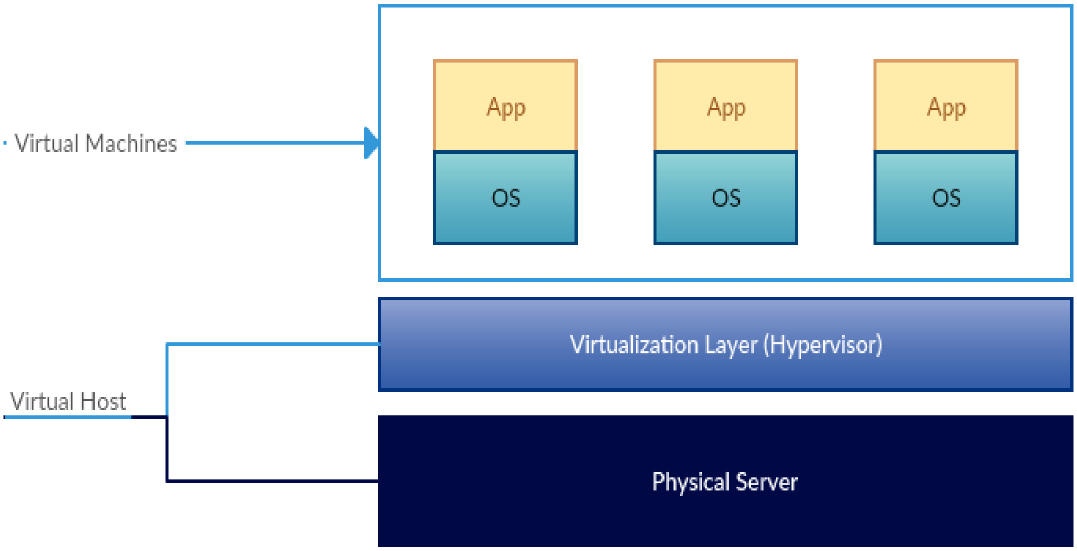


Figure 8: Server Virtualization Model. Reproduced from [23]

The hypervisor contains the Virtual Machine Manager which controls the Virtual Machines (VM) running on a physical server. Hypervisors can be loaded immediately on the hardware such as Hyper-V and XenServer or they can be loaded as an application running over an installed OS such as Virtual Server and Fusion. In the latter case, the performance is lowered due to the additional OS layer and the storage is also minimized but it is nonetheless useful in rendering a PC into a server and managing different virtual servers. Mainly, there are three types of server virtualization: *Hardware Emulation, Para virtualization*, and *Operating System Virtualization* as illustrated in Figure 8.

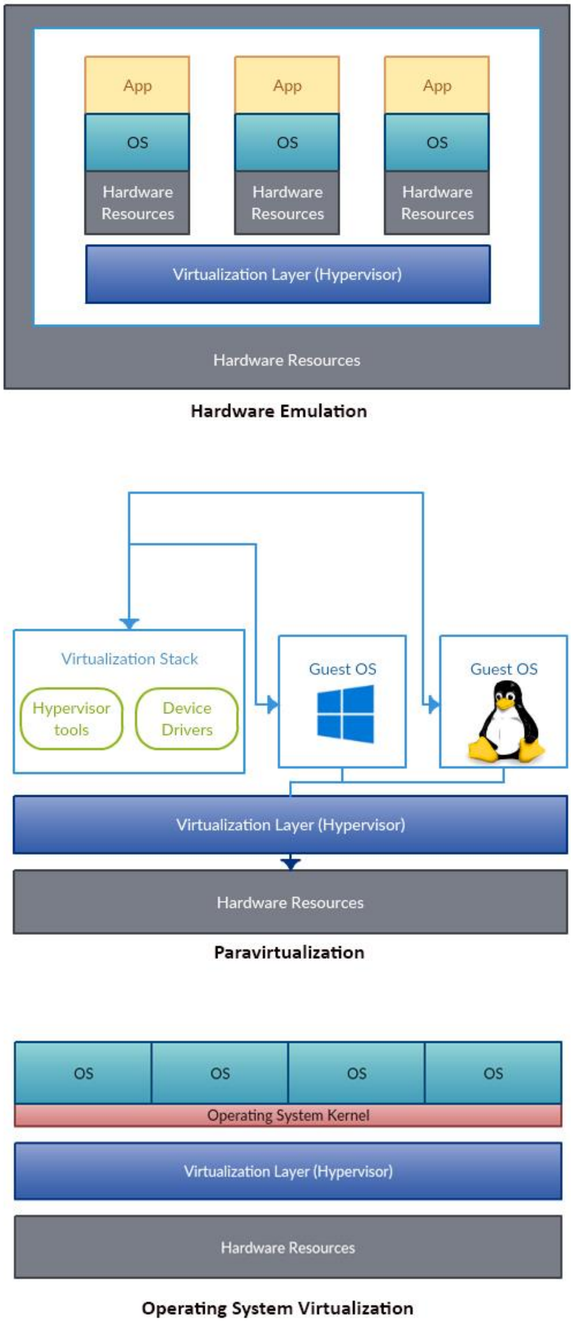
In Hardware Emulation, the hypervisor emulates the hardware resources to create a virtual machine. When the operating system installed on the virtual machine makes calls for resources, the hardware emulation software redirects the calls to the hypervisor. Then, the hypervisor redirects these calls to the actual physical hardware resources. In Para virtualization, the hypervisor coordinates the access of a guest operating system to the actual hardware environment directly. In Operating System Virtualization, the kernel of the host operating system is partitioned into several isolated spaces where virtual machines can be installed.

In 2009, it was announced that the number of virtual servers being purchased was exceeding the number of physical servers. Server Virtualization has become the de facto technique in datacenters because it substantially reduces power consumption, cuts off energy and cooling systems' costs and provides a more efficient way to manage resources [24]. Server Virtualization also offers flexibility to business processes. If a physical server needs maintenance, the virtual servers hosted by that server can be immigrated to another server and the original server can be switched off for maintenance. Some of the popular available virtualization technologies include:

- **Microsoft's Hyper-V**: A product that offers similar features to that of Vmware but uses Operating System virtualization. It offers enhanced security options by including just-in-time administrative privileges and shielded virtual machines.

- **XenServer:** Critix offers this technology which uses Paravirtualization. XenServer is an open source project coordinated by Citrix and it is widely adopted by the enterprise.

Using Virtualization technologies in datacenters can remarkably reduce power consumption and CO2 emissions resulting in cost savings. The following table 1 presents several examples to illustrate the key benefits of Virtualization. It draws a comparison in energy costs of using physical servers against virtual servers. Table 2 illustrates the cost and CO2 emissions showing how it is possible to go greener by going virtual.



Hardware Emulation

Paravirtualization

Operating System Virtualization

Figure 9: Server Virtualization Types: Hardware Emulation, Paravirtualzation, and Operating System Virtualization. Reproduced from [23]

Table 1: Comparison between the Energy Consumption of Physical Servers and Virtual Servers

Number of Servers (in thousands)	Energy consumption (kWh) Physical (in millions)	Energy consumption (kWh) Virtual (in millions)	Savings (kWh) (in millions)
20	151	18	132
10	76	9	66
5	38	5	33
1	8	0.9	6

Table 2: Comparison between Cost and CO2 Emissions of Physical Servers and Virtual Servers

Number of Servers (in thousands)	Cost ($) Physical (in millions)	Cost ($) Virtual (in millions)	Savings($) (in millions)	CO2 Emissions (lbs) (in millions)
20	15	1.8	13	209
10	7	0.94	6.6	104
5	3.7	0.47	3.3	52
1	0.75	0.94	0.66	10.4

Challenges

In the era of big data, Internet of Things and cloud computing, low-power solutions has gained paramount importance. However, these solutions face the *power vs performance* trade off. Usually, power can be reduced but the performance is degraded as a consequence. When performance has priority over power consumption, the natural choice is to increase the power so that the performance does not suffer i.e performance degradation is acceptable up to a certain level. Another challenge for lower-power electronics is *power vs flexibility*. When it comes to power efficiency, application specific hardware offers a great deal of power reduction. However, programming-wise, this constrains the flexibility.

Yet another challenge is the heat generated by electronics. Not long ago, researchers acknowledged that heat dissipation needs to be considered as an issue on its own, rather than a by-product of power dissipation issue. This has led to the rise of *Temperature-aware computing* as a separate field from Power-aware computing[1].

Conclusions

Driven by Moore's Law, the IT community has enjoyed the high performance by increasing the number of transistors per chip. This rapid growth in speed and efficiency, however, created a new problem of high power density in electronics. The future of electronics should no longer emphasize performance at the expense of power. Low-power computing offers the best of the two worlds. With less power consumption and dissipation, power costs will decrease, battery life will increase, and the impact of electronics on the environment will be minimized. Electronics engineers, as well as computer scientists, are working on developing power-efficient solutions starting from the very beginning by designing circuits that draw less power, designing power-efficient algorithms, creating power-aware programs. Also, datacenters being the most consumer of power in the IT are having their fair share of power-efficient solutions led by server virtualization technology. As the world heads towards the Internet of Things, the future is indeed for low-power electronics.

References

i. Benini, L., De Micheli, G. and Macii, E. (2001) 'Designing low-power circuits: Practical recipes', IEEE Circuits and Systems Magazine, 1(1), pp. 6–25.

ii. Stan, M.R. and Skadron, K. (2003) 'Power-aware computing', Computer, 36(12), pp. 35–38.

iii. V.Nandakumar, S.S.Dhenakaran, 'A Six Point Charter of Green Computing', IJCA, Vol.71, June 2013.

iv. V.Chithra, K.Jayasree, V.Anand, 'A Study of Green Computing Techniques', IJCSITR, Vol.2, Issue.2, June 2014.

v. K.Shalini, K.Naga, JTI, 'Green Computing', Vol.1, Issue.1, March 2013.

vi. P.Mittal, N.Kaur, 'Green Computing – Need and Implementation', IJARCET, Vol.2, Issue.3, March 2013.

vii. M.Anwar, M.Abdulkareem, J.Abdullah, 'Green Computing Beyond the traditional ways', IJMCR, India, Vol.3, Aug 2015.

viii. Wong, Samuel, 'Dynamic power management for faster, more efficient battery charging', Texas Instruments Incorporated, Analog Applications Journal, 44, 2013.

ix. Ace Cloud Hosting, 'Green Computing: The Environmental Benefits of Going Green', 2015.

x. L.Dhanam, S.Sivaraj, R.Vanathi, 'Green Computing For System Efficiency and Optimization', IJIRCCE, Vol.2, Issue.1, March 2014.

xi. Patil, A. and Goti, A. (2013) 'Green computing in communication networks', International Journal of Computer Applications, 68(23), pp. 1–5.

xii. Sinha, A. and Chandrakasan, A. (2001) 'Dynamic power management in wireless sensor networks',IEEE Design & Test of Computers, 18(2), pp. 62–74. doi: 10.1109/54.914626.

xiii. Mudge, Trevor, Blaauw, David, 'Creating a Wireless Future: CSE Gets an ARM up on Low Power Mobile Computing', The university of Michigan, 2006.

xiv. P.R. Panda, 'Basic Low Power Digital Design', Springer, Springer Science Business Media, 2010

xv. IBM and MontaVista Software, 'Dynamic Power Management for Embedded Systems', 2002.

xvi. Benini, Luca, Bogliolo, Alessandro, Micheli, 'A Survey of Design Techniques for System-level Dynamic Power Management', IEEE Transactions on Very Large Scale Integration (VLSI) Systems Vol.8 , Issue.3 , 2000.

xvii. Lu, Yung-Hsiang, Micheli, Giovanni, 'Comparing System-Level Power Management Policies', IEEE Design &Test, Vol.18, Issue.2, 2001.

xviii. Suman, Soniya, Mehak, 'A Greener Approach to Computing', IJARCSSE, Vol.2, Issue.2, February 2012.

xix. Oguchi, Yamamoto, 'Server Virtualization Technology and its Latest Trends', Fujitsu Sci. Tech, Vol.1, Journal.44, 2008.

xx. V.Ch-ithra, K.Jayasree, V.Anand, 'A Study of Green Computing Techniques', IJCSITR, Vol.2, Issue.2, June 2014.

xxi. S.Shinde, S.Nalawade, A.Nalawade, 'Green Computing: Go Green and Save Energy', IJARCSSE, Vol.3, Issue.7, July 2013.

xxii. Ponnusamy, Yuvapriya, 'Application of Green Cloud Computing for Efficient Resource Energy Management in Data Centers', IJCSIT, Vol.3, Journal.5, 2012.

xxiii. Golden, Bernard, 'Virtualization for Dummies', HP and Publishing Inc, 2011.

xxiv. K.Shalini, K.Naga, JTI, 'Green Computing', Vol.1, Issue.1, March 2013.