# New Horizon of Energy Efficiency in Sorting Algorithms: Green Computing

Tej Bahadur Chandra, V.K. Patle, Sanjay Kumar

{*tejbahadur1990@gmail.com, patlevinod@gmail.com, sanraipur@rediffmail.com* }
School of Studies in Computer in Computer Science & IT,
Pt. Ravishankar Shukla University, Raipur, India

**Abstract:** *Reducing energy consumption by using optimized programming techniques and energy efficient algorithms in order to maximize the uptime of various battery operated devices has been an interesting research topic over the past several years. Initially the computing was performed on the basis of lesser space and quicker calculation. But in recent past the energy consumption is one of the major driving factor for selecting the computing algorithms to attain the power efficiency and Green Computing. A lot of optimization has been made by manufacturers to minimize the energy consumption in hardware but it depends on internal algorithm efficiency too. This paper, mainly focuses on energy efficiency of some standard sorting algorithms, which are not only used in various devices directly but also very often implicitly by other algorithms. Experiments shows, that different sorting algorithms have different energy consumptions.*

**Keywords:** Sorting Algorithms, Energy Efficiency, Joulemeter, Green Computing.

## 1. Introduction:

The increasing dependencies on the computer system profoundly affects the environment. A lot of researchers are engaged in optimizing the energy efficiency of the hardware components. However, algorithmic energy efficiency also has great impact.

Sorting algorithms have been ubiquitously used in several applications nowadays. As the data size goes on increasing, energy-efficiency is gradually becomes equally important. In this paper, we analyze the performance and energy consumption of five standard sorting algorithms, bubble sort, insertion sort, selection sort, quick sort, and quick2 sort on Intel Core-i3 (2.40 GHz) processor. The main goal of this paper is to stimulate the use of energy efficient sorting algorithms that will contribute in green computing.

## 2. Related Work:

Due to complex and costly hardware requirements for computing the energy consumption of the specific algorithm, only few researches has been conducted in this field. Researchers has concluded that insertion sort is the best, keeping an optimal balance between energy consumption and processing speed during multi-party mobile communication [1]. Also the energy consumption greatly depends on the time as well as space complexity of the algorithms [2]. Peter P. Puschner compared the average execution time of the standard algorithms and investigate the hard and soft real-time performance of sorting algorithms and compared it to their average performance [3]. Michael N. Skaredoff has discussed the standard O notation, general principals of computing efficiency, advantages, disadvantages of each variant and method that may enhance the speed and resource utilization [4]. Research efforts are classified into four levels of abstraction; the logic design level, the processor level, the operating system level and the compiler level [5].

Various researches are going-on in the hardware category which attempts to optimize the energy consumption by investigating the hardware usage [6].

## 3. Sorting Algorithms:

Since the emergence of computing, the sorting problem has attracted a great deal of research, perhaps due to the complexity of solving it efficiently despite its simple, familiar statement. Sorting algorithms appear very simple but its machine implementation is too much complex. In following we describe the various widespread sorting algorithms that are used in the context of this study.

**Bubble Sort:** It is a simple comparison sorting algorithm. It works by repeatedly stepping through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. First pass bubble out the largest element to the last position and second pass place the second largest in the second last position & so on [7].

Worst-Case and
Average Case complexity = $O(n^2)$ *
Memory Complexity = $O(n)$.

**Insertion Sort**: It is also a comparison sorting algorithm and is relatively efficient for small lists and mostly sorted lists. It is often used as part of more sophisticated algorithms. Every iteration of insertion sort removes an element from the input data, inserting it into the correct position in the already-sorted list, until no input elements remain [7, 8].

In general case Time Complexity = $O(n^2)$
Its average Time Complexity = $O(n^2/4)$
In best case Time Complexity = $O(n)$
In All cases Memory Complexity = $O(n)$

* *n is the number of items being sorted.*

**Selection Sort:** It is a simple in-place comparison sorting algorithm that works by finding the minimum value in the list and swap it with the value in the first position for all the elements. It is quite inefficient for sorting large data volumes and generally performs worse than the similar insertion sort. Selection sort is notable for its programming simplicity also has performance advantages over more complicated algorithms in certain situations [7, 8].

In all cases
Time Complexity = $O(n^2)$
Space complexity = $O(1)$

**Quick Sort:** *Quicksort* is a divide and conquer algorithm which relies on a *partition* operation: to partition an array an element called a *pivot* is selected. All elements smaller than the pivot are moved before it and all greater elements are moved after it. The lesser and greater sub-lists are then recursively sorted. Quick sort is among the fastest sorting algorithms in practice [7].

Time Complexity
In Average case = $O(n \log n)$
In worst case = $O(n^2)$
Space Complexity
In Average case = $\log n$
In worst case = $n$

**Quick2 Sort:** It is the optimized version of the quick sort. On reason for its speed is that its inner loop is very short and can be optimized very well. It is also called MedianHybridQuickSort. The optimized version chooses the median of the first, last and middle items of the partition as its pivot, and it stops partitioning when the partition size is less than sixteen [9].

## 4. Role of Algorithms in Green Computing

The efficient use of computers and computing is what green computing is all about [10]. Green computing aims at to minimize the impacts of Global warming by reducing the energy consumption, minimizing the use of hazardous materials, and promote the recyclability or biodegradability of outdated products and factory waste [11 to 15].

Operational energy required in the computer system is not only dependent on the hardware components but the algorithms we use also has a great impact on it. When we are working on battery powered devices or highly conscious about green computing then energy efficiency of algorithm is major factor concerning energy consumption and performance. Use of efficient algorithms will reduce the time as well as space required and ultimately reduce the energy consumption and hence will contribute to green computing [2]. In current scenario energy consumption has become one of the major issues for battery powered computing devices and thus in green computing [16].

## 5. Experimental Design

The experimental setup for this study uses a simulator tool "*Joulemeter*" from Microsoft Corporation for computing energy consumption of the sorting algorithms [17]. We have also developed a software for Sorting Algorithms in Visual Basic 6.0 that provide visual interface for executing sorting algorithms and compute the time required in each individual sort [18].

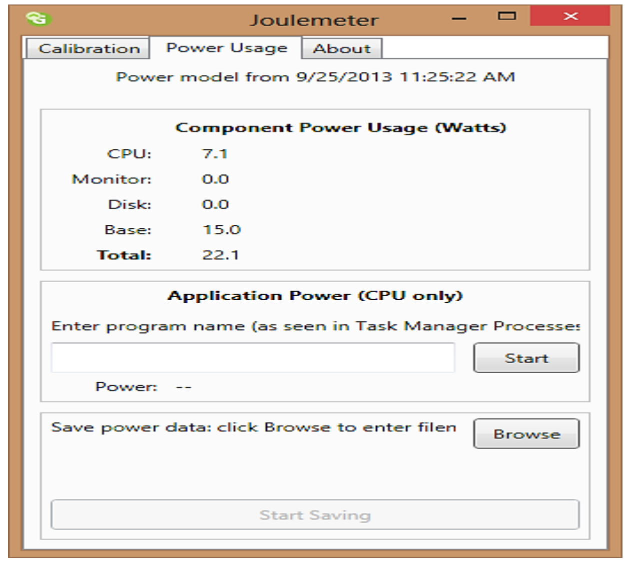
### 5.1. Joulemeter

Joulemeter is a software tool to view the power consumption of your computer as a whole as well as for the key hardware components. We can also compute the power consumption due to a specific application. The data can also be stored periodically to a file if desired. Following figure shows a screenshot of the power metering interface exposed by Joulemeter.[17]

For Joulemeter to estimates the power usage, we need to calibrate it. Once the calibration is over, the total energy use for the entire run may be simply obtained by adding up the power values (in Watts) for the duration of interest. Since each value is the power use over one second, the sum is the energy use in Joules. Joules can be converted to other units such as Watts-hours or kW-hours using the following conversion factors:

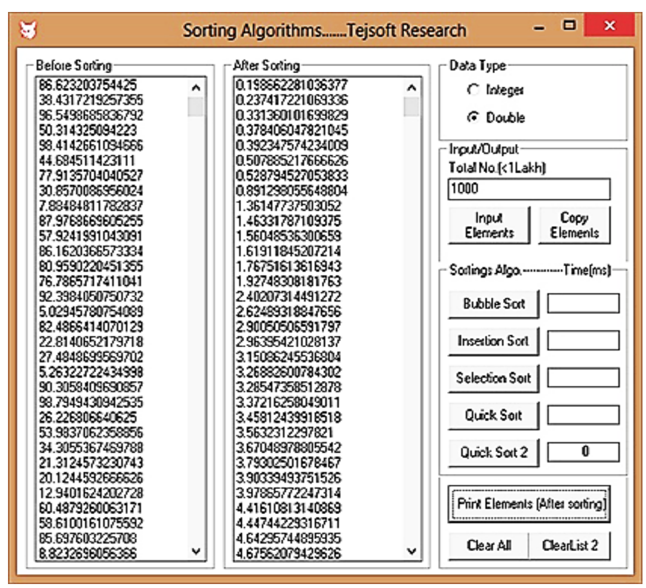
$$
\begin{aligned}
1\ \text{kWh} &= 1000\ \text{Watts} * 1\ \text{Hour} \\
&= 1000\ \text{W} * 3600\ \text{s} \\
&= 3{,}600{,}000\ \text{Joules.}
\end{aligned}
$$

Thus, $1\ \text{Joule} = 1/3{,}600{,}000\ \text{kWh}$.




### 5.2. Sorting algorithm software

This software is developed in Visual Basic 6.0 which implements five sorting algorithms in separate procedures and allow sorting on the elements of integer and Double data types. This application software also computes the total time (in ms) required in individual sorting algorithms [18]. Following is the screenshot:



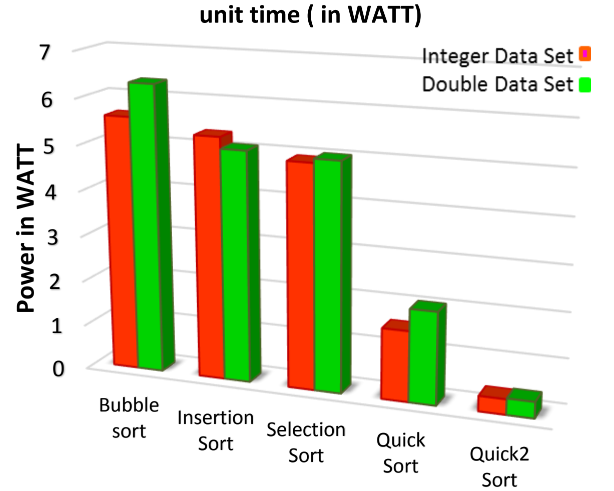
### 5.3. Experimental Runs

In this experiment, we have compared five different sorting algorithms and computed their performance on the basis of energy consumed and computing time required. First of all, we select ten thousand integer data and perform six tests of each sorting algorithm and repeat the same experiment for another ten thousand integer data. Same above procedure is also repeated for the double data type elements. Power data at each second for every run is collected in a file.

## 6. Experimental Result and Analysis

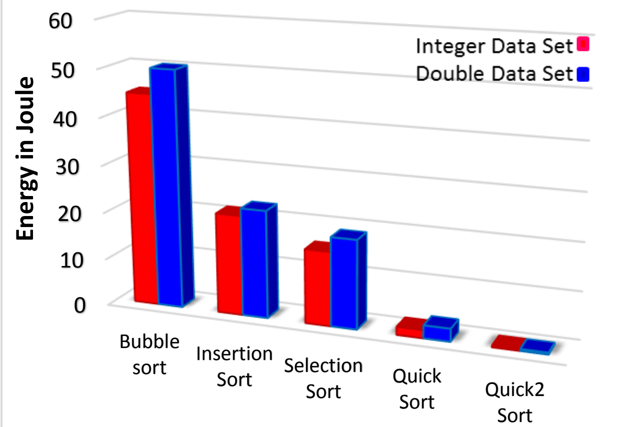
Here, we discussed the energy consumption of different sorting algorithms on ten thousand data sets of integer and double data types. The collected records are analyzed from three perspectives. First, the average energy consumption of the sorting algorithms are measured in watt for each time unit during its total run time. Second, the total energy consumed for each sorting algorithm is computed in joule per second. Third, we also calculate the total time required in milliseconds for performing the sorting operation.

**Figure 1. Average Power Consumed per unit time ( in WATT)**



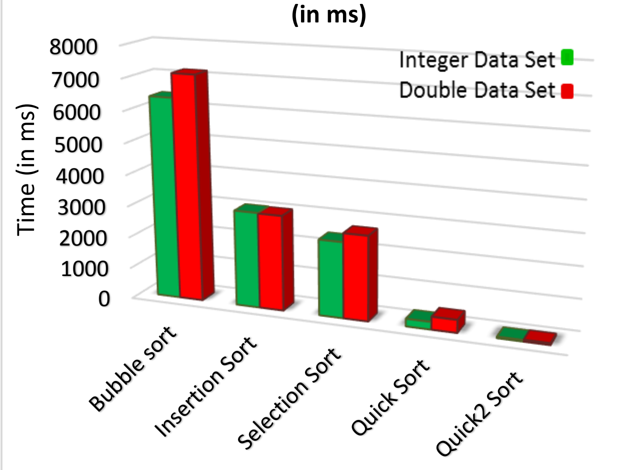

In figure 1, we note that each sorting algorithm consumes different power per unit time (here it is taken as one second). It is found that bubble sort consumes average 5.602 watt on integer and 6.335 watt on double type data sets. On other hand optimized quick sort consumes lowest power (average 0.3375 watt on integer and 0.3583 watt on double type) for sorting.

**Figure 2. Total Energy Consumed (in Joule)**




In figure 2, we observe that the total energy required for sorting the same data set is highest when we use bubble sort (average 44.82 J on integer type and 50.15 J on double type) followed by insertion sort, selection sort, quick sort and optimized quick sort require least energy (average 0.357 J on integer type and 0.358 J on double type).

**Figure 3. Average time required for sorting (in ms)**




Also, if we compare the sorting algorithms with respect to time then optimized quick sort takes the least time to sort the elements then all other discussed here.

### 6.1. Result Analysis

On the basis of the above experiments it is found that, the energy consumption of sorting algorithms depends mainly on time and space complexity of that algorithm, which means algorithms that needs $O(n^2)$ comparisons requires more energy than that of algorithms with $O(n \log n)$ comparisons. It is also observed that the data type of the elements being sorted have major impact on the energy consumption. The more energy is required for sorting data sets of double type than that of integer type. This might be due to involvement of more registers during floating point operations or realization of floating point unit (FPU) of the processor [16].

## 7. Conclusion

Green Computing is a little step to save our environment by using such techniques and algorithms that are highly efficient, cost effective and consume less power.

This paper mainly focuses on power efficiency of the sorting algorithms. Based on the experiments we find that optimized quick sort requires least time as well as consume least energy than other sorting algorithms discussed here. Also the experiment shows that the energy consumption depends not only on the sorting algorithms but also on the data types of the elements being sorted. The integer type data requires less energy than that of double type data.

The overall goal of this paper is promote the use of energy efficient algorithms that contribute in green computing.

## 8. References

[1]. Christian Bunse, Hagen H¨opfner, Suman Roychoudhury and Essam Mansour — choosing the ”best” sorting algorithm for optimal energy consumption, h*ttp://elab.ws/portal/files/publications/icsoft09-energy.pdf*, accessed during October 2013

[2]. Bhavana Narain and Sanjay Kumar — Impact Of Algorithms On Green Computing, *International Journal of Computer Applications (2013)*, ISSN No. 0975-8887

[3]. Peter P. Puschner — Real-Time Performance of Sorting Algorithms, *The International Journal of Time-Critical Computing Systems (1999)*, Vol 16, Issue 1 pp. 63–79.

[4]. Michael N. Skaredoff — Sorting Efficiency, *Journal of Clinical Monitoring (1987).*

[5]. R. Jain, D. Molnar, and Z. Ramzan — Towards understanding algorithmic factors affecting energy consumption: switching complexity, randomness, and preliminary experiments, *Proc. of the 2005 joint workshop on foundations of mobile computing, ACM.*

[6] C. Bunse and H. H¨opfner — Resource substitution with components Optimizing Energy Consumption. *Proc. Of the 3rd ICSOFT, volume SE/GSDCA/MUSE*, pp. 28–35, Setubal, Portugal, July 2008. INSTICC press.

[7]. G.S. Baluja, *Data Structure through C,* Second Edition 2005, pp -10.1 to 10.15, Dhanpat Rai & Co. (Pvt.) Ltd. Educational and technical publishers, Delhi.

[8]. Seymour Lipschut and G A Vijayalakshmi Pai, *Data Structures,* pp. - 9.6 to 9.8, Tata McGraw-Hill Publishing Company limited New Delhi. 2006

[9]. http://warp.povusers.org/SortComparison/q2sort.html, accessed during October 2013.

[10]. Pardeep Seelwal — Current Trends in Green Computing, *International Indexed & Refereed Research Journal*, ISSN 0975-3486, (Print) E-ISSN-2320-5482, July,2013 VOL-IV *ISSUE- 46

[11]. Khalid Raza1, V. K. Patle, and Sandeep Arya— A Review on Green Computing for Eco-Friendly and Sustainable IT, *Journal of Computational Intelligence and Electronic*

*Systems*, Vol. 1, 1–14, 2012, pp. 2-4, Printed in the United States of America

[12]. G. Saranya and S.P. Ponusamy — Green Computing and its application, *National conference on recent trends in computer application and technology*. http://ncrtcat.comyr.com/paper/NCRTCAT13P031.pdf, accessed during October 2013

[13]. Sanghita Roy and Manigrib Bag — Green Computing - New Horizon of Energy Efficiency and E-Waste Minimization - World Perspective vis-à-vis Indian Scenario, IEEE, Communication Systems and Network Technologies (CSNT), 2011, pp. 688 - 693, E-ISBN: 978-0-7695-4437-3, Print ISBN: 978-1-4577-0543-4

[14]. Namrata Agrawal and Kamal Nayan Agarwal— Current Trends in Green ICT, Current Trends in Green ICT. JOAAG, Vol. 7. No. 1.

[15]. Surendra Kumar Patel, Anil Kumar Sharma, Gupteshwar Gupta and Sanjay Kumar— High Performance Computing: Grid Computing and Cloud Computing for Green ICT, *International Journal of Computer Applications (0975 – 8887)*, ISSN No. 0975-8887

[16]. Christian Bunse, Hagen H¨opfner, Essam Mansour and Suman Roychoudhury — Exploring the Energy Consumption of Data Sorting Algorithms in Embedded and Mobile Environments,*IEEE*, Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09, pp. 600 - 607, E-ISBN -978-0-7695-3650-7, Print ISBN- 978-1-4244-4153-2

[17]. Joulemeter *user manua*, Version 1.1, http://research.microsoft.com/joulemeter/, accessed on October 2013.

[18]. Steven Holzner , Visual Basic 6 Black Book,*Publisher: The Coriolis Group*, ISBN: 1576102831, Publication Date: 08/01/98.