Heterogeneity: the key to achieve Power-Proportional Computing

Georges Da Costa
IRIT (UMR CNRS)
Université de Toulouse
Toulouse, France
Email: dacosta@irit.fr

Presented at
2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing
http://doi.ieeecomputersociety.org/10.1109/CCGrid.2013.90

Abstract

The Smart 2020 report on low carbon economy in the information age shows that 2% of the global CO_2 footprint will come from ICT in 2020. Out of these, 18% will be caused by data-centers, while 45% will come from personal computers.

Classical research to reduce this footprint usually focuses on new consolidation techniques for global data-centers. In reality, personal computers and private computing infrastructures are here to stay. They are subject to irregular workload, and are usually largely under-loaded.

Most of these computers waste tremendous amount of energy as nearly half of their maximum power consumption comes from simply being switched on. The ideal situation would be to use proportional computers that use nearly 0W when lightly loaded.

This article shows the gains of using a perfectly proportional hardware on different type of data-centers: 50% gains for the servers used during 98 World Cup, 20% to the already optimized Google servers. Gains would attain up to 80% for personal computers.

As such perfect hardware still does not exist, a real platform composed of Intel I7, Intel Atom and Raspberry Pi is evaluated. Using this infrastructure, gains are of 20% for the World Cup data-center, 5% for Google data-centers and up to 60% for personal computers.

keywords: **Energy efficiency, Power proportional, Heterogeneous architectures, Data-centers, Large scale**

1 Introduction

Lots of current energy-efficient computing research are based on an assumption: The future is a giant cloud globally consolidated. In this case energy efficiency would improve greatly as all servers would be loaded at the level where they reach their most energy-efficient profile.

In reality there still exist a large number of private computing infrastructures due to several reasons such as private data, or the will to manage locally those infrastructures for example. Even if energy-wise this solution is far from optimal, it seems unrealistic to consider that all those private infrastructures will be merged in a global cloud simply because of the energy-efficiency of this solution.

These private infrastructures are often subject to irregular workloads, as well as low workloads as they are not large enough to aggregate large amount of different workloads. These infrastructures are often small, and as they are usually build step by step and partially upgraded, they are often composed of several types of hardware.

It is even more the case as dedicated hardware systems are being build for particular usages, such as big-data processing.

For these types of infrastructure, having large and powerful servers leads to inefficiency most of the time as their basic power consumption is high for a low number of requests.

But this assumption also forgets that most of these resources are in fact accessed through an even less efficient system: personal computers or laptops, of companies or even of home users.

The ideal situation would be to use proportional computers that use nearly $0W$ when lightly loaded instead of having half their maximum power consumption just by being switched on. For a server, always being ready to handle the maximum load does not follow the principle of least effort [12] and leads to large waste due to the large amount of chipset, buses, memory banks,..., switched on but scarcely used.

Achieving this goal would have two tremendous effects. First it would reduce the overall energy consumption of data centers, but more importantly it would reduce the need of such infrastructures. One of the goal of a data-center is to aggregate enough workload so the static power consumption is negligible. Having real power-proportional hardware will reduce this fact and will enable the possibility of more pervasive infrastructures. In this case, cooling will no more be mandatory as the density of computing resources will no more be a goal.

Proportional computing will have an even more important impact on personal computers and laptops. Indeed [10] shows that in offices personal computers are idle more than 75% of the time. From a power point of view, it means most resources are consumed just on the static part. Here potential gains are of one order of magnitude !

In the following, this article will show how to attain such a system using only existing hardware (Section 3) after exploring the state of the art (Section 2). Finally it will demonstrate (Section 4) on the worst case (data-center) that large amount of energy can be saved with simple and available technology.

2 State of the art

In [3], authors proposed the concept of *energy proportional computing.* They analyzed Google commodity servers and observed that traditional servers consume half of their maximum power while being idle. Statistics of Google servers usage show that usually utilization is low. So the wasted energy is large. In [2], same authors also indicate that for modern distributed systems, the aggregated cost/performance ratio of an entire system takes a huge impact due to the idle power-consumption.

Indeed, in current systems, servers operate rarely at high load (were they are energy-efficient) or are rarely completely idle (were it could be possible to switch them off). They are instead operating most of the time between 10 and 50 percent of their maximum utilization levels [6].

Two types of initiatives exists to reach proportional power consumption: Dark silicon and heterogeneous on-die cores. While the first one is still far-reached, the second one is already available on the market.

Moore law predicts that transistor density would double every two years. It was achieved at the price of power-consumption as the later improved several order of time slower. As more and more transistors are integrated on dies, and as increasing density increases also electricity leakage, processors base consumption remains high. The dynamic part, linked with the actual workload, is often of the same order of magnitude of this static part. The static part would be way lower if processors could switch off unused processing units. The concept of *Dark Silicon*[5, 8] is to improve the dynamism of processor and to power only the utilized parts. This can drastically improve efficiency of processors at low load. But it does not solve totally the problem as usually there is also a fixed power consumption for the motherboard that is not negligible. Motherboard are designed by taking into account the maximum needs of processors. Also, compared to processors, they are often quite limited on the possibility to change their power-consumption in function of their usage.

The second method is to integrate heterogeneous on-die cores. A lightweight core will be switched on as long as the load is low, and an increased workload will be transferred to a more powerful core at run-time. This method is implemented in the Big.LITTLE ARM technology[7] which integrates a light ARM Cortex A7 with a more powerful but more power-hungry ARM Cortex A15. In the standard mode, all applications will run on only one core, and will be migrated depending on the load. The key element of this system is the migration time which must be low so the transition has a reduced impact. In the Cortex-A15-Cortex-A7 case, this migration takes less than 20,000 cycles, or 20μs (at 1GHz). In the future, ARM will provide more elastic hardware, integrating a GPU (Mali T604), allowing for a broader range of possibilities depending on the load.

For domain specific hardware, it is also possible to have several version of the same hardware, switched on depending on the load. NVIDIA Optimus graphic system[11] can use CPU-integrated video chipset when the graphic workload is low and automatically switch to a full-fledged GPU when this workload increases. In this case, the *migration* lasts 1/5th of a frame.

Integrating heterogeneous on-die cores has limits as the motherboard still has to provide facilities (buses, network, ...) for all cores at all time, wasting electricity. Also in a data-center, it is not necessary to have this type of plasticity for all servers as most of them will only serve on case of high workload.

3 Achieving power-proportional computing

The goal of power-proportionality is to have the same power-efficiency whichever the load. In this case, with a nearly zero load, power consumption would be also nearly zero. It would render unnecessary consolidation. The largest challenge to overcome is the power consumption on idle which is often nearly half the power consumption on full load.

Figure 1 shows an increase of load and the resulting power consumption for two types of system: A real Intel I7 and an ideal power-proportional hardware with the same efficiency at high load. In this case, load is achieved by increasing the web workload on a web server located in the monitored computer. This example uses a complex web service. In this case the computing resource is preponderant compared to communications. The web-server is the multi-threaded *lighttpd*. Workload is generated on an external computer using the web benchmark tool *siege*[1].

Results would be equivalent for other type of load. In the following, as an example, load will always be of this type.

In a data-center, at around 350 requests/s a second I7 would be started to answer to the increasing load.

Idle power-consumption leads to a very low efficiency for small number of requests (see Figure 2). When workload increases, energy efficiency increases also as the overhead is split on all the requests. An I7 server can serve up to 350 requests per seconds, consuming 42W. The problem is on low load as if there are only a handful of requests it still consumes the idle power consumption of 12W, nearly a third. Using the measures of [3] where authors measured the load distribution of Google servers, with a median workload of 30%, it consumes 21W whereas a perfectly proportional hardware would consume 13W, i.e. 40% less.

Figure 3 shows a zoom on the behavior of Intel I7 processor. Power measures were done between the power supply unit (PSU) and the motherboard. Each point in this figure (and Figure 4) was obtained by averaging 10 experiments. Going up to 100 experiments shows the same trend but smooths the curve as the standard deviation is small. Using only 10 experiments allows to have a feeling of the difference of standard error between power measures (3W) and the other measured values (negligible). The web pages were dynamic and putting a lot of burden on the processor. The Siege web benchmark software was used to load the web server.

Figure 3 shows two phases. The first one is when the number of clients is below 175. In this phase the I7 is able to cope with the requests, and power

[1] http://www.joedog.org/siege-home/

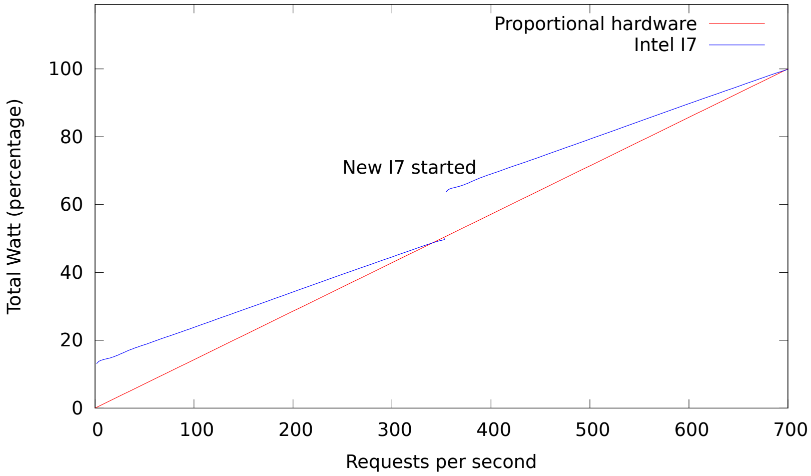

Figure 1: Measures of power consumption of a complex web service (computing power is preponderant on communication) on an Intel I7 (100 runs) and model of an ideally proportional hardware.

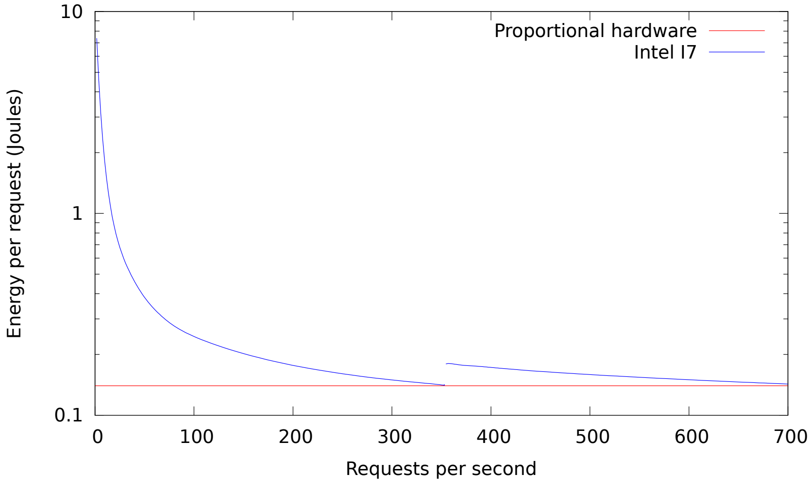

Figure 2: Energy efficiency of requests.

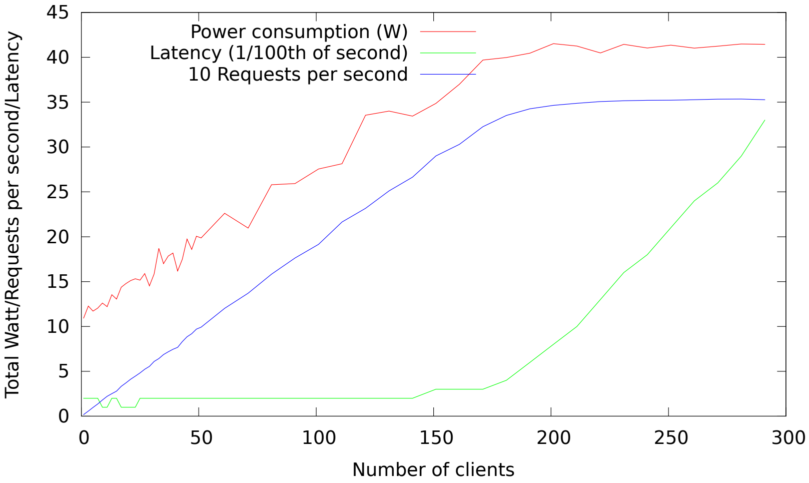

Figure 3: Zoom on Intel I7 serving a web workload. Scales are adapted to be visible. Points are average value of 10 experiments.

consumption grows linearly with the load, while latency remains stable. In the second phase, starting at 175 clients, the I7 reaches the limit of 350 requests per seconds and latency increases as no more requests can be served. During this time, power consumption remains stable.

Figure 4 shows the same experiment with an Intel Atom. The same behavior is visible. Here the maximum number of requests served is 35 requests per second. It has to be noted that the power consumption profile is the same but with large difference in the static/dynamic ratio. The static part is 8W and the dynamic part is 1W. It has to be compared with respectively 12W and 30W in the case of Intel I7. Nevertheless for a small amount of requests, i.e. less than 35, it is more interesting to use an Atom instead of an I7.

The same experiment was done for a Raspberry Pi B (ARM processor), and the same behavior was measured. The maximum number of requests served was 5.6 requests per seconds with a minimum power consumption of 2.6W and a maximum of 2.81W. Characteristics of all the processors are summarized in Table 5. The only difference is that for the Raspberry Pi, power consumption was measured between the wall and the PSU.

Different hardware have different profiles of power consumption and of performance. It has to be noted that usually in data-centers, servers are of different generations, composed of different types of hardware. Even the fact that they are at different distances to different types of cooling elements have an impact on their power consumption profile.

In order to reach power proportional computing, this heterogeneity is to be harnessed and even encouraged, in our example, adding Atom and Raspberry

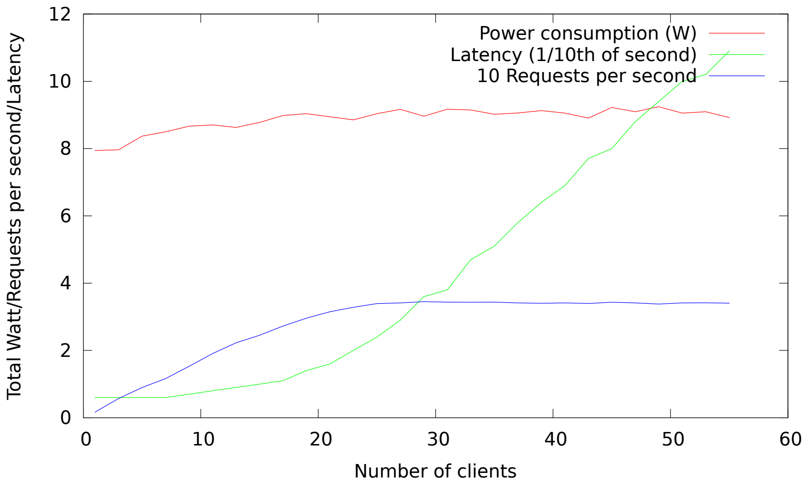

Figure 4: Zoom on Intel Atom serving the same web workload as in Figure 3. Latency is of 1/10th of a second to be visible.

Processor	Watt range	Max request/s	Efficiency
Intel I7	11 - 42	353	.12 W/r/s
Intel Atom	8 - 9	34	.26 W/r/s
Raspberry Pi	2.56 - 2.81	5.6	.50 W/r/s

Figure 5: Characteristics of Intel I7 and Atom, and of Raspberry Pi for the web workload

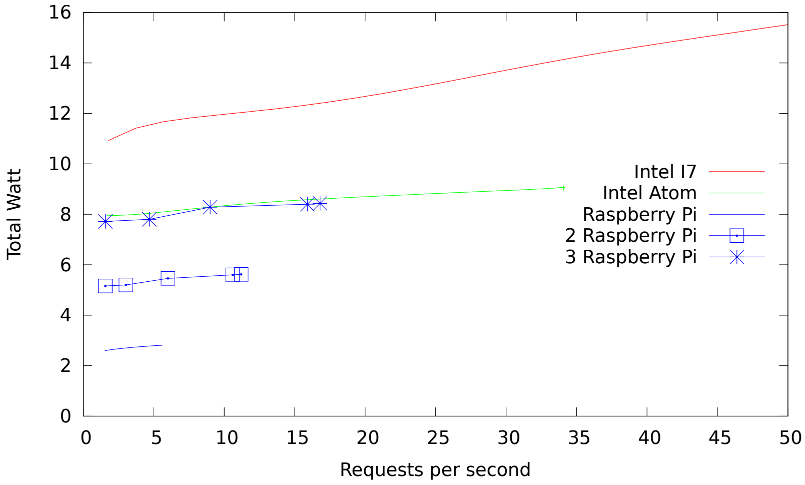

Figure 6: Comparison of efficiency in watt per number of requests for the different types of processors. Intel I7 is not represented completely (it can serves up to 350 requests per seconds.)

Pi to I7 servers.

In this case, it is possible to use nodes depending on actual load, not only on peak load. In a more general way, it is a simple bin-packing problem. It can be illustrated by Figure 6. The goal is to reduce power consumption and thus to chose the right combination of servers depending on the load. For example, aggregating three Raspberry Pi is quite equivalent to one Atom when workload is less than 17 requests per seconds.

In a more general way, to express the bin-packing problem, we can define C_i and P_i respectively the capacity and power of node type i at maximum load. C_i will serve as the capacity of bins. In this case, node type is *Intel I7*, *Intel Atom* and *Raspberry Pi*. Each node type can be selected an arbitrary number of times. The goal load will be C and it will be the number of unitary elements that the bin-packing will fit in the bins. C and C_i have to be large enough so the algorithm provide precise allocation (such as number of requests per seconds in our example). The objective function will be the weighted number of bins, each bin being weighted by its P_i. This algorithm selects the best nodes to be charged.

Using the bin-packing algorithm based on the data from Table 5, it is possible to use several small nodes and intermediary nodes to have a multi-scale smooth curve.

To obtain an optimal near-linear behavior with the available processors:

- 0→5 req/s 1 Raspberry Pi
- 5→10 req/s 2 Raspberry Pi

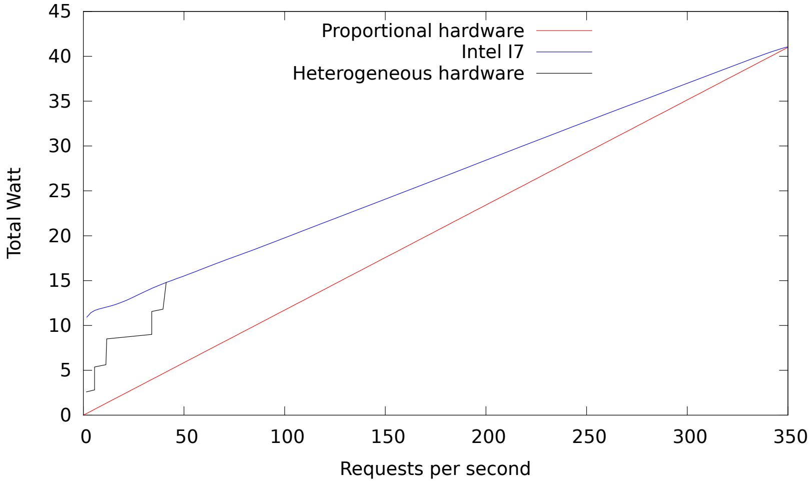

Figure 7: Power consumption in function of the load for the optimal combination of Raspberry Pi, Intel Atom and Intel I7.

- 10→35 req/s 1 Intel Atom
- 35→40 req/s 1 Intel Atom + 1 Raspberry Pi
- 40→350 req/s 1 Intel I7

Figure 7 shows the profile of power consumption for this aggregation. In this case, three phases are visible.

- On the left side, power consumption rises fast in function of the load but with a quite low static part.
- On the right side, power consumption is quite near the optimal one as the static power consumption is shared on a large number of requests.
- The central part, which is the largest, where using the available components does not allow to improve energy efficiency.

The achieved goal is visible on Figure 8 where the energy per request is clearly improved. The best improvement is for small numbers of requests as there is an improvement of nearly a factor 4 compared to Intel I7.

4 Experiments

To evaluate realistically the impact of this proportional method, access logs from the 1998 World Cup web site[9] will be used. The first access logs were collected on April 30th, 1998; the final access logs were collected on July 26th,

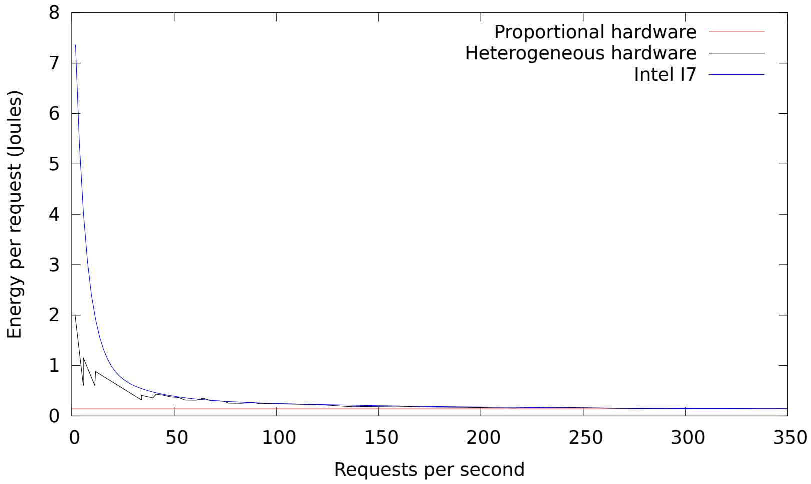

Figure 8: Energy efficiency of requests.

1998. During this 88 day period, 1,352,804,107 requests were received by the World Cup site.

During the collection period, there were four geographic locations: Paris, France; Plano, Texas; Herndon, Virginia; and Santa Clara, California. The World Cup data-set provides information about the localization of data-centers. Their localization (east, center and west of USA and Europe) were chosen in order to reduce latency. As the content was complex and evolving fast, this solution was chosen instead of using a classical CDN (content delivery network) infrastructure.

Each line in the data-set describes at which time the requests arrived (one second precision) and on which data-center. Two types of phases are visible on the data-set (Figure 9):

- Two low activity phases, first 40 days and last 10 days;
- One high activity phase, during the competition.

Please note that Figure 9 shows the aggregated numbers for each site but also the total numbers. Having only three date-centers would have been sufficient, or even one. To obtain a good latency several distributed data-centers are used. It implies that their average utilization is low compared to using only one data-center where there are more possibilities of workload consolidation.

4.1 One single data-center

In this section the system is considered as a single data-center. It is the worst case for proportional hardware as having all requests going to a single data-center reduces the need to be efficient during low activity phases.

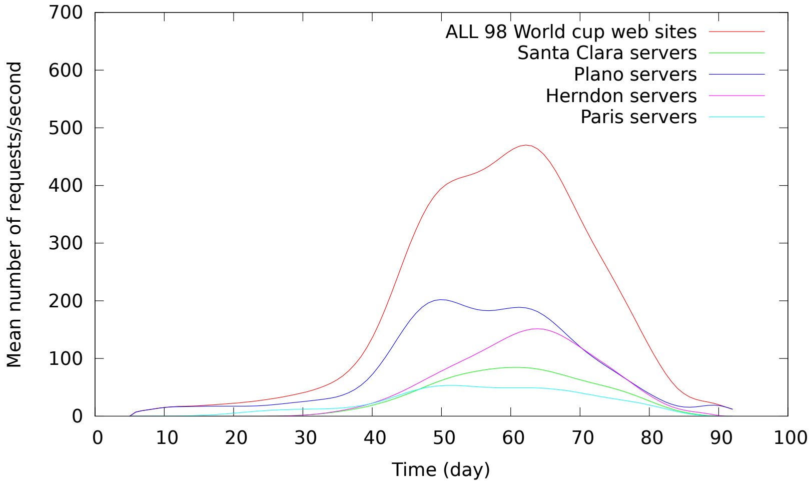

Figure 9: Accesses to the 98 World cup web site. Aggregated value and values for each region are displayed. It shows for each day the mean number of access per second.

Figure 10 shows the estimated power consumption of each data-center type following the assumption that they are all homogeneous except the one called *Heterogeneous hardware*.

For each data-center type, cooling is not taken into account, neither the energy-consumption of the load-balancer. The load-balancer is considered as a fixed cost as its behavior does not depends on the web servers that it serves. Energy needed by cooling is usually proportional to the computing infrastructure it cools, so comparison results are not impacted.

The *heterogeneous hardware* data-center is composed of the hardware proposed in the previous section: Two Raspberry Pi, one Intel Atom, a large number of Intel I7. This data-center load-balancer is assumed to follow the profile proposed at the end of the previous section.

Figure 10 shows the worst case for proportional hardware as having all requests going to a single data-center increases consolidation possibilities.

Even in this case, it is visible that *proportional hardware* is always better than the others, and can be better by one order of magnitude during low workload phases. This figure also shows that even if for very light workload Intel Atom and Raspberry Pi are quite efficient, this is more than compensated by their higher cost on high load.

Figure 11 shows a zoom on the best methods: *proportional hardware*, *heterogeneous hardware* and *Intel I7*.

It is interesting to note that even on high workload, a proportional hardware is able to save 5% of energy as for such a website workload is highly variable.

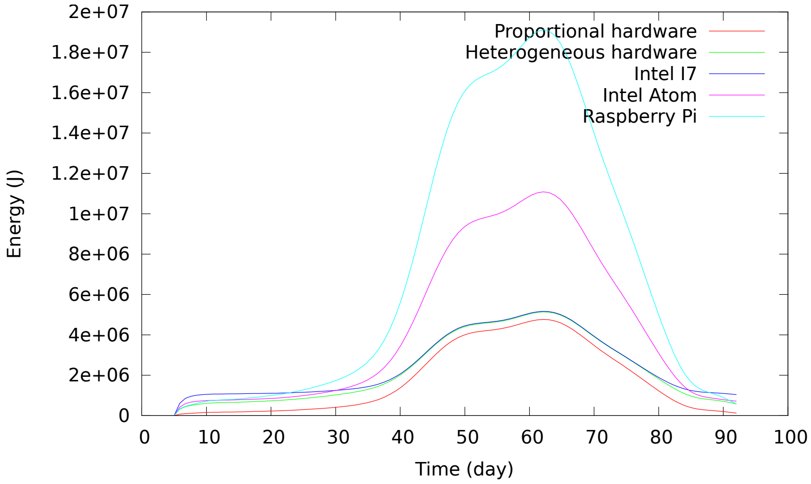

Figure 10: Energy consumed in the case of different types of data-centers. All the requests are assumed to go to a single data-center.

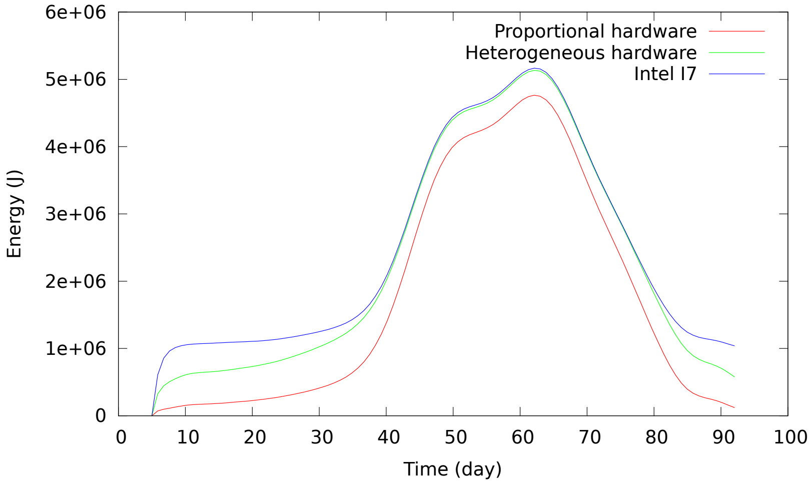

Figure 11: Zoom on the most efficient methods of Figure 10

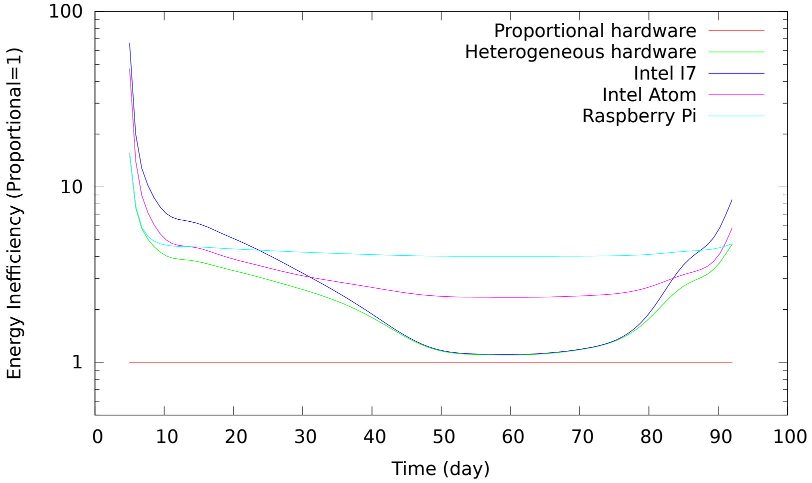

Figure 12: Comparison of the energy-cost per requests for all types of servers. Values shown are compared to ideal proportional hardware which is assigned a value of 1.

During the two low activity phases, the ratio of energy consumed by the perfect hardware and Intel I7 reaches an order of magnitude.

Using heterogeneous hardware shows an improvement on low activity phases were such servers consume 50% less energy than the homogeneous Intel I7 data-center. On high activity, the fact that the used combination of hardware is only able to improve energy efficiency on low number of requests (Figure 7) does not allow to save large amount of energy. At least, using heterogeneous hardware is always slightly more efficient than a I7 data-center.

From a global point of view, the total energy consumed by a proportional hardware and by the proposed heterogeneous hardware are respectively 35 % and 8% lower than the one consumed by an I7 data-center.

Figure 12 shows efficiency of each type of data-centers compared to proportional hardware. Using heterogeneous hardware allows to be more efficient than any homogeneous system. In this case the limiting factor is the hardware composing the mix. Depending on the activity level, even Raspberry Pi can be the most efficient.

4.2 Several data-centers

This section evaluates the gains of two configurations, having one single data-center, or keeping the original data-centers locations. The second one is more realistic at it helps guarantying a good quality of service. As a reminder, the current classical latency for Trans Atlantic communication is 80ms, communication in the North America network can go up to 40ms on the backbone

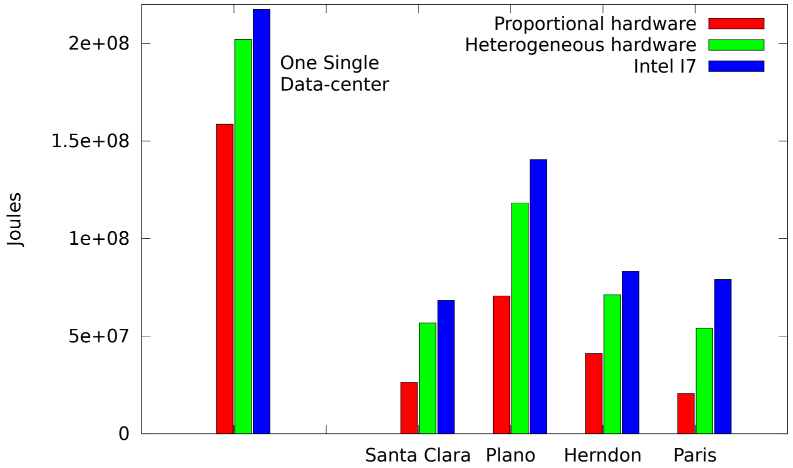

Figure 13: Comparison of total consumed energy for each type of data-center for each location. Fusion shows the energy consumed if all the traffic is redirected on a single location.

infrastructure. Oceania to Europe is more than 300ms. It is the reason that several data centers are spread around the world even if, from the energy-point of view, it is less efficient as it prevents consolidation.

Figure 13 shows the total energy consumed by the different data-centers in function of the technology. Proportional hardware is 2 to 2.6 times more efficient than Intel I7. The proposed heterogeneous hardware is 16 to 20% more efficient than Intel I7.

If the whole traffic was redirected to a single data-center, the gains would be respectively of 35% and 8%.

4.3 Extension to workstations and other type of servers

These results can be extended to workstations. In [4], it is evaluated on a university campus that the load of computers switched on and on which a student is logged have a mean load of 5.5%. Using a perfectly proportional system or an heterogeneous one would reduce power consumption by respectively 80% and 60% compared to an Intel I7. In this case it is assumed that during the whole time the computer is used during an interactive session, its load is switching from full load to idle load.

In [1], authors monitored several hundreds of workstations in several computer science departments. They showed that depending on the time of the day, between 40 and 80% of workstations were considered as Idle and switched on, the mean value being around 60%. Using a perfectly proportional system or an heterogeneous one would reduce power consumption by respectively 30% and

20% compared to an Intel I7. In this case we consider that load is 100% the remaining time, i.e. each time the workstations are not idle. This assumption is the worst case for the proposed approach.

In [3], authors monitored usages of a Google data-center. This article provides the distribution function of load of the servers. Using these different types of hardware on the data of would give a gain on power consumption of 60% for perfect proportional hardware and of 5% for the heterogeneous method. This result is obtained by using the measured load distribution in [3] and by selecting the best combination of nodes for each measured load.

5 Conclusion & perspective

Having a perfectly proportional hardware would allow not only to split in half the power consumption of data-centers, but also would reduce the need of increasing computing density. Having such hardware would allow to distribute more widely data-centers and thus to reduce cooling energy-cost and dramatically reduce overall energy consumption.

This article shows that even using simple different hardware it is possible to build more efficient data-center using 20% less energy. Using the same method on a wider variety of hardware would lead to reduce further this energy consumption. These improvements are also possible for Google-like data-centers, and would improve efficiency by 5% whereas lots of efforts are already done to manage their load.

Using proportional hardware for a workstations would provide even more improvement: from 60 to 80% for a perfectly proportional hardware, and from 20 to 60% for a workstation composed of an Intel I7, an Intel Atom and two Raspberry Pi.

In this article only the architecture type is taken into account. A next step will be to also take into account the possibility of changing processor frequency or to activate other energy-efficient leverages such as the possibility to migrate tasks or virtual machines between architectures (x86 and ARM). The current proposed hardware combination is only suitable for service-oriented workload because of the two open issues: migration between different architectures, latency of migration.

Another step will be to increase the range of possible hardware composing the heterogeneous proportional hardware.

Acknowledgement

The results presented in this paper were obtained partly using the platform of the CoolEmAll project. This project is funded by the European Commission under contract 288701 through the project CoolEmAll. This platform is composed of a 1U rack of 6 Intel I7 and 12 Intel Atom from Christmann[2]. Thanks

[2] http://www.christmann.info/

to Philippe Truillet for the Raspberry Pi and Jean-Marc Pierson for his help.

References

[1] Anurag Acharya, Guy Edjlali, and Joel H. Saltz. The utility of exploiting idle workstations for parallel computation. In *SIGMETRICS*, pages 225–236, 1997.

[2] L.A. Barroso. The price of performance. *Queue*, 3(7):48–53, 2005.

[3] L.A. Barroso and U. Holzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007.

[4] P. Domingues, P. Marques, and L. Silva. Resource usage of windows computer laboratories. In *International Conference Workshops on Parallel Processing, 2005. ICPP 2005 Workshops*, pages 469 – 476, june 2005.

[5] H. Esmaeilzadeh, E. Blem, R.S. Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, pages 365–376. IEEE, 2011.

[6] X. Fan, W.D. Weber, and L.A. Barroso. Power provisioning for a warehouse-sized computer. *ACM SIGARCH Computer Architecture News*, 35(2):13–23, 2007.

[7] P. Greenhalgh. Big. little processing with arm cortex.-a15 & cortex-a7. Technical report, ARM Whitepaper, 2011.

[8] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki. Toward dark silicon in servers. *Micro, IEEE*, 31(4):6–15, 2011.

[9] Arlitt M. and Jin T. World cup web site access logs, August 1998. Available at http://www.acm.org/sigcomm/ITA/.

[10] Matt W. Mutka and Miron Livny. The available capacity of a privately owned workstation environment. *Perform. Eval.*, 12(4):269–284, August 1991.

[11] James Wang. Nvidias next generation notebook technology: Optimus. Technical report, Nvidia Whitepaper, 2010.

[12] G. Zipf. *Human Behaviour and the Principle of Least-Effort.* Addison-Wesley, Cambridge, MA, 1949.