Developing Energy Efficient Filtering Systems

Leif Azzopardi, Wim Vanderbauwhede, Mahmoud Moadeli
Dept. of Comp. Sci., University of Glasgow
Glasgow, United Kingdom
{leif, wim, mahmoudm}@dcs.gla.ac.uk

ABSTRACT

Processing large volumes of information generally requires massive amounts of computational power, which consumes a significant amount of energy. An emerging challenge is the development of "environmentally friendly" systems that are not only efficient in terms of time, but also energy efficient. In this poster, we outline our initial efforts at developing greener filtering systems by employing Field Programmable Gate Arrays (FPGA) to perform the core information processing task. FPGAs enable code to be executed in parallel at a chip level, while consuming only a fraction of the power of a standard (von Neuman style) processor. On a number of test collections, we demonstrate that the FPGA filtering system performs 10-20 times faster than the Itanium based implementation, resulting in considerable energy savings.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software: Performance evaluation

General Terms: Performance

Keywords: Efficiency, Filtering, FPGA

1. TOWARDS GREENER SYSTEMS

Servicing millions of user search requests and processing volumes of information requires massive amounts of computational power that consumes energy and produces heat [1]. Energy costs for computing and cooling represent a significant expense to the operation of data centers [1]. This motivates the development of energy efficient solutions for processing large amounts of data and information. Reducing the amount of energy consumed provides a win-win situation: service providers can significantly reduce their costs by consuming less energy, and the impact upon the environment is greatly reduced.

In this poster paper, we report on our initial steps towards developing environmentally friendly Information Retrieval systems. We focus on the task of information filtering, where given a set of information needs (profiles), incoming documents are matched against these profiles [2]. When faced with large volumes of incoming documents, processing needs to be performed in real time, and therefore time based efficiency is paramount. Our aim is to filter documents efficiently by processing the requests on low power FPGAs as opposed to power hungry microprocessors.

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.
ACM 978-1-60558-483-6/09/07.

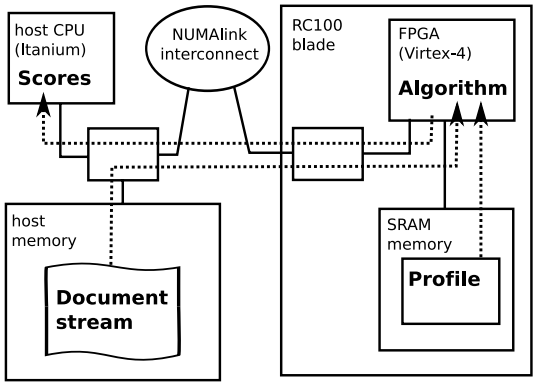

Figure 1: Schematic of FPGA-accelerated filtering

2. SYSTEM ARCHITECTURE

An FPGA is a reconfigurable semiconductor device which can be programmed to encode any logical function that an application specific integrated circuit could perform. FPGAs offer the parallelism inherent in logic integrated circuits combined with programmability and have therefore become increasingly popular for many applications. In particular there has been a lot interest in FPGA-accelerated scientific computing [6]. Until recently, a major drawback was the lack of high-level programming solutions for FPGAs. The traditional Hardware Design Languages such as Verilog and VHDL require specialized hardware developers and in general the design process is very time-consuming. For this work we have used Mitrion-C from Mitrionics, a high-level language which greatly reduces the complexity of FPGA programming and lends itself well to rapid application development.

The FPGA filtering application (Fig. 1) matches a stream of documents against a profile (a set of terms with weightings estimated using the relevance model [3]) which it receives via a network interface and returns a stream of scores. The system was implemented on an SGI Altix 4700 machine which hosts two RC100 blades. Each blade contains two Xilinx Virtex-4 LX200 FPGAs running at 100MHz. The profile is stored in the SRAM memory on the RC100 blade; the documents are streamed from the host memory over a high-speed I/O interface (NUMAlink).

For the CPU reference application, the algorithm was implemented wholly in C++ and executed on a standard CPU (a dual-core Itanium 64-bit processor running at 1.6GHz). The Itanium processor consumes approximately 130 watts

Collection	# Docs	Avg. Doc. Len.	Avg. Uniq. Terms
Aquaint	1,033,461	437	169
USPTO	1,406,200	1718	353
EPO	989,507	3863	705

Table 1: Collection Statistics

Collection	Profile # Docs	Profile # Uniq. Terms	Processor CPU (secs)	Processor FPGA (secs)	Gain
Aquaint	1	254	21.3	2.6	8.3x
	10	1,444	27.4	2.6	10.5x
	50	4,713	34.5	2.6	13.2x
USPTO	1	28	64.0	7.2	8.9x
	10	148	68.3	7.1	9.6x
	50	615	76.9	7.5	10.3x
EPO	1	1,327	107.3	8.4	12.7x
	10	4935	153.3	8.1	19.0x
	50	12,314	177.1	8.5	20.8x

Table 2: Performance Statistics

of power [4] whereas the Virtex-4 FPGAs consumes approximately 1.25 watts of power [5], which clearly illustrates the potential for power saving offered by FPGA technology [1].

3. EXPERIMENTS AND RESULTS

To compare and contrast the performance of the FPGA based filtering implementation against a standard implementation, we used three test collections where the average length and average number of unique terms in a documents varies. Table 1 shows the collections used: TREC Aquaint, and two collection of patents from the US Patent Office (USPTO) and the European Patent Office (EPO), respectively.

To simulate a number of different filters, for each collection, profiles were constructed by selecting a random document, using the title as the query, then selecting the top n documents as pseudo-relevant documents. These n documents given the query, were then used to construct a relevance model [3]. The relevance model defined the profiles which each document in the collection was matched against (as if it were being streamed from the network). n was varied from 1 to 50, to determine the impact on performance as the size of the profiles increased (both number of terms, and number of documents). This was repeated 30 times. The results are summarized in Table 2 and Figure 2. From the table, it is clear that the FPGA implementation is typically an order of magnitude faster than the standard implementation. From the figure, it can be seen that as the profile increases (i.e. the number of terms that require matching increases) the standard implementation becomes slower and slower, while the FPGA implementation remains relatively constant. This is because of the wide and pipelined parallelism that can be utilized within the FPGA.

[1]It should be noted that due to the system architecture, the FPGA implementation requires a Itanium processor to control the processing, resulting in limited power savings. However, a purpose-built FPGA based system could be substantially more power efficient.

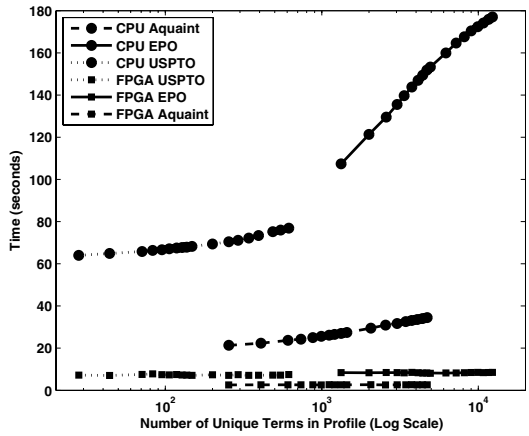

Figure 2: Time in Secs vs. No. of Docs in Profile.

4. CONCLUSIONS AND FUTURE WORK

In this poster, we have demonstrated that FPGAs provide a substantial reduction in the time taken to filter a stream of incoming documents (by up to a factor of 20), whilst consuming only a fraction of the power of a conventional processor. It should be noted that even if the FPGA system would consume the same amount of power as the conventional implementation, the energy consumption would be reduced by a up to factor of 20 because of the speed-up. These results demonstrate that the usage of greener hardware could deliver tremendous benefits by reducing the power consumed, and also increasing the speed of execution. Future work will be directed towards: (i) improving the performance of the current prototype, (ii) scaling the prototype up to data centre scale, and (iii) implementing more sophisticated filtering algorithms, along with other IR tasks such as ad-hoc retrieval and classification/clustering.

Acknowledgments This project was funded by Matrixware, and supported by Mitrionics and the Information Retrieval Facility. We would like to thank Fredrik Larsson from Mitrionics for his help and assistance with Mitrion-C.

5. REFERENCES

[1] C. L. Belady. In the data center, power and cooling costs more than the it equipment it supports. *Electronics Cooling*, 13(1), 2007.

[2] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: two sides of the same coin? *Commun. ACM*, 35(12):29–38, 1992.

[3] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th ACM SIGIR*, pages 120–127, 2001.

[4] C. McNairy and R. Bhatia. Montecito: A dual-core, dual-thread itanium processor. *IEEE MICRO*, 25(2):10–20, 2005.

[5] S. Sharp. Virtex-4 Dynamic Power Comparison - A Case Study, dec 2005.

[6] O. Storaasli, D. Strenski, and C. Inc. Exploring Accelerating Science Applications with FPGAs. *Proc. of the Reconfigurable Systems Summer Institute, July*, 2007.