DesignJet Grid Computing—Optimizing Corporate Resources for Digital Publishing

Jose L. Abad[1], Angel Perles[2], Juan J. Serrano[2], Josep Giralt[1], Sergi Jarne[1]
1. Hewlett-Packard, Inkjet Commercial Division, Barcelona
2. Universidad Politécnica de Valencia, DISCA/GSTF, Valencia

Abstract

The usage of cluster printing capabilities is not as widely spread today in corporate environments as it could be. Printing remains for most places a single-device task, even when multiprocessor systems and inter-network protocols are evolving to integrate in other fields, e.g., ad-hoc networking, self-configuration protocols, and peer-to-peer transfers. Printers do not yet benefit from the potential advantages of having a high-level of inter-device collaboration.

In essence, the main reason is that a higher level of attention and control from the user is usually required; making distributed printing processes a complex task. But, what if our printing devices would become more autonomous? They could take by themselves some of the underlying decisions that are required for devices to work together liberating the user from this decision-taking process. How to effectively balance all of these requirements to maximize cost-effectiveness of our daily work?

Hewlett-Packard is investigating these possibilities with large format printers by using Grid computing technology. Large format printers are powerful devices with high-bandwidth for data transmission and processing, and that many times lay idle. This technology makes large format printers become a proactive powerful component in printing workflows, e.g., when adding a new printer, the overall bandwidth increases in a balanced manner; the grid is as easy to use as having just one printer – see Figure 1 to illustrate it.

However, there are limitations in the suitability of Grid technology for printing; communication overhead can be an issue as for large-format-printers generally deal with large volumes of printable data. This paper does an analysis on the conditions in which printing in a grid makes sense.

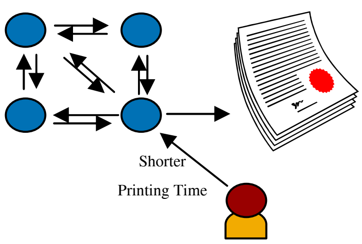

Figure 1- Optimal Scenario.

Introduction

Distributed collaborative printing environments are almost non existent today in corporations. The problem is that requires (among other things) a high level of attention and control from users. Printing remains a non-distributed task today other than for maintenance or printing over the network.

This could happen if our devices would be more autonomous, so that they could take by themselves some of the underlying decisions that are required for collaborative printing. For example, deciding to use an external centralized RIP for color processing-intense operations in base to the nature of the job.

We propose achieving autonomy by using Grid computing techniques. In general a computing grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed "autonomous" resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements [1].

In our specific case, resources are printers and printing options RIPs, languages supported, finishing options, etc. There are obviously limitations to the remote availability of some operations, for example, distance and the amount of data to transfer can be an issue.

The key distinction between clusters and grids mainly lies in the way resources are managed. In case of clusters, the resource allocation is performed by a centralized resource manager and all nodes cooperatively work together as a single unified resource. In case of grids, each node has its own resource manager and does not aim for providing a single system view. The autonomous aspect of grid is more interesting than for clustering.

Scenarios of Example

A user would perceive the grid as a "mega printer" having available all capabilities existing from printers in the grid, but not knowing the exact location of all of them. Previous projects and products have offered these capabilities before; the difference now is based on the flexibility, dynamism, and autonomy on which the grid can configure itself. In addition, the large computing power embedded within some of our printing devices can be used to collaborate in other grid-enabled activities taking place in an adaptive enterprise, some of them not related to printing.

Scenario 1. Marketing employee in a corporation has a poster. She wants to print it at her facilities where there are many large format printing machines around, but she has no printer herself, and she does not care on which printer to run the job as soon as this is done quickly.

Scenario 2. An employee of a SMB print service provider needs 10 posters on time. This is no urgent but she does not want to wait or monitor the process. What is more, she needs her computer free for other things. She wants to print where it will be cheaper and does not know whether there are RIPs or not installed some place. She will print to the grid knowing that it will be done using all available printers in the most cost-effective and efficient way. She will be receiving the posters via internal mail, and that is fine.

Scenario 3. A graphic artist will print to an external RIP because she likes changing the color curves in there. She stores post-ripped jobs in that machine for later reproduction, but she does not care where to finally print them as long as the profiles are correct and calibrated machines do a correct reproduction.

This paper has analyzed the feasibility of these scenarios from an analytic perspective using queuing theory techniques to model large format printers, devices and workflows. A natural continuation of this work would be prototyping the experiences to run in-lab customer satisfaction measurements.

The Model

We propose the use of a discrete-event based model of the system and prototype it in a C/C++ simulation environment using CSIM [2]. CSIM is adequate to express complex management algorithms and can be used in all evolving phases of the project to test the functionality that eventually will run in the devices.

Figure 2 shows a high-level view of the system containing printers, user workstations and external raster image processors (RIPs) that are attached to a standard local area network (LAN).

The LAN represents any internet or shared network-based connection of general purpose, not a dedicated network for printing and graphics-related operations. However, for simplification purposes in our model we consider the LAN having at least 100 Mbps of bandwidth. We also consider grid management cost in CPU as negligible compared with the one involved in printing and ripping. For the same reason the network cost of the grid management network can be considered as negligible when compared with the cost of transferring print jobs.

The user workstation is the source of jobs to the system. When a job is sent to a printer in order to get a print-out, the job goes through different processing stages as reflected in Figure 3. This is also a simplification in which we take job parsing and rendering as the main cost-computational consuming operations in the printing workflow (not in the communications) that are internal to the machine.

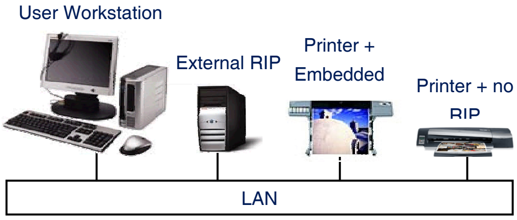

Figure 2- The Model.

Figure 3- Model of Job Processing.

Jobs are represented in our simulation model by using a set of conveniently selected attributes such as file size in Kbytes, file size after parsing in Kbytes, rendering size, job complexity indicator, etc. It is evident that sizes after parsing and rendering a job are not correlated to the size of the original file, but simply these sizes give us a representation of the complexity of the job if nothing else than for simplifying our model.

A Raster Image Processor (RIP) transforms original printable jobs into a raster language that is suitable to the printing engine and defined on the final device CMYK color space. The engine will then use half-toning algorithms bridging at last digital formats with paper output. Consider that embedded RIPs in the printer or software RIPs in user workstations are interesting capabilities to service the grid.

In our model the RIP is represented by a server using a FIFO queue discipline. Other scheduling methods, e.g., with priorities, or multi-queue servers have not been taken for simplicity of the model. The server has a performance processing base of R_{pref} Kbytes/sec. To compute the processing cost of a given ripped job we consider the $R_{complexity}$ and R_{size} attributes with the following expression:

$$Trip = R_{size} \ x \ R_{complexity} / R_{pref} \qquad (1)$$

The printer receives now jobs from the network, but these jobs are not necessarily received for producing a printout. They can also be received for ripping, color managing, etc before resending them out. They can also be temporarily stored until "the grid decides so" for example choosing where to produce the final printout.

Figure 4 shows the model using queues, all queues using FIFO schedulers for simplicity. The model has been simulated using the properties of a real load batch of complex – and large – graphic files such as maps, plots, poster-sized pictures, etc. The load was constructed to be representative of that present in a large format printing workflow.

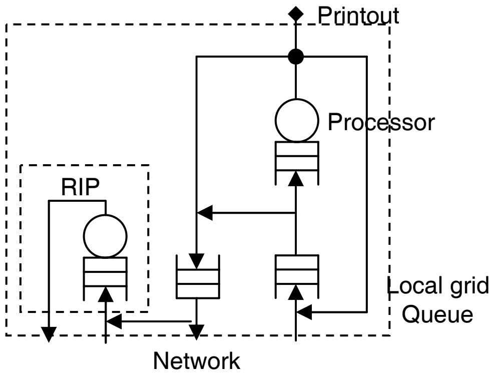

Figure 4- Queueing Model of a Printer.

Analysis

The results of the simulation of the previously described model lead to an interesting set of conclusions. First of all, let's explain the characteristics of the simulation. Simulations were executed for two types of networks, a 10 Mbps and a 100 Mbps shared Ethernet network. For each load with the benchmark files we varied the number of copies requested per each file, and the number of printers in the grid (changing from one to five). We were assuming network delays remained constant independently of the distance between nodes in a grid.

Figure 5 condensates the results of the simulations. We see how little is network bandwidth affecting the overall waiting times, as for jobs spend most of their lifetime in ripping and rendering phases. On these results it becomes also clear the polynomial reduction of waiting time with the increase in the number of printers in the grid (orange line). The distances A, B, C are the extra time to wait when a large (50) number of copies is requested. Not surprisingly, when 5 printers are available in the grid, the total copying time (C) is almost a fifth of what it takes having just one printer to do the job (A).

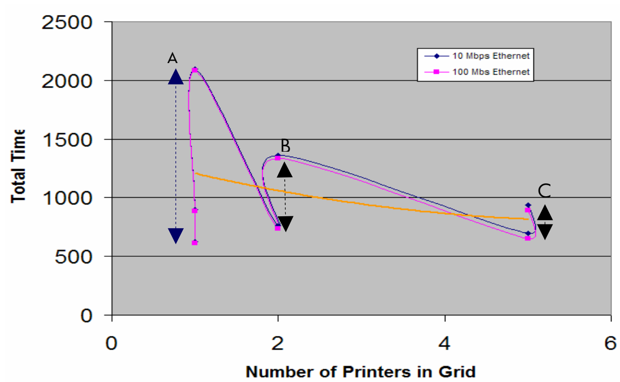

Figure 5- Results of the Simulation.

Conclusion

Although computing power is each time increased with newer and smarter devices, their level of autonomy is still very limited. We propose using grid technology for that matter and then analyze the infrastructure requirements as well as the feasibility limits on large format printing workflows.

This analysis has proven the feasibility of the idea from its very bases, showing how much transfer times become irrelevant. However, we can already anticipate a number of problems that will rise on products implementing these features, for example managing access control issues in these printing grids.

Future steps will guide us into solving these problems so that customers, corporations and small-medium-business could use printing grid technology to easily and effectively increase their computing power when purchasing a new printer.

References

1. K. Krauter, R. Buyyam and N. Nageswarab, "A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing", Software – Practice and Experience 2001; 00:1-7, John Wiley & Sons.
2. H. Schwetman. CSIM: A C-based, process-oriented simulation language. In Proceedings of the 1986 Winter Simulation Conference, pp 387-396, Washington, DC, December 1986.

Biography

Jose L. Abad is a senior software architect working for Hewlett-Packard in Barcelona. Previous working for HP Jose has held several management and research positions in the Silicomp Research Institute (previously known as Open Software Foundation) in Grenoble (France), IBM Zurich Research Laboratory in Zurich (Switzerland) and European Space Agency in Darmstadt (Germany).

Angel Perles received M.S. and Ph.D. degrees in computer engineering from the Polytechnic University of Valencia, in 1994 and 2003 respectively. He is assistant professor of the Department of Computer Engineering at the Polytechnic University of Valencia since 1997.

Juan Jose Serrano is professor and director of the fault tolerant computer group of the Computer Engineering Department in the Universidad Politécnica de Valencia. He received a BS in electrical engineering in 1979 and PhD in 1987. Professor Serrano has co-authored over 80 publications in conference proceedings and book chapters.

Josep Giralt graduated in Electrical Engineering in 1986 at the Universitat Politecnica de Catalunya. After working three years as a researcher in computer graphics at the Departament de Mètodes Informàtics of the UPC, he joined the HP R&D center in Spain in 1989. He is now a main senior software architect for large format printing devices and workflows.

Sergi Jarne graduated in Computer Science in 1989 at the Universitat Politecnica de Catalunya. His career started as development engineer in Microsoft Corp. and he then moved to the HP R&D center in Spain in 1993. He is now managing the development of software for large format printing devices and workflows.