Changing Demand - Evaluating Effects Using Simulation as a Service

Sunil Kothari, Thomas Peck, Jun Zeng, Francisco Oblea, Gary Dispoto; Hewlett-Packard Laboratories, Palo Alto, CA – 93404, USA

Abstract

Many industrial and commercial print service providers (PSPs) are dealing with fragmentation of high copy count, low-mix demands to low copy count, high-mix demands due to mass customization and personalization of content. To understand the true dynamics of a changing demand, a stochastic discrete event simulator such as Ptolemy, from UC Berkeley, could be a very useful tool. However, simulation usage presents a high technology and knowledge barrier to solution architects. Additionally, IT requirements such as periodic upgrade and update of software and hardware infrastructure further imperils the use of simulation for solution architects. With the advent of cloud technology and service broker architecture, both problems can be solved effectively. Solution architects can now quantify effects of these changes using simulation-as-a-service (SimaaS) without worrying about simulation and the hardware and software updates.

Introduction

Many industrial and commercial print service providers (PSPs) are dealing with fragmentation of high copy count, low-mix demands to low copy count, high-mix demands due to mass customization and personalization of content. Both the pre-press and the post-press processes are affected by a change in demand differently. One of the effects of these fragmented demands is on cycle time and throughput. To understand the true dynamics of a changing demand, a stochastic discrete event simulator such as Ptolemy [1], from UC Berkeley, could be a very useful tool. However, simulation usage presents a high technology and knowledge barrier to solution architects. One-off implementation can be done for a specific customer, but, for digital printing press manufacturer such as Hewlett-Packard, such an approach is clearly not scalable, given the range of applications, equipment configurations, and the combinatorial explosion of options. Additionally, IT requirements such as periodic upgrade and update of software and hardware infrastructure further imperils the use of simulation for solution architects. With the advent of cloud technology and service broker architecture, both problems can be solved effectively. Solution architects can now quantify effects of these changes using simulation-as-a-service (SimaaS) without worrying about simulation and the hardware and software updates [2].

SimaaS and the Service Broker Architecture

The SimaaS service uses a novel service broker architecture as shown in Figure 1 to orchestrate flow of jobs across a factory, taking into account the heterogeneity of demands, resources, operating policies and does not require knowledge of simulation for its end users. Solution architect can provide domain specific inputs and the back end code handles all the plumbing needed to run the simulation in the cloud and displaying the results back to the user.

SimaaS is currently a HP Labs prototype that helps in evaluating change of demands, a change of substrates, and changes in equipment. It quantifies different scenarios by simulating production and analyzing the production and financial metrics. For one of our customer studies, SimaaS was used to quantify the effects of change in demand from a high copy count, low-mix to a low copy count, high-mix demand. The SimaaS service uses a novel service broker architecture as shown in Figure 1 to orchestrate flow of jobs across a factory, taking into account the heterogeneity of demands, resources, operating policies and does not require knowledge of simulation for its end users.

The service broker architecture consists of a service broker actor which has three sub-modules: 1) A module that prioritizes order stream and releases a fixed amount of orders for production after a given period; 2) A task router module that breaks individual orders into tasks (based on a product's production plan) and then routes new tasks, rework tasks and old tasks as per the production plans; 3) A payload analyzer that determines the amount of work needed to accomplish a task. A group of machines, attributed by the process that they accomplish, are collected together and termed as *servergroups*. The servers or machines in a servergroup can be from several manufacturers and may have different capabilities. The flow of tasks to servergroups is coordinated by the means of a *dispatcher*. Apart from standard operating policies such as FIFO, earliest due times, minimum WIP, we allow for custom policies such a task with certain should always go to a selected group of machines. Several other details are elaborated about the service broker architecture and the algorithms used for task routing are elaborated in [7].

Solution architect can provide domain specific inputs and the back end code handles all the plumbing needed to run the simulation in the cloud and displaying the results back to the user.

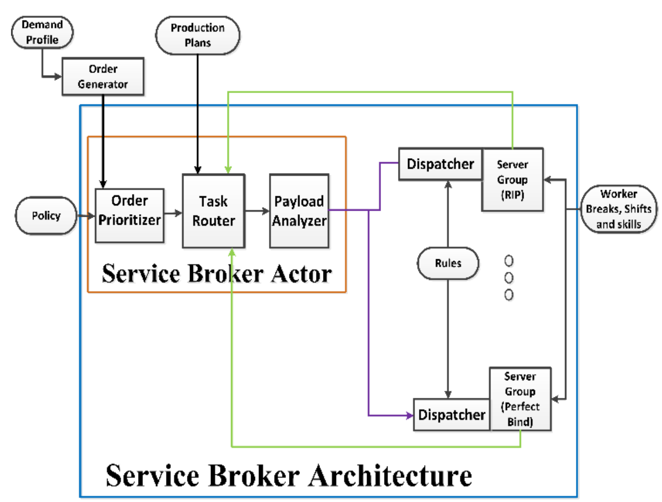

Figure 1: Service Broker architecture helps in dealing with heterogeneity of equipment configurations and demands

There is no limit to how many scenarios can be launched. Besides, SimaaS also provides information sharing such as standard equipment performance and cost characteristics yet proprietary workflows and operating policies are hidden from other customers. Our experiments suggest that such a service can be offered at a low subscription fee. A typical scenario involving an additional resource what-if takes 30 minutes from information gathering to schema generation to execution in the cloud.

Our prototype was deployed in a datacenter at HP Labs running an open-source cloud computing system called Eucalyptus [4]. As the business needs arose, we successfully migrated the cloud application to HPCS public cloud and then to an OpenStack [5] private cloud, again hosted at HP Labs. A technical report describing our experience with the migration is documented in [3]

Experiments and Results

To study the effect of a fragmented demand on an equipment set, we created two scenarios to reflect the current demand (high copy count, low-mix), the transitional demand (medium copy count, medium-mix) and the future demand (low copy count, high-mix). Without loss of generality, we assume that the print factory in consideration only produces a single product and the production plan consists of a directed graph of the following processes:
a) RIP; b) PrintBlock; c) PrintCover; d) Laminate; e) CutFoldBlock; f) PerfectBind; g) ThreeKnifeTrim; h) ThreeHolepunch; i) TrimCover; j) Shipping; k) Kitting; and l) Shrinkwrap

Table 1 shows the copy count and interarrival times for the three demands. Each of these demands will be investigated as a separate scenario.

Table 1. Scenarios and the corresponding demand profiles

Scenario No	Scenario Name	Attributes	Range
1	High copy count, low-mix	Copy Count	100-150
		Interarrival Times	10-15 hours
2	Medium copy count, medium-mix	Copy Count	55-75
		Interarrival times	5-6 hours
3	Low copy count, high-mix	Copy Count	1-15
		Interarrival times	1-2 hours

For all the scenarios, the page count and the price per copy of booklets remained the same. The price per copy was on a tiered structure based on the number of pages in a booklet as shown in Table 2.

Table 2. Determining price per copy for each booklet

Page Range	Price
100-150	$12.5
151-200	$13.5
201-250	$14.5
>250	$15

The operating policies for various task assignment and task prioritization remained the same in all the scenarios. All scenarios were run for a 4 production day period with the print factory running 24/7 with 4 shift breaks and 2 lunch breaks over a 24 hour period. Table 3 shows the results obtained from the three scenarios. We assumed that all the above jobs are produced on the same roll and the width of the roll on the press did not change throughout the production. A sophisticated model was used to model the sequence dependent setup times [7].

Table 3. Throughput and average cycle time from the two scenarios over 4 production days

Scenario	Number of Orders	Throughput	Avg. Cycle Time (in seconds)
1	13	3841	62031
2	28	4243	24398
3	114	4948	47368

Table 3 clearly shows that fragmentation of demand leads to increase in cycle time for roughly the same throughput.

Tables 4, 5 and 6 show the waiting time as a percentage of cycle time for the Scenarios 1, 2 and 3 respectively.

Table 4. Waiting time percentage and total cycle time for top 10 processes for scenario 1

Process	Total cycle time (in seconds)	Waiting time %
PrintBlock	110031919	100
PrintCover	13644083	99
ThreeKnifeTrim	20117	78
CutFoldBlock	29258	76
Laminate	17592	51
RIP	53709	49
Shipping	271779	47
Kitting	25774	38
PerfectBind	239457	21
TrimCover	10480	18

Table 5. Waiting time percentage and total cycle time for top 10 processes for scenario 2

Process	Total cycle time (in seconds)	Waiting time %
PrintCover	8732550	99
PrintBlock	29222895	99
PerfectBind	7895703	97
ThreeKnifeTrim	21199	75
CutFoldBlock	20073	57
RIP	133457	52
Laminate	16955	37
Shipping	192840	32

Kitting	57869	31
Shrinkwrap	200546	3

Table 6. Waiting time percentage and total cycle time for top 10 processes for scenario 3

Process	Total cycle time (in seconds)	Waiting time %
RIP	13558637	98
PrintBlock	722025	91
PrintCover	966945	90
PerfectBind	1339445	84
ThreeKnifeTrim	24213	80
Laminate	17071	42
CutFoldBlock	11788	23
ThreeHolepunch	32098	14
Shipping	200433	13
TrimCover	10960	8

Processes which are not done on a per copy basis such as RIP suddenly become important if we do not want additional bottlenecks for a given equipment set facing a fragmented demand.

Conclusions and Future Work

Many times print service providers are faced with the problem of being able to quantify multiple future scenarios, especially for demands which their print factory has not seen in the past. In this paper, we addressed the scenarios where the existing demand got fragmented. We highlighted some of the findings but we believe there are many more insights we can gain from such experiments. In the future, we plan to address related issues such as how to optimize production when sequence dependent setup times can no longer be ignored for very short run demands.

References

[1] J. Eker, J. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Sachs, Y. Xiong, "Taming heterogeneity - the Ptolemy approach", Proceedings of the IEEE, 91(1):127-144, January 2003.

[2] S. Kothari, J. Zeng, G. Dispoto, Print Production Designer: Answering Commercial/Industrial Print Production What-Ifs using Simulation-as-a-Service, 2013 Winter Simulation Conference.

[3] F. Oblea, A. E. Votaw, S. Kothari, J. Zeng, "Migrating SimCloud to HP Helion", HP Technical Report HPL-2014-36. Available at http://www.hpl.hp.com/techreports/2014/HPL-2014-36.html.

[4] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, D. Zagorodnov, "The Eucalyptus Open-Source Cloud-Computing System", CCGRID '09. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009, pp.124 – 131.

[5] OpenStack Website http://www.openstack.org

[6] T. Koponen, T. Virtanen, "A service discovery: a service broker approach", Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004.

[7] S. Kothari, J. Zeng, T. Peck, F. Oblea, G. Dispoto, "Simulation as a cloud service for short-run high throughput industrial print production using a service broker architecture", To appear in the Journal of Simulation Modelling Practice and Theory, 2015.

Author Biography

Sunil Kothari is a research scientist at the Hewlett-Packard laboratories in Palo Alto since Nov'2010. He graduated with a PhD in Computer Science from the University of Wyoming and a MS in Artificial Intelligence from the University of Edinburgh, UK and a BS in Industrial Engineering from the Indian Institute of Technology, Roorkee, India.

Thomas Peck received his BS from the Oregon State University in 1994. Since then he has worked for the Hewlett Packard's Inkjet High Productivity Solutions business unit. His work has focused on the development and integration of finishing equipment with HP's IHPS presses.

Francisco Oblea received his BS in computer systems engineering from ITESO in Guadalajara, Mexico in 2010. In 2010, he joined PiSA Laboratories as part of the development team. In 2011, he joined Hewlett Packard. He's been supporting Hewlett Packard laboratories in Palo Alto, CA with software engineering enabling cloud development for software-as-a-service efforts.

Jun Zeng is a senior scientist with Hewlett-Packard Laboratories. PhD (mechanical engineering) and MS (computer science) from Johns Hopkins University, and BS (modern mechanics) from USTC, China. Jun's publication includes 60 peer-reviewed papers and a co-edited book on CAD. He was a guest editor of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

Gary Dispoto is the director of the Print Production Automation Lab at the Hewlett-Packard laboratories. He holds several U.S. patents related to color imaging. Gary received B.S. and M.S. degrees in electrical engineering from Stanford University and an MBA degree from the University of Santa Clara.