Novice to Expert continuum may affect System Response Time

Arindam Das[1], Olivia Das[2]
York University (Alumni)[1], Toronto, Canada; Ryerson University[2], Toronto, Canada.

Abstract

To familiarize herself with a user-interface of a software system, a user needs practice. With practice, a user's think time gradually decreases—the novice to expert transition. We propose a queueing model that accounts for this transition in analyzing the performance of a distributed software system. We solve the model using deterministic simulation. Our model captures system performance in terms of system response time. We use the model to demonstrate how users—who are at various experience levels in the novice to expert continuum—may affect the system response time.

Keywords: User Think Time; System Response Time; Queueing Network, Novice to Expert Transition.

Introduction

Nowadays, distributed systems are getting deployed in cloud since cloud computing allows for dynamic scaling of computational resources as required on a pay-per-use basis. This relieves the providers of cloud system services from buying and maintaining data centers thereby reducing the operational cost.

The performance of a cloud system can get affected due to novice to expert transition as the end users of the system learn to use the system through its user interface. When a user learns, gradual *decrease* in her think time occurs. Once a software system has responded to a request that was submitted by an end user (through a user interface), the *user think time* (UTT) refers to the number of seconds the user takes to plan before submitting the next request to the system. The request submission is usually accomplished by clicking an icon on a computer screen. This think time is negatively correlated to the user's expertise level—lower expertise level leads to larger think time and higher expertise level leads to smaller think time. The system workload is affected as a user continues to learn using a system. A novice (one having lower expertise level) requires larger UTT and therefore submits less number of requests per unit time (to the system) compared to an expert (one having higher expertise level). Consequently, as UTT of a user gradually decreases with practice (i.e. repeatedly using the system), the number of requests submitted to the system gradually increases, thereby impacting the system response time (SRT).

Performance evaluation of distributed systems has always considered the UTT as a random variable with a constant mean that does not change with practice (Trivedi, 2005). It *never* considered novice to expert transition during evaluation.

In this work, we focus on cloud-based systems where the number of computer terminals is fixed at any given point of time. Examples abound—ATMs for a banking system, check-in terminals for an airline at an airport, navigation training simulators for aircraft pilots—that are deployed in cloud. For our modeling purpose, we assume a hypothetical scenario of a tutorial that is run from inside a classroom. The tutorial consists of learning the fixed locations of buttons on a graphical user interface (GUI) of a cloud system. The classroom has a fixed number of computer terminals, one per student, for the tutorial. No sooner a student finishes the requirements of the tutorial, she is replaced by a new one who is assumed to be at the lowest expertise level.

To realize the above tutorial scenario, we choose a closed queuing network as our system performance model. Choice of a closed queuing network helps us in two ways—one, it allows us to conform to the number of terminals being constant at any given point of time during the tutorial. Secondly, it enables us to account for the decreasing UTT of a user that could occur with repeated use of the interface.

The **key contribution** of our work is as follows: We demonstrate the effect of novice to expert transition on SRT of distributed systems. We accomplish this through a queueing model. The model *accounts* for novice to expert transition. Through our model, we show how users—who are at various experience levels in using a system—may affect the system response time.

Novice to Expert Continuum: In the Context of Human Computer Interaction

We briefly explain what a learning curve is in the traditional context of human computer interaction (HCI).

Any new skill takes time to learn. End users take a while to ramp up on a new user interface; software designers take a while to ramp up on a new project. Learning refers to the acquisition of skill in performing a task through repeatedly executing the same task over time. People get faster and make fewer errors with practice—i.e. with repeated task execution.

In the domain of user interface, the core focus is always a human-centered approach to design—be it the design of a smartphone interface or the interface of a desktop screen. By doing so, we concede that the target of our user interfaces is a population of users with a wide span of skills. We must be aware that these skills change over time as a result of learning. If there are multiple users, they may operate at different expertise levels at the same time due to differences in their experience (Ritter, Baxter, Kim, and Srinivasmurthy, 2013). Such variation in user expertise often calls for a planning of computational resources that would ensure usability satisfaction with respect to SRT for users across all expertise levels.

To take learning into account, what we need is a graph that plots UTTs at different practice sessions for a given task. A graph like this is popularly referred to as the *learning curve*—the novice to expert continuum. Figure 1 elucidates the *three level hypothesis* of learning postulated by Fitts (1964), Anderson (1982), and Kim and Ritter (2015). The hypothesis posits that a learning curve is roughly divided into three levels of user expertise. The first level is where a user is a *novice* trying to acquire the knowledge to execute a trial. The UTT is usually high at this level. The next level is the *intermediate* level. At this level, the user tries to consolidate the knowledge acquired in the novice stage. The final level is the *expert* level. At this third level, the user fine tunes the existing knowledge—users still get faster at the trial, although the improvements get diminishingly smaller (Ritter, Baxter, Kim, and Srinivasmurthy, 2013).

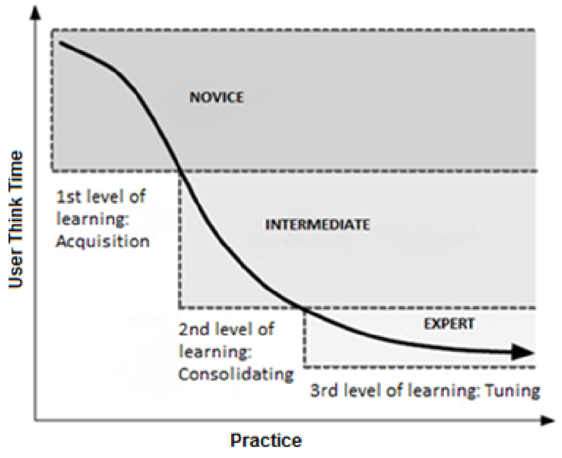

Figure 1. Three Level Hypothesis—Change in user think time across three levels of learning. The thick continuous line indicates continuous practice. (Figure adapted from Kim and Ritter (2015)).

In HCI, a learning curve for a user interface task is often obtained through empirical studies. Here, an interface under study is evaluated in a standalone mode of the client device such that the client software does not have to depend on anything other than the device it is hosted on. As a result, the delay between the submission of a user request (in form of a finger-press or mouse-click on the interface) and the corresponding response is assumed 0. The interface is evaluated through an interactive task. Multiple human subjects are sampled from a population of novice users of the task. The task involves completing a set of trials—let a trial be the submission of a user request to the software system (in the context of this paper). Each of the users is given equal number of practice sessions to perform the task. The time to complete every trial of the task is measured at each practice session. This measurement is taken for every subject over all the practice sessions. The mean time to complete a trial at a given practice session—mean trial completion time—is then obtained by averaging over the trial completion times measured across all the subjects at that session. Since the delay between every user request and its response is assumed 0, the mean trial completion time at a given practice session (which normally would have been the sum of mean user think time and mean system response time at the session) reduces to the mean user think time at that session.

Closed Queueing Network

A queueing network can be thought of as a network of servers with a queue of jobs (e.g. requests submitted due to mouse-click actions) at each server (Trivedi, 2001). We think of a server as being a world-wide-web server (web server), an application server (app server) or a database server (DB server).

In this work, we exploit a type of queueing network known as closed queueing network. An interactive terminal driven system is often modeled as a closed queueing network with a fixed set of terminals assumed to be part of the network. Each terminal models a delay center in the network. A terminal submits a request and waits for the response. It cannot submit another request until the response of the previous request returns. At any given point of time, the number of terminals in the network remains fixed.

A notion of *think time* exists in a closed queueing network. It is the time at each terminal between receiving the response of a request and sending out the next request. The idea of think time in a closed queueing network makes it a natural candidate for modeling an interactive system where a user ponders between the completion of a service request and the initiation of a new service request—the time to ponder thus being the UTT.

A hypothetical Scenario to predict the Learning Effect on System Performance

Our hypothetical scenario consists of the followings: a *tutorial* that enables a student to learn the fixed locations of buttons on a GUI; a *location learning task* that is repeatedly executed by a student in the tutorial; and a *software system* that is used to conduct the tutorial. The tutorial involves the learning of button locations on the GUI only. It does not involve learning of any other type of content.

Note that the task of learning stable layouts of GUI items is not uncommon in real life. While such tasks are routinely executed on banking ATMs and interactive kiosks on a daily basis, they are also prevalent in aviation training, command and control scenarios, and process control plants (Waldron et al., 2005).

Tutorial scenario

The focus of this paper is to demonstrate the effect of a learning curve on the performance of a 3-tier cloud system. To do so we imagine a tutorial scenario. We assume that a student attending the tutorial interacts with the system through a web-based user interface at a dedicated computer terminal.

The aim of the tutorial is to learn the location of buttons on a stable GUI of the system. The tutorial involves a given number of practice sessions. The practice sessions are assumed to be separated from one another by a constant period of inactivity. Each student is required to complete all the practice sessions.

The tutorial is conducted inside a classroom having a fixed number of computer terminals. We assume that one student uses one terminal only for her practice sessions. The tutorial begins with one student at every terminal.

Different students may complete their practice sessions at different points of time. It is assumed that when a student completes all her practice sessions, she is replaced by a new student joining the tutorial at the lowest expertise level—the first practice session.

Location learning task

We assume a simple location learning task that a student will repeatedly perform across multiple practice sessions in the aforementioned tutorial scenario. A location learning task is one where a student learns the locations of graphical items present on a user interface, given that the position (i.e. location) of the items on the interface do not change. We adopt such a task from Ehret's empirical study (Ehret, 2002). The interface on which the task is performed is a graphical layout that consists of 12 unlabelled square buttons arranged along the periphery of a circle. The locations of these square buttons are to be learned through practice. We refer to this interface as Unlabelled Interface. The twelve square buttons are mapped to twelve distinct colors. The colors are not visible; they stay hidden. The circle of square buttons surrounds a centrally located rectangular button. While every square button in the periphery acts as a potential target, the rectangular button at the centre of the circle acts as a cue color.

We refer to the task performed in learning the Unlabelled Interface as "Ehret's task". One practice session of Ehret's task consists of twelve trials. Each trial involves first *locating* and then *clicking* a square button that corresponds to a color displayed on the rectangular cue button. In a given practice session, the cue color is different for each of the twelve trials—every trial in a practice thus involves finding a target that is different from the rest eleven targets. In a trial, if the cue color is the color that is associated with the *clicked* square button, the user has found the target—the trial is therefore considered complete; otherwise the trial is to be repeated. For example, in a trial when the color in the rectangular cue button is green, the trial would be deemed complete only if the square button mapped to green is clicked by the user, not otherwise.

When Ehret conducted this task, he considered only the completed trials. We do the same while considering human learning in our model—we assume that every trial ends up finding the desired target. This helps us keep our model simple.

In Ehret's study, several human subjects had performed multiple practice sessions of Ehret's task on a standalone desktop computer with no internet connection. As a result the delay between the submission of a user request (in form of a mouse-click) and the corresponding response was assumed 0. We therefore consider a trial completion time in Ehret's study to be essentially a *user think time* (UTT) for the modeling purposes in this paper.

In our model, we utilize the mean trial completion time corresponding to each practice session of Ehret's task as the *mean user think time* for that session. We incorporate a 3-tier, cloud-based distributed system that is responsible for processing a submitted user request (mouse-click on a square button). We assume that this system processes the mouse-click and returns the response—the cue color for the next trial—after a non-zero delay (the delay being the system response time, SRT).

Software System

Figure 2 shows the software architecture of our hypothetical cloud-based distributed system that is used for conducting the aforementioned tutorial. We analyze this system in this work.

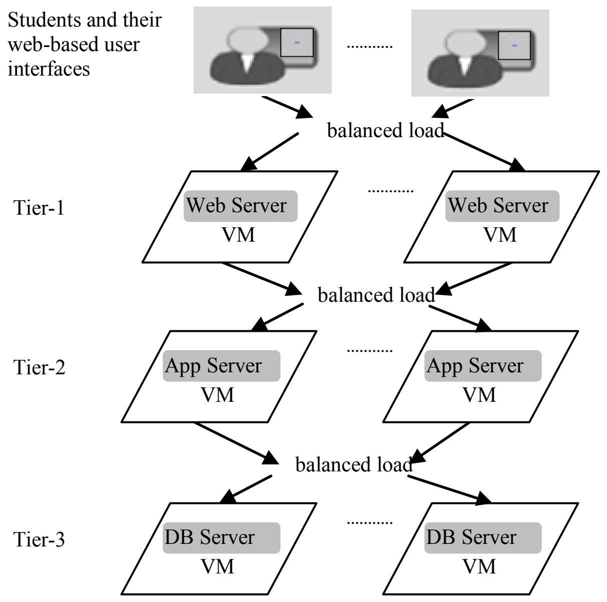

Figure 2. Our hypothetical 3-tier, cloud-based distributed system that is used to accomplish our tutorial scenario.

The system consists of 3 tiers. One or more Web servers run in the first tier (tier-1), one or more application servers (App servers) run in the second tier (tier-2) and one or more database servers (DB servers) run in the third tier (tier-3). The servers run of virtual machines (VMs). VM is a term used in cloud computing. It refers to a virtual (i.e. emulated) host. Similar to a physical host (computer), a VM can run an operating system as well as other processes. A VM can be acquired, networked, or released on demand through software based mechanisms. We assume that at any given tier, one or more VMs can be provisioned, each running a single instance of a server relevant to that tier.

We assume that the workload is equally distributed among the servers at any given tier. We indicate this in Figure 2 using the phrase "balanced load".

The student users access the application at the web servers through their web-based user interfaces.

Control flow of a trial

A trial refers to first locating and then clicking a target square button on Ehret's Unlabelled Interface. In a trial, first a user spends time in reasoning and planning where the target square button would be. Then she submits a *request* to the system by clicking the potential target. Here, a *request* refers to a job that is generated due to a button-click and that is to be subsequently processed by the software system (Figure 2) starting from tier-1 and until tier-3. We assume that a *request* will be processed exactly once (in a server) at each tier. After completion of processing at the third tier, a *response* corresponding to the processed request is returned to the user. We assume that a request incurs a waiting time in the server's queue before being processed, if the server is busy. The request then incurs a service time for getting processed in the server.

The request is first sent to a Web server in tier-1 for processing. If the Web server is busy then the request needs to wait in the server's queue before getting processed.

Once the processing of the request at tier-1 is finished, the request is redirected to an App server in tier-2. If the App server is busy then the request needs to wait in the server's queue before getting processed.

Once the processing of the request at tier-2 is finished, the request is redirected to a DB server in tier-3. If the DB server is busy then the request needs to wait in the server's queue before getting processed.

Once the processing of the *request* is finished at the DB server, the *response* corresponding to the request is sent back to the user. At this point, the trial is complete. We assume that the *response* returned to the user contains the information about a new cue color whose associated button is to be located (on the interface) in the next trial.

A trial thus incurs two delays. One is the time spent by a user in reasoning and planning where the target square button is located, given a cue color. This delay period is the *user think time* (UTT). The other is the system delay due to waiting times and service times incurred by the request between the click of a target button on the user interface and the return of the response. This second delay is the *system response time* (SRT).

Once all the trials of a practice session are complete, there is period of inactivity before the first trial of the next practice session begins—this period of inactivity is the *inter practice time*.

A Queueing Model Considering the Effect of Novice to Expert Continuum

A user with lower expertise level requires larger UTT and therefore submits less number of requests per unit time to the system. In contrast, a user with higher expertise level requires smaller UTT and therefore submits more number of requests to the system. Thus, as UTT of a user gradually decreases with practice, the number of requests submitted to the system gradually increases. The decreasing UTT thus influences the system workload which in turn affects the waiting times of the requests and consequently, the SRT. Keeping this in mind, we model our hypothetical distributed system as a closed queuing network.

The queuing model parameters, their meaning and their values are summarized in Table 1. The parameter values are specified inside bold parenthesis in the right column of the table. Each parameter is explained in due context as our work unfolds. Since the queueing network is a closed one, the total number of terminals (concurrent users) ***N*** in the system is constant at any point of time. An individual terminal user initiates a practice session p by first thinking for a certain amount of time with mean u_p (mean user think time per trial of practice session p) and then submitting the first request of that session to the system. After the completion of the request, the user thinks again for a time with mean u_p and then submits the subsequent request of the practice session p.

Once a user finishes ***T*** number of trials needed to complete a practice session, she takes a break for some time with mean α (mean inter practice time). The user then proceeds with the next practice session. The user completes ***P*** practice sessions in total before leaving the system. A departing user is replaced by a new novice user who begins her practice at practice session 1.

We assume that μ_1 is the mean service time of each Web server replica at tier-1, μ_2 is the the mean service time of each App server replica at tier-2, and μ_3 is the mean service time of each DB Server replica at tier-3.

Table 1. Model Parameters

Parameter	Meaning
P	Total number of practice sessions assumed to be completed by a user before leaving the system ($1 \le p \le P$). **(P = 15 sessions)**
N	Number of computer terminals (concurrent users). Once a user completes ***P*** practice sessions, she leaves the system and a new novice user occupies the terminal. The new user begins her practice at practice session 1. ***N*** thus stays fixed during a simulation run, thereby abiding by the "constant number of customers" requirement for a closed queueing network. ($1 \le i \le N$). **(Simulation data collected at N = 120 simulated concurrent users)**
N_p	Number of concurrent users at practice session p at the start of a simulation run ($1 \le p \le P$) where $N_1 + N_2 + ... + N_P = N$. This is applicable when human learning is considered.
T	Number of trials to be completed in a practice session. This is assumed to be equal for all practice sessions. **(T = 12 trials per session)**
$r_{i,p}$	Actual number of completed trials in practice session p at terminal i during the simulation.

α	Mean inter practice time **(1 sec)**
μ_1	Mean service time at tier-1 **(0.5 sec)**
μ_2	Mean service time at tier-2 **(0.5 sec)**
μ_3	Mean service time at tier-3 **(0.5 sec)**
u_p	Mean User Think Time per trial of practice session p **(u_1 = 12.5, u_2 = 10.6, u_3 = 8.9, u_4 = 6.8, u_5 = 6.5, u_6 = 6.1, u_7 = 5.1, u_8 = 4.2, u_9 = 4.3, u_{10} = 4.3, u_{11} = 3.1, u_{12} = 2.7, u_{13} = 2.9, u_{14} = 2.5, u_{15} = 2.2. The values are in seconds. They are obtained from Figure 3)**

The system response time of a request is the time between the arrival of the request at a tier-1 server to the completion of the request at a tier-3 server. This time includes the waiting times at the queues of the relevant servers at different tiers and the service times of those servers. This implies that the SRT of a request is affected by the rate of request submissions (i.e. the number of requests submitted to the system per unit time) in addition to the service times of the servers.

User Think Time when human learning is not considered: When human learning is not considered, the think times of a user across all practice sessions will be identically distributed random variables with *same* mean $u_1 = u_2 \ldots = u_P$.

User Think Time when human learning is considered: When human learning is taken into account, the think times of a user across all practice sessions will be identically distributed random variables with unequal means $u_1 \neq u_2 \ldots \neq u_P$. Here, we take unequal means instead of purely decreasing means because of the following reason: Although a learning curve obtained through empirical studies show an overall decreasing trend in user think time with practice, sometimes it may exhibit exceptions in form of increased user think times at some practice sessions possibly owing to user fatigue.

We accomplish the analysis of our queuing model through deterministic discrete-event simulation. For simplicity, we assume that the user think times, the service times of the servers, and the inter practice time are deterministically distributed.

Let $s_{i,j,p}$ denote the system response time for a trial j of practice session p by a user at terminal i. During every simulation run, we record the response times $s_{i,j,p}$. Let $r_{i,p}$ denote the number of *completed* trials of practice session p at terminal i.

The *Mean System Response Time* (*Mean SRT*) per trial $\bar{s_p}$ of practice session p, where p = 1,2, …, P can be estimated as:

$$\bar{s_p} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{r_{i,p}} s_{i,j,p}}{\sum_{i=1}^{N} r_{i,p}}$$

The numerator of the above equation denotes the total system response time of all the trials of practice session p completed from all the terminals. The denominator represents the number of those trials.

Model Results

We use the human learning curve observed by Ehret (2002) as an input to our model. This empirical curve of human learning was measured when human subjects executed Ehret's task—the task to learn the locations of square buttons on an Unlabelled Interface explained earlier. Figure 3 shows the learning curve. The curve is in terms of the mean *user think times* across the first 15 practice sessions completed by the sixteen human subjects. Subjects' point of regard was measured as they performed the task. The eye data was collected via an ASL 5000 eye-tracker.

Our simulated users are assumed to execute the aforementioned Ehret's task. The simulation emulates the hypothetical tutorial scenario explained earlier.

The simulated tutorial is assumed to start with a fixed number of computer terminals—one student using one terminal only. Once a user completes all the 15 practice sessions she leaves the system. A departed user is then replaced by a new novice user who begins her practice at practice session 1.

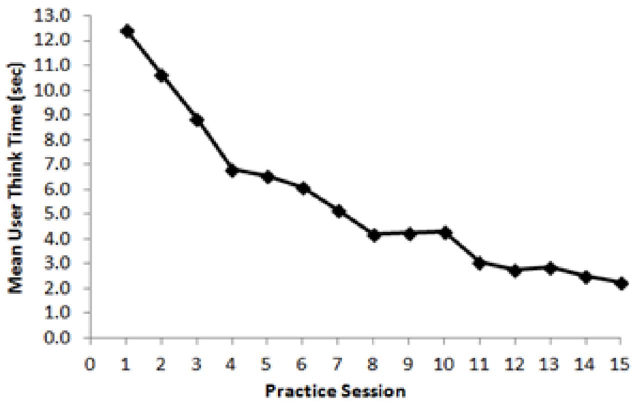

Figure 3. Human learning curve for Ehret's task (Ehret, 2002).

We show how users—who are at various experience levels in using a system—may affect the system response time. To do so, we run our simulation model with an initial proportion of users who are at various expertise levels.

We refer to the learning curve of Ehret's task (Figure 3). Let [N_1 / N_6 / N_{11}] denote the initial proportion of users where N_1 users begin their practice at session 1, N_6 users begin their practice at session 6, and N_{11} users begin their practice at session 11 at the start of a simulation run. We assume that there are no users at other expertise levels at the start of the simulation run, i.e. $N_1 + N_6 + N_{11} = N$. Here, we choose the practice sessions 1, 6 and 11 assuming that novice-level experience starts at session 1, intermediate-level experience starts at session 6 and expert-level experience starts at session 11. Our choice is dictated by the *three level hypothesis* of learning (discussed earlier; see Figure 1) applied on the learning curve of Figure 3.

We consider 120 concurrent users in the system, i.e. $N_1 + N_6 + N_{11}$ is 120. We perform one logical-hour analysis for a VM configuration where every tier of our three-tiered system has 6 VM replicas.

Table 2 shows mean SRTs $\overline{s_{novice}}$, $\overline{s_{intermediate}}$ and $\overline{s_{expert}}$ for two initial proportions of users [120/0/0] and [40/40/40]. Here $\overline{s_{novice}}$ refers to $\overline{s_1}$, the Mean SRT at practice session 1; $\overline{s_{intermediate}}$ refers to $\overline{s_7}$, the Mean SRT at practice session 7; and $\overline{s_{expert}}$ refers to $\overline{s_{15}}$, the Mean SRT at practice session 15.

With respect to analyzing the initial user proportion [120/0/0] for one logical-hour, the reason for low mean SRT per *novice* trial (2.25 sec) but high mean SRT per *expert* trial (7.22 sec) is as follows: In this case, the system is transiting from all-novices to all-experts. The user think times (UTTs) at the novice level are substantially higher than those at the expert level. Therefore when all the users are at novice level, the rate of request submissions (to the system) is lower compared to all-experts. This leads to less waiting times during novice request executions and higher waiting times for expert request executions.

Table 2. Mean SRTs $\overline{s_{novice}}$ (i.e. $\overline{s_1}$), $\overline{s_{intermediate}}$ (i.e. $\overline{s_7}$) and $\overline{s_{expert}}$ (i.e. $\overline{s_{15}}$) for different initial proportions of users. One logical-hour analysis. VM configuration consists of 6 VM replicas per tier. *N* = 120 users.

Initial User Proportion [N_1 / N_6 / N_{11}]	$\overline{s_{novice}}$ (sec)	$\overline{s_{intermediate}}$ (sec)	$\overline{s_{expert}}$ (sec)
[120/0/0]	2.25	5.12	7.22
[40/40/40]	4.36	4.5	5.03

On the contrary, the mean SRT of 5.03 sec per *expert* trial is less in case of [40/40/40] in comparison to 7.22 sec for the proportion [120/0/0]. This is found from a one logical-hour analysis. The reason is as follows: In case of [120/0/0], all the users start at the novice level. They then transition to the intermediate level almost at the similar time. Finally, all the users transition from the intermediate to the expert level—again, almost at the similar time. Once at the expert level, the rate of request submissions by a user is higher in comparison to her rate of submissions either at the intermediate or at the novice level. On top of that, since almost all the users have transitioned to the expert level, an expert trial has no choice but to compete for resources against majority of the trials which are also occurring at the expert level. In contrast, for the case [40/40/40], an expert trial competes for resources against a mixture of novice, intermediate and expert trials. Consequently, the mean SRT for an expert trial is higher in case of [120/0/0] than the proportion [40/40/40].

Let an example SRT requirement be as follows: "The mean SRT should be less than or equal to 5.5 sec". Table 2 suggests that the VM configuration (6 VM replicas per tier) will satisfy the threshold of 5.5 sec for only the proportion [40/40/40] in one logical-hour analysis since all of $\overline{s_{novice}}$, $\overline{s_{intermediate}}$ and $\overline{s_{expert}}$ are below the threshold for that proportion. The other proportion [120/0/0] will not satisfy the threshold since $\overline{s_{expert}}$ = 7.22 sec being more than 5.5 sec, the proportion will not be able to meet the SRT threshold for the expert trials.

This analysis of the effect of user proportion on SRT can be used by the system analysts when the workload trend for the system under analysis is known. An example workload trend could be a plot of initial user proportions of 120 concurrent users of the system against different times of the day obtained from the historical data of the system's usage. Suppose the plot shows an initial user proportion [120/0/0] at 8am and [40/40/40] at 3pm. Our aforementioned one logical-hour analysis predicts that from 8am to 9am, the configuration of 6 VM replicas per tier would not satisfy the threshold SRT of 5.5 sec. But, the same configuration would ensure usability satisfaction (with respect to SRT) across all expertise levels from 3pm to 4pm.

Conclusions

We propose a queueing model that accounts for novice to expert transition in analyzing the performance of a distributed software system. Our model captures system performance in terms of system response time. We use the model to demonstrate how users—who are at various experience levels in the novice to expert continuum—may affect the system response time.

Acknowledgment

We would like to thank NSERC Canada for their financial support through the Discovery Grant of Olivia Das.

References

Anderson J. R. (1982). "Acquisition of cognitive skill," *Psychological review*, vol. 89, no. 4, pp. 369–406.

Ehret, B. D. (2002). "Learning where to look: Location learning in graphical user interfaces," CHI, pp. 211–218.

Fitts, P. M. (1964). "Perceptual-motor skill learning," in *Categories of human learning*, New York, NY: Academic Press, pp. 243–285.

Kim, J. W., & Ritter, F. E. (2015). "Learning, Forgetting, and Relearning for Keystroke-and Mouse-Driven Tasks: Relearning is Important," *Human-Computer Interaction*, vol. 30, pp. 1–33.

Ritter, F. E., Baxter G., Kim, J. W., & Srinivasmurthy, S. (2013). "Learning and retention," in *John D. Lee and Alex Kirlik* (*eds.*). *The Oxford Handbook of Cognitive Engineering*, New York, NY: Oxford, pp. 125–142.

Trivedi, K. S. (2001). *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, 2nd ed. John Wiley and Sons.

Waldron, S.M., Duggan, G.B., Patrick, J., Banbury, S., & Howes, A. (2005). "Adaptive information fusion for situation awareness in the cockpit," in Proc. 49th Annual Meeting of the Human Factors and Ergonomics Society, pp. 49–53.