Branching Processes: Simulating Birth and Assassinations …. update this

Dennis Murray, Jared Law, Julien Bataillard, Cory Nichols

April 3rd, 2018

## Abstract

Monte Carlo simulation is an extremely useful method to analyze complex mathematical relationships. When used to investigate a stochastic process, Monte Carlo simulation allows a researcher to analyze thousands of independent random outcomes given underlying distributional assumptions. In this paper, we model computer job queueing as a Poisson process. We use a Poisson distribution to estimate the number of jobs and an exponential distribution to estimate the lifetime of each respective job. We investigate the number of jobs created given these assumptions by simulating outcomes using different values of the Poisson lambda and exponential rate parameters. We find that the ratio of the two distribution parameters is more important than two discrete values in determining the number of jobs created. Thus, simulation parameters can be simplified to use only the ratio of the two parameters instead of iterating over hundreds of combinations.

## Introduction

In the context of software engineering, it is often good practice to anticipate the likelihood of a release date, the cost of a project, or the impact of implementing new code base. These types of exercises usually fall into the category of risk analysis. However, it is often extremely difficult to successfully identify or anticipate outcomes if those outcomes are considered mostly random. One such software engineering example proudcing random outcomes is computer job queueing.

Even with parallel processing, a queue of interdependent tasks can form, often slowing down processing time. The number of these tasks can be completely random, specifically when considering the amount and type of requests the computer must consider. Typically, a parent job generates children which then generate more children. Additionally, each one of these child jobs have a particular lifetime or run time and can generate additional child jobs. Each job ultimately depends on the completion of prior tasks, causing a slow down in processing time.

In order to model the queueing process we must utilize probability distributions to estimate the number of jobs created and their associated lifetimes. Given the randomness of the queueing process, we explore many different outcomes utilizing these probability distributions via Monte Carlo simulation. These simulations allow us to explore the distributions of job counts our queue model produces.

Of particular importance are the associated distributional parameters that must be identified as part of our simulation analysis. Our objective is to determine if these parameters and their values are related to the longevity of the queueing process. Specifically, we investigate whether discrete parameter values or the ratio of the two parameters is more important to the longevity of the queueing process.

In the following section, we review academic literature on stochastic processes and Monte Carlo simulation. In the methods section, we provide a description of the probability distribution assumptions used to model the CPU queueing process. We also identify the parameter selections and logic used to test the sensitivity of our model given the distributional parameters. Results of Monte Carlo simulation for the queueing process and associated parameters are then presented and compared. We conclude this paper by summarizing results and considering future work.

## Background

Gabbiani and Cox [1] define a stochastic process as a collection or ensemble of random variables indexed by a variable t, usually representing time. They exist in two types - discrete-time and continuous-time stochastic processes. They are widely used as mathematical models to describe systems that occur in a random manner. The applications are vast in computer science, cryptography, telecommunication, and finance but also in biology, physics, chemistry, and neuroscience.

One of the easiest stochastic processes to understand is the Bernoulli process [2] as it only has two possible outcomes - yes/no, success/failure, 1/0, blue/red, head/tail. with probabilities of {p, 1-p}. For example, tossing a fair coin ten times can be modeled as a Bernoulli process with a sequence of ten coin-flip results and probabilities of 0.5 for head and 0.5 for tail (since it is a fair coin). Each coin-flip is a binary-valued random variable.

Another example of a stochastic process is the Poisson process which we use in this paper to estimate the number and length of the jobs in the queue. It is used in instances when we need to count the number of occurrences of an event that happens at random. In practice [3] the Poisson process is used to model the number of car accidents at a site, the location of users in a wireless network, the request for individual documents on a web server and the outbreak of wars.

The Poisson process with rate is defined as [3]: Let > 0 be fixed. The counting process {N(t), t $\elementof$ [0,???)} is called a Poisson process with rate if all the following conditions hold: 1. N(0) = 0; 2. N(t) has independent increments; 3. the number of arrivals in any interval of length ??>0 has Poisson(????) distribution.

In order to estimate the number of jobs and their lengths of time we simulate different outcomes by using Monte Carlo simulations. These are used to generate random variables for modeling risk or uncertainty of a system [4]. Monte Carlo simulations have some key advantages over more deterministic analyses [5] including - the results land probabilities so they show you what could happen but also how likely it is to happen; it is easy to generate graphical representations based on the results; and one can easily spot which output has the largest effect on the bottom line results.

[1] Fabrizio Gabbiani, Steven James Cox, in Mathematics for Neuroscientists (Second Edition), 2017 [2] <http://www.statisticshowto.com/bernoulli-distribution/> [3] <https://www.probabilitycourse.com/chapter11/11_1_2_basic_concepts_of_the_poisson_process.php> [4] <https://economictimes.indiatimes.com/definition/monte-carlo-simulation> [5] <http://www.palisade.com/risk/monte_carlo_simulation.asp>

## Methods

In order to anticipate job counts for our queueing example, we need to identify a method to model the previously described queue forming process. Given each job has a fixed lifetime and we are interested in the number of child jobs each parent job generates, we use a Poisson process to count the number of jobs. Additionally, we can view each job over a fixed interval of time, thus we can use the Poission distribution to estimate the number of jobs each parent generates. Additionally, each job runs for a certain amount of time. We can generate these run times using a random variable from the exponential distribution to determine the lifetime of each job.

We then identified several values for both lambda and kappa across both cases where kappa is kept constant at 1 and where kappa = c and does not equal 1. These values are indicated in the table below.

We selected several values for lambda and kappa with matching ratios to determing first in the test case to see the overall patterns that would emerge from the changes, and then the simulation case, iterating over 40 times to provide a representative sample. Our first attempt was to change the lambda while keeping kappa constant at 1 as shown below.

The first sample simulation over 40 iterations for the requisite sample size was then completed.

From this, we then changed the values for kappa and lambda to reflect the ratios used in the first example while ensuring that kappa did not equal 1.

And then the full run of 40 iterations of the simulation for our sample size.

## Results

## Conclusions and Future Work

The use of simulation, and especially Monte Carlo simulation, has become an increasing part of the public lexicon in recent years. Many models that are visible to the public include at some level a Monte Carlo simulation. These include election results modeling, as well as the outcomes of major sports leagues. Fivethirtyeight.com’s models for both the election, as well as for results of sports leagues, include Monte Carlo simulations. [1] The method has also been mentioned in the context of quantative finance, genetics and many other fields.

The pitfalls of Monte Carlo simulation may be somewhat obvious. It relies on an assumption that the future follows some probabalistic distribution, and that the future distribution is consistent with the past values of the variable. Like most statistically-based forecast methods, there’s some expectation that major shifts in the underlying causes of change in the behavior of the variable will remain the same. In the example of a model for a Financial Market, while the major shifts and catastrophic market events are possible in the simulation, the actual probability of the event might differ from the *n*-standard deviations from the mean that would be expected.

In this case, we simulated a process of birth and death of jobs, with dependencies between the start and finish, to understand the total time required. In the context of the case, these events were referenced as jobs or processes that are broken up for parallel computing, and the process by which the later jobs must wait for the completion of earlier jobs. This same method could be employed for the analysis of a manufacturing line. In manufacturing, the Monte Carlo process can be used to simulate a process with interdependencies, and variability.

The use of the simulation can be used to identified bottlenecks in the process.[3]

## References

[1] <https://fivethirtyeight.com/features/how-our-club-soccer-projections-work/> [2] <https://www.sciencedirect.com/science/article/pii/S0168160509006680> [3] <https://www.qualitymag.com/articles/91769-monte-carlo-and-manufacturing>

## APPENDIX - BOOK CODE