Analysis of Hypoexponential Computing Services for Big Data Processing

Conference Paper · August 2015
DOI: 10.1109/FiCloud.2015.57

CITATIONS 0
READS 105

3 authors:

Natalia G. Miloslavskaya
National Research Nuclear University MEPHI
116 PUBLICATIONS 35 CITATIONS

Sergey V. Zapechnikov
National Research Nuclear University MEPHI
22 PUBLICATIONS 4 CITATIONS

Alexander Tolstoy
National Research Nuclear University MEPHI
91 PUBLICATIONS 31 CITATIONS

All in-text references are linked to publications on ResearchGate, letting you access and read them immediately.

Available from: Natalia G. Miloslavskaya
Retrieved on: 28 September 2016
Analysis of Hypoexponential Computing Services for Big Data Processing

Sergey Zapechnikov, Natalia Miloslavskaya and Alexander Tolstoy
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
Moscow, Russia
{SVZapechnikov, NGMiloslavskaya, ATolstoj}@mephi.ru

Abstract—The paper presents two analytical models to study the performance and availability of queueing systems with the hypoexponential service time and finite queue. The analytical results obtained are verified using discrete-event simulation. A few numerical examples for varying number of service stages, service rates and arrival rates are given. The results presented in the paper can be used for analysis of MapReduce and multi-stage big data processing.

Keywords—big data processing; hypoexponential computing services; analytical models; performance and availability of queueing systems; discrete-event simulation

I. INTRODUCTION

The most common concept for big data processing is MapReduce. This is a two-stage information processing technique, where Map and Reduce procedures are distributed among some processors integrated in a cluster. Usually the number of map tasks in the MapReduce system is more than the number of workers in the cluster. In that case the task execution time for each worker has a hypoexponential distribution (Fig. 1).

![Graph showing computer loading in case when the number of map tasks is much more than the number of workers](image)

Moreover, at present the big data processing technologies with more than two stages attract more and more attention. These technologies may be the result of two or more MapReduce procedures being deployed one after another as a conveyor. Another possibility is an implementation of advanced big data processing techniques, represented by an arbitrary acyclic graph [2, pp. 41–45].

Queueing theory is an important technique for modeling and analysis of such systems. It enables efficient analyzing of the performance, availability and some other characteristics of a data processing system. The queueing system consists of a set of servers with a buffer for income requests. The requests in such a system may be, for example, the network packets or users’ computing tasks. A server may be a CPU performing pipeline operation or a firewall applying some sequence of rules to the input packets. For the purpose of big data processing, we assume that the requests are some types of users’ application tasks, e.g., the web search requests or more complicated data mining requests.

While analyzing the queueing systems the most frequent assumption is that the time of each processing stage has an exponential distribution. This assumption may be not adequate to real processing in all cases. However, it makes the performance and availability analysis very simple. Most of the performance characteristics can be analytically expressed in that case. In this work we initiate the investigation of finite queueing systems with more than two sequential processing stages, where each stage has exponential distribution.

The rest of the paper is organized as follows. In section 2 we analyze the related works in the area. Further, section 3 gives the mathematical background of hypoexponential distribution. In sections 4 and 5 we offer two analytical models of the finite queueing system with the hypoexponential service time. Section 6 describes the simulation of such a system. The numerical results, verifying analytical models by simulation, are discussed in section 7. Section 8 concludes the study and identifies a few directions of further research.

II. RELATED WORKS

The queueing systems with a finite buffer (also known as the finite queueing systems) are the subjects of intensive study at present. Some interesting results for the finite queueing systems and their applications to the analysis of modern data processing technologies have been obtained recently. In particular, the following types of queueing systems have been already investigated: the two-stage system with different stage processing rates [3], the multi-stage system with equal service rates except the first stage [4], the two-stage system with more servers on the second stage [5], the three-stage system with many servers at the second or at the second and third stages [6, 7], the multi-stage system with the equal rates (Erlang service) [8]. The other series of works [9–12] are devoted to more complex finite queueing models of large-scale cloud services with the changing structure and a variable number of
The infinite queueing systems with the hypoexponential service time are the most general one among all multi-stage systems with one server at each stage and the exponential service times at each stage. The two-phase and Erlangian systems are the special cases of hypoexponential systems.

### III. BACKGROUND

It is well known that the exponential distribution is the most important and at the same time the most simple being used for the analysis in queueing theory. Let us suppose that a server in the queueing system processes requests in $N$ stages, or has a line of $N$ sequential servers. The $N$-stage service means that the next request cannot enter the first service stage until the previous request does not leave the $N$-th service stage.

Then suppose that each stage has the exponential service time and all mean service times are different in general case. So the time from the beginning of serving one request by the first server up to the end of serving this request by the $N$-th server has the hypoexponential probability distribution.

The stream of requests is considered to be Poisson. Let $\lambda$ be the requests’ arrival rate, $\gamma$ is the output rate, $K-1$ is the number of places in the buffer or queue. Additionally one and only one request can be on the service at one time moment. So the total number of requests in the system is $K$. All requests are processed according to the FCFS (First Come, First Served) discipline. Then the queueing system with the hypoexponential service time can be shown in Fig. 2.

Fig. 2. The finite queueing system with the hypoexponential service time

If all $N$ stages have different service rates $\mu_1, \mu_2, \ldots, \mu_n$, then the total service time has the following probability density function (pdf) [13, pp. 24-25]:

$$f_x(x) = \sum_{i=1}^{N} a_i e^{-\mu_i x}, \quad x > 0,$$

where $a_i = \prod_{j=1, j \neq i}^{N} \frac{\mu_j}{\mu_i}, \quad i = 1, N$.

The mean service time for this system is

$$\bar{X} = \sum_{j=1}^{N} \frac{1}{\mu_j}.\quad (2)$$

If the server has the stages with the equal rates, then according to [14, 15] the pdf is

$$f_x(x) = \prod_{j=1}^{N} \mu_j \cdot \sum_{i=1}^{N} \Phi_{k_i}(-\mu_i) x^{k_i-1} e^{-\mu_i x} \frac{(\tau_i - 1)!(l-1)!}{(l-1)!},\quad (3)$$

where $\Phi_{k_i}(t) = (-l)^{-l} \cdot (l-1)! \sum_{i=0}^{k_i} \left( \frac{t}{l} - 1 \right)^i \Omega(l)\Omega(l)$ is the set of all $i_j$ that $\sum_{j=1}^{l} i_j = -1$ for $j = 1, a$ and $r_j = (\mu_j + x)^{(l+1)}$.

$r_j$ is the number of stages having the service rate $\mu_j$, while $\sum_{j=1}^{a} r_j = N$.

$a$ is the number of different stage’s service rates.

The mean service time of such a system is

$$\bar{X} = \sum_{j=1}^{N} \frac{r_j}{\mu_j}.\quad (4)$$

It is obvious that the integration of the pdf gives the cumulative distribution function (CDF) in both cases.

Let

$$\rho = \frac{\lambda}{\bar{X}} = \begin{cases} \frac{\lambda}{\sum_{i=1}^{N} \frac{1}{\mu_i}} & \text{if } \mu_i \neq \mu_j \text{ for } \forall i, j = 1, N, \\ \frac{\lambda}{\sum_{i=1}^{a} \frac{r_j}{\mu_j}} & \text{in the other case}. \end{cases}\quad (5)$$

Formulas (1)–(5) are then used in the following analytical models of the queueing systems.

### IV. FIRST ANALYTICAL MODEL

For this model we use the embedded Markov chain to represent the states and transitions of the queueing system with the hypoexponential service time. Let $S = \{(k,n), k=0, K, n=0, N\}$ be the state space of the hypoexponential system, where $k$ denotes the number of requests in the system and $n$ denotes the number of the stage processing some request. State $(0,0)$ corresponds to the empty system. All other states $(k,n), k \neq 0, n \neq 0$ represent the states, where there are $k$ requests in the system, and the server is processing the only one request on the stage $n$. The state transition diagram is shown in Fig. 3.

Let $p_{k,n}$ be the steady-state probabilities at the state $(k,n)$. Then the steady-state balance equations are the following:

state $(0,0)$: $0 = -\lambda p_{0,0} + \mu_n p_{0,n}$, \quad (6)

state $(1,1)$: $0 = -\left(\lambda + \mu_1\right) p_{1,1} + \lambda p_{0,1} + \mu_n p_{2,n}$, \quad (7)

state $(1,j)$: $0 = -\left(\lambda + \mu_j\right) p_{1,j} + \mu_j p_{0,j+1} + \mu_n p_{2,n}$, \quad \text{for } j = 2, K-1 \quad (8)

state $(i,1)$: $0 = -\left(\lambda + \mu_j\right) p_{i+1,1} + \mu_j p_{i,j+1}$, \quad \text{for } i = 2, K-1, j = 2, N \quad (9)

state $(i,j)$: $0 = -\left(\lambda + \mu_j\right) p_{i+1,j} + \lambda p_{i-1,j} + \mu_n p_{i+1,n}$, \quad \text{for } i = 2, K-1, j = 2, N \quad (10)

state $(K,1)$: $0 = -\mu_K p_{K,1} + \lambda p_{K-1,n}$ \quad (11)
state \((Kj)\): \[
0 = -\mu_j p_{k,j} + \lambda p_{k-1,j} + \mu_{j-1} p_{k,j-1},
\]
\(j = 2, N-1 \tag{12}\)

state \((KN)\): \[
0 = -\mu_N p_{K,N} + \lambda p_{K-1,N} + \mu_{N-1} p_{K,N-1},
\]
\(j = 0, \ldots, N-1 \tag{13}\)

Then the performance and availability characteristics can be derived as follows.

The mean system throughput \(\gamma\) is the departure rate or the rate of successful finishing of the \(N\)-th service stage, so
\[
\gamma = \mu_N \sum_{j=1}^{N} p_{j,N} \quad \text{(this formula is used for the numerical computations further)}.
\]
Alternatively it can be shown that
\[
\gamma = \frac{1}{\bar{K}} \quad \text{or} \quad \gamma = \lambda (1 - P_{\text{loss}}),
\]
where \(P_{\text{loss}}\) is the request loss (or blocking) probability.

Using (5), \(P_{\text{loss}}\) can be expressed as
\[
P_{\text{loss}} = 1 - \frac{1}{\rho}.
\]
Alternatively \(P_{\text{loss}} = \sum_{j=1}^{N} p_{j,j}\) (this formula is used for the numerical computations further), so
\[
\gamma = \frac{\lambda}{1 - \sum_{j=1}^{N} p_{j,j}}.
\]

The mean number of requests in the queueing system is
\[
\bar{K} = \sum_{j=1}^{N} \sum_{j=1}^{N} p_{j,j}.
\]

The mean time spent by a request in the system if it was not rejected at the entrance to the queue because of the buffer overflow is expressed as
\[
W = \frac{\bar{K}}{\gamma} = \frac{1}{\rho} \frac{1}{1 - \sum_{j=1}^{N} p_{j,j}},
\]
using Little’s law [16, pp. 52–55].

The server utilization coefficient can be found as
\[
U = \gamma \bar{X}.
\]

And the server’s availability to the other tasks with the lower priorities (if the main task has the highest priority) is
\[
V = 1 - U = 1 - \gamma \bar{X}.
\]

V. SECOND ANALYTICAL MODEL

The finite queueing system with the hypoexponential service time can be modeled as the \(M/G/1/K\) queueing system with the Poisson arrival rate \(\lambda\), the arbitrary distributed service time and the queue size of \(K\), including the request in service.

The approach of [17] is used for the second analytical model’s construction. The approach is based on computing the steady-state probabilities at the moments of departure of requests from the queue. It allows to find the queue size distribution in the arbitrary time moments. To find the steady-state probabilities immediately after the service completion, we should compute the transition probabilities of the Markov chain at the steady state.

Let \(a_k\) be the probability of \(k\) new requests arriving to the queue during the request’s service time. We can derive expression for \(a_k\) as follows:
\[
a_k = \frac{1}{k!} \left( \frac{\lambda x}{k!} \right)^k e^{-\lambda x} f(x) dx,
\]
\(14\)
where \( f(x) \) is the convolution of the density functions of \( N \) random service times.

There are two possible cases:

1) all stages of the queueing system have the different mean service times: \( \mu_i = \mu_j \) for \( i \neq j, 0 < i < N \).

2) the queueing system has a different stage’s service rates \( \mu_j \), such that \( r_j \) are the numbers of stages with rate \( \mu_j \), and \( \sum r_j = N \).

Case 1. Using (1), it follows from (14) that

\[
\alpha_i = \frac{\lambda^i}{k!} \sum_{j=1}^{\infty} a_j \mu_j e^{-\lambda i \mu_j} dx = \frac{\lambda^i \pi}{k^i} \sum_{j=1}^{\infty} a_j \mu_j e^{-\lambda i \mu_j} dx, \quad \text{where}
\]

\[
a_i = \prod_{j=1}^{N} \frac{\mu_j - \mu_i}{\mu_j}. \quad i = 1, N.
\]

So,

\[
\alpha_i = \frac{\lambda^i}{k!} \sum_{j=1}^{\infty} a_j \mu_j \int_0^\infty x e^{-(i+j) \mu_j} dx.
\]

Using the integral representation of Gamma function

\[
\Gamma(s) = \int_0^\infty x^{s-1} e^{-x} dx,
\]

it is possible to conclude that

\[
\alpha_i = \frac{\lambda^i}{k!} \sum_{j=1}^{\infty} a_j \mu_j \frac{\Gamma(k+1)}{(\lambda+\mu_j)^{k+1}} = \frac{\lambda^i \pi}{k^i} \sum_{j=1}^{\infty} a_j \mu_j \frac{\Gamma(k+1)}{(\lambda+\mu_j)^{k+1}}.
\]

Case 2. Using (3), it follows from (14) that

\[
\alpha_i = \frac{\lambda^i}{k!} B \sum_{r=1}^{\infty} \Phi_{r,1}(t) \frac{(\lambda^i \mu_j)^{r-1}}{(r-1)!} \int_0^\infty x e^{-(i+j) \mu_j} dx, \quad \text{where}
\]

\[
B = \prod_{j=1}^{N} \frac{\mu_j}{\mu_j}, \quad \Phi_{r,1}(t) = (-1)^{r-1}(1-t)^r \frac{(r-1)!}{r} \sum_{j=1}^{r-1} \left( \begin{array}{c} i+j \mu_j \, r \end{array} \right) \Omega_j(t), \quad \Omega_j(t) = \Omega_{j+1}(t) - \Omega_j(t). \]

is the set of all \( j \) that \( \sum i_j = 1 \) for \( j = 1, a \) and

\[
r_j = (\mu_j + x)^{r_j}. \quad \text{So,}
\]

\[
\alpha_i = \frac{\lambda^i}{k!} B \sum_{r=1}^{\infty} \Phi_{r,1}(t) \frac{(\lambda^i \mu_j)^{r-1}}{(r-1)!} \int_0^\infty x e^{-(i+j) \mu_j} dx.
\]

Let

\[
I_i = \int_0^\infty x e^{-(i+j) \mu_j} dx.
\]

Using the integral representation of Gamma function, we have

\[
I_i = \frac{\Gamma(k+i)}{(\lambda+\mu_j)^{k+i}}. \quad \text{And finally,}
\]

\[
\alpha_i = \frac{\lambda^i}{k!} B \sum_{r=1}^{\infty} \Phi_{r,1}(t) \frac{(\lambda^i \mu_j)^{r-1}}{(r-1)!} \frac{\Gamma(k+i)}{(\lambda+\mu_j)^{k+i}}.
\]

The following arguments are valid for both cases.

Let \( p_0 = \text{Prob}(L_{n+1} = j | L_n = i) \) where \( L_n \) and \( L_{n+1} \) are the numbers of requests in the system immediately after departure of \( n \)-th and \( (n+1) \)-th requests respectively.

The probabilities can be found as follows:

\[
p_{ai} = \frac{\alpha_i}{k} \sum_{j=1}^{\infty} a_j \mu_j e^{-\lambda i \mu_j} dx, \quad k = 0, K - 2, \quad \text{for} \ j = 0.
\]

\[
p_{ai} = \frac{\alpha_i}{k} \sum_{j=1}^{\infty} a_j \mu_j e^{-\lambda i \mu_j} dx, \quad k = K - 1, \quad \text{for} \ j = 1, K - 1.
\]

Using the state transition probabilities (15), (16), the steady-state probabilities \( \{ \pi_k ; k = 0, K - 1 \} \) at the departure moments can be computed by solving \( K - 1 \) balance equations

\[
\pi_i = \sum_{j=0}^{i} \pi_j p_k, \quad k = 0, K - 1.
\]

with the normalization condition \( \sum_{i=0}^{K-1} \pi_i = 1 \).

The equations (17) can be solved recursively. At first step we compute the ratios:

\[
\frac{\pi_{k+1}}{\pi_0} = \frac{1}{\pi_0} \sum_{j=0}^{k} \pi_j \pi_{k-j-1}, \quad k = 0, K - 2. \quad (18)
\]

where \( \pi_{k+1} = 1 \) for \( k = 0 \), then compute \( \pi_0 = \frac{1}{1 + \sum_{i=0}^{k} \pi_i} \).

At the second step substituting \( \pi_0 \) into (18), we compute all other state probabilities \( \{ \pi_k ; k = 1, K - 1 \} \).

Let \( p_k \), \( k = 0, K \) be the probabilities that there are exactly \( k \) requests present in the system at an arbitrary time moment. If \( P_{\text{loss}} \) is the equilibrium probability to lose the arrival request because of the full queue, then \( p_k = (1 - P_{\text{loss}}) \pi_k \), \( k = 0, K - 1 \).

It is obvious that the mean system throughput can be expressed as \( \gamma = \lambda (1 - P_{\text{loss}}) \) or as \( \gamma = \frac{1 - p_0}{\rho} = \frac{\lambda}{\rho} (1 - p_0) \).

Equating the right parts of two expressions, we get

\[
P_{\text{loss}} = 1 - \frac{1 - p_0}{\rho}. \quad \text{As} \quad p_0 = (1 - P_{\text{loss}}) \pi_0, \quad \text{finally} \quad P_{\text{loss}} = 1 - \frac{1}{\pi_0 + \rho}.
\]

The mean number of requests in the system can be expressed as

\[
\bar{K} = \sum_{k=0}^{K} k p_k = \frac{1}{\pi_0 + \rho} \sum_{k=1}^{K} k \pi_k + K \left( 1 - \frac{1}{\pi_0 + \rho} \right).
\]

Using Little’s theorem [16, pp. 52–55], the mean time spent by a request in the system can be expressed as

\[
W = \frac{\bar{K}}{\gamma} = \frac{1}{\lambda} \sum_{k=1}^{K} k \pi_k + K \left( \pi_0 + \rho - 1 \right),
\]

and the mean time spent in the queue (before the processing beginning) can be expressed as
**W_q=W−X=1λ(∑k=0k−1kπk+K(πk+ρ−1)−ρ).**

VI. SIMULATION

Although two analytical models are numerically equivalent, we verified both models by discrete-event simulation of the finite queueing system with the hypoexponential service time. The simulation is implemented according to recommendations of [18].

We used Python 3.3 programming language with the standard libraries and additionally “matplotlib” library for visualizing the results. The standard Python’s pseudorandom generator was used for generating uniformly distributed random values. Then they were transformed into the exponentially distributed random values using the standard approach: \( ERV = \frac{1}{\lambda} \ln(URV) \), where \( URV \) are the random values uniformly distributed on \([0;1]\), \( ERV \) are the exponentially distributed random values, and \( \lambda \) is the parameter of exponential distribution (in our case the rate of income arrival requests or the rate of stage service).

The exponentially distributed random values are used for inducing the interarrival periods for the income requests. To model the service periods we use the sum of exponentially distributions with different parameters.

To improve the accuracy of computations and to get a single point on the plot, we modeled from \( 2 \times 10^4 \) to \( 5 \times 10^7 \) requests service cycles and took the mean values obtained in the simulation. The criterion of stopping the modeling is the service completion for a given number of requests that can be configured by the user. All other parameters of the simulated queueing system, being essential for modeling, can be also configured by the user.

VII. NUMERICAL RESULTS AND DISCUSSION

The numerical results were obtained for different values of incoming mean requests’ arrival rates \( \lambda \). Fig. 4 shows some of these results. We study the mean system throughput, the mean requests delay and the service availability for another, low prioritized tasks.

For our numerical examples (Fig. 4) we have chosen \( \lambda \) in the interval from 0 to 200 requests per second (Rps), the queue length (the buffer size) \( K=\{2,5,10,25\} \) (for \( K>25 \) the system behaves itself as a system with an infinite queue) and other parameters as follows: \( \alpha=2, \mu_1=\frac{1}{5 \times 10^{-4}} \) Rps, \( r_1=2, \mu_2=\frac{1}{6 \times 10^{-3}} \) Rps, \( r_2=3 \).

The solid lines present the results of analytical modeling for the models 1 and 2 (they are completely numerically equal within the computing precision). The separate points present the simulated results with the same parameters and conditions. The numerical results obtained by the analytical modeling and simulation are in good correspondence with each other. Therefore, this fact indicates the correctness of our analytical models.

It should be stressed out that the first analytical model is universal for any finite queueing system with the hypoexponential service time. The second analytical model has differences in two cases. The first case in related to the situation when all service rates are different. In the second case some of these rates can be equal. This case is very complicated for implementation. Hence, it is more convenient to use the approximation of the second case by the first one, when the rates for some stages are defined not equal but very similar to each other (for example, for the two-stage queueing system instead of two equal rates of \( \frac{1}{5 \times 10^{-4}} \) Rps for both stages we define one stage’s rate of \( \frac{1}{5 \times 10^{-4}} \) Rps and the second stage’s rate of \( \frac{1}{4.999 \times 10^{-4}} \) Rps).

This approximation is possible because all performance characteristics studied are the continuous functions of the stage’s service rates. We specially investigate these dependencies, some examples of which are shown in Fig. 5.

In our example we take the two-stage system with \( K=25 \), the constant first stage’s service rate of \( \mu_1=\frac{1}{3 \times 10^{-4}} \) Rps and the variable second stage’s service rate of \( \mu_2=\left[0,\frac{1}{6 \times 10^{-3}}\right] \) Rps at the different values of \( \lambda=\{100,200,300\} \) Rps.

![Fig. 4. The example of performance and availability characteristics for the finite queueing system with the hypoexponential service time as the functions of the requests’ arrival rates: o for K=2; + for K=5; * for K=10; x for K=25](image-url)
All dependencies are continuous. Therefore, the minor changes in the service rates of some stages do not significantly influence the performance characteristics. If the system has many stages with the same rates, we recommend to slightly modify the rate values in both directions, e.g. about half of the rates are a little less and the rest of the rates are a little more than in the investigated system.

The plots in Fig. 4 have the shape typical for many finite queueing systems (for example [3, 8]). It can be seen that the plot of the average system throughput has the saturation point at the requests’ arrival rate $\lambda$ with the value approximately equal to the reciprocal of the mean service time: $\lambda_{sat} = \frac{1}{X}$.

Other plots have similar sharp with increase (decrease) in the saturation point.

For the plots from Fig. 4 the saturation point is $1 \times 10^{2} + 3 \times 10^{-7} = 52$ Rps. This fact could be explained.

When $\lambda$ is low enough, there are no requests in the buffer and these requests follow straight to the server. When $\lambda$ increases, the queue in the buffer becomes longer and longer. In addition, when the service rate is higher than the saturation point, the buffer overflows, a significant part of requests remains unserved and the system throughput is exhausted.

VIII. CONCLUSION

The main result of the present research is the creation of two analytical models of the finite queueing system with the hypoexponential service time. The models are verified using the discrete-event simulation. The developed models can be useful in modeling of the big data processing systems based on MapReduce and many other cloud-based and specific network services (like Next-Generation FireWalls – NGFWs). These models allow to obtain the main performance and availability characteristics of the systems under investigation.

In our future research we are going to study the finite queueing systems with the Coxian and hyper-Erlangian service time.

REFERENCES


