ANALYSIS AND OPTIMIZATION OF EXPONENTIAL QUEUEING NETWORKS BY MEANS OF PARALLEL CALCULATIONS

A. Pankov, E. Koluzaeva

Grodno State University
Grodno, Belarus
koluzaeva@gmail.com

The method of a finding of mean characteristics of an open exponential queueing network (QN) and solution of the problem of coasts optimization in such network by means of parallel calculations is considered. The software for solution of the formulated problems, developed with usage of technology of programming MPI and libraries PETSc is described.

*Keywords:*queueing network, parallel calculations.

1. INTRODUCTION

Now one of the most complicated problems in the world is considered the solution of tasks which need big calculation resources. During research of the time-probability characteristics of the QN with usage of classical methods [1] it is often necessary to solve systems of great number of difference-differential equations which they satisfy. The number of equations in these systems corresponds to number of the network states and grows exponentially with adding in the network at least one message or new queueing system (QS) (for closed networks). For open QN the situation especially worse cause the number of states of such networks is countable. Therefore evaluation of characteristics of QN by regular computer machines can vary from several hours (for numerical methods) to several days and more (for analytical methods). The next step in development of QN research methods is usage of high performance computing (HPC) for calculations. However, for parallel computing one needs to develop special software. That is time and effort consuming process.

The problem of finding of the exponential QN characteristics as well as solution of the network cost coasts optimization problem using parallel computations are considered in the paper.

2. FINDING OF MEAN CHARACTERISTICS OF

JACKSON OPEN NETWORK

Let's consider an open QN consisting of n QS S_1, S_2, ..., S_n. For unification we enter QS S_0, corresponding to an external environment. Each system has FIFO service discipline. An arrival flow of messages in the network is Poisson with rate λ. Service times of messages in various QS are independent and don't depend on flows arriving in the systems; thus the service time of the messages in i-th QS has an exponential distribution with parameter $\mu_i(l)$, where l – number of messages in QS. The message serviced by i-th QS, passes into j-th QS with probability p_{ij} , and leaves the network with probability p_{i0} , $\sum\limits_{j=0}^{n} p_{ij} = 1$, $i = \overline{1,n}$.

From the Jackson expansion theorem [2] follows that final stationary probabilities of Jacson's network states look like

$$P(k) = \prod_{i=1}^{n} P_i(k_i), \tag{1}$$

where $P_i(k_i)$, $P_i(0)$ are defined by following:

$$P_i(k_i) = \frac{\lambda e_i}{\mu_i(k_i)} P_i(k_i - 1) = \frac{\lambda e_i}{\mu_i(k_i)} \frac{\lambda e_i}{\mu_i(k_i - 1)} P_i(k_i - 2) = ... = \prod_{l=1}^{k_i} \frac{\lambda e_i}{\mu_i(l)} P_i(0), \tag{2}$$

$$P_i(0) = \left[1 + \sum_{k_i=1}^{\infty} \prod_{l=1}^{k_i} \frac{\lambda e_i}{\mu_i(l)} \right]^{-1}, \quad i = \overline{1,n}. \tag{3}$$

Here values e_i, $i = \overline{1,n}$, are roots of linear equations system:

$$e_j = p_{0j} + \sum_{i=1}^{n} e_i p_{ij}, \quad j = \overline{1,n}. \tag{4}$$

Thus, in steady state mode the number of messages in the network systems at each fixed time instant is independent random variables.

In case when systems of the Jackson open network are multilinear, i.e. QS S_i consists of m_i identical service lines, and service rate of messages equals μ_i, $i = \overline{1,n}$, for each line, steady probabilities of the network states also look like (1), thus

$$P_i(k_i) = \begin{cases} \frac{\rho_i^{k_i}}{k_i!} P_i(0), k_i \le m_i, \\ \frac{\rho_i^{k_i}}{m_i! m_i^{k_i - m_i}} P_i(0), k_i > m_i, \end{cases}$$

$$P_i(0) = \left[\sum_{j=0}^{m_i} \frac{\rho_i^j}{j!} + \frac{\rho_i^{m_i+1}}{m_i!(m_i - \rho_i)} \right]^{-1},$$

here $\rho_i = \dfrac{e_i \lambda}{\mu_i}$, $i = \overline{1,n}$.

Knowing values of network states probabilities, it is possible to calculate various probability characteristics of the QN. For example, average number of messages N_{i0} in system S_i queue, average number of messages N_i in system S_i (in queue and on service) and average time of stay τ_i in system S_i are finding by

$$N_{i0} = P_i(m_i)\frac{\rho_i}{(m_i - \rho_i)^2}, \quad i = \overline{1,n},$$

$$N_i = N_{i0} + \sum_{k_i=1}^{m_i-1} k_i P_i(k_i) + \frac{P_i(m_i)}{1 - \frac{\rho_i}{m_i}}, \quad i = \overline{1,n},$$

$$\tau_i = \frac{N_i}{\rho_i \mu_i}, \quad i = \overline{1,n},$$

where $\sum\limits_{k_i=1}^{m_i-1} k_i P_i(k_i) = 0$ when $m_i = 1$.

Proceeding from theoretical results follows that calculation of state probability using (1) can be easy parallelized, for example, each factor $P_i(k_i)$, $i = \overline{1,n}$, can be computed by separate processor irrespectively of other factors. Probability computation process $P_i(0)$, $i = \overline{1,n}$, can also be parallelized, for example, each item of the infinite sum in (3) can be evaluated irrespective of other items on different processors of a computing cluster.

Solution of linear equation system by computing clusters is well studied problem. The matrix of the equation system (4) is rarefied since the matrix of probabilities of message transitions between QS is such. So the problem of values e_i, $i = \overline{1,n}$, finding is reduced to usage of one of well studied methods of the solution of the equation systems multisequencing [4].

3. COASTS OPTIMIZATION IN OPEN JACKSON NETWORK

Let's consider the problem of coasts minimization of the QN by the number of service lines in the queueing systems. We introduce next designations: d_i – coast of one message in i-th QS and E_i – coast of one service line in i-th QS, $i = \overline{1,\ n}$. So the network optimization problem is

$$\begin{cases} W(m) = W(m_1,\ m_2,\ ...,\ m_n) = \sum\limits_{i=1}^{n} (d_i N_i(m_i) + E_i m_i) \to \min\limits_{m_1,\, m_2,\, ...,\, m_n}, \\ m_i \le a_i,\ i = \overline{1,\ n}, \end{cases} \tag{5}$$

where a_i – given numbers in which limits the number of service lines in i-th QS can change, $i = \overline{1,\ n}$.

The problem (5) is integer programming problem, it could be solved by exhaustive search mathod. However even for the networks of rather small dimension with number of the systems in the network equal, for example 20, and number of service lines in systems equal to 10, the number of such searches will be 10^{20} and the solution of the problem (5) on one processor could occupy some days.

The MPI technology was applied for parallel computing to solve the optimization problem (5). Software based on programming library PETSc [3] was developed. Open source library PETSc, distributing by GPL license, was developed taking into account usage in the big scalable applications. PETSc includes a huge set of subroutines for parallel solving of linear and non-linear equations that are easily integrated into applications written in programming languages C, C++, Fortran and Python. Also PETSc provides set of mechanisms used for development of parallel programs, for instance creation of parallel matrixes and vectors, allowing to lead a data interchange and to carry out various calculations.

The criterion in optimization problem (5) can be presented in a matrix form:

$$\mathbf{W}\left(m^{(l)}\right)_L = \mathbf{M}\left(m^{(l)}\right)_{L\times 2n} \times \mathbf{E}_{2n} \to \min_{m^{(l)},\, l=\overline{1,L}},$$

where $\mathbf{E}_{2n} = (d_1,\, ...,\, d_n,\, E_1,\, ...,\, E_n)^T$,

$$\mathbf{M}\left(m^{(l)}\right)_{L\times 2n} = \left\| \begin{array}{cccccc} m_1^{(1)} & ... & m_n^{(1)} & N_1\left(m^{(1)}\right) & ... & N_n\left(m^{(1)}\right) \\ m_1^{(2)} & ... & m_n^{(2)} & N_1\left(m^{(2)}\right) & ... & N_n\left(m^{(2)}\right) \\ & & & ... & & \\ m_1^{(L)} & ... & m_n^{(L)} & N_1\left(m^{(L)}\right) & & N_n\left(m^{(L)}\right) \end{array} \right\|,$$

where $m^{(l)} = \left(m_1^{(l)},\, m_2^{(l)}, ...,\, m_n^{(l)}\right)$, $m_i^{(l)}$ – number of service lines in i-th system in l-th search, $L = \prod_{i=1}^{n} a_i$ – number of all possible searches, $l = \overline{1,L}$, $i = \overline{1,n}$.

The process of problem (5) solving with library PETSc is reduced to creation of parallelized matrix $\mathbf{M}\left(m^{(l)}\right)_{L\times 2n}$ arranged line by line on all processors, and further multiplication of it by parallelized vector $\mathbf{E}_{2n}$. The resulting vector $\mathbf{W}\left(m^{(l)}\right)_L$ will be also parallelized. The minimum of coast criterion is a minimum element of vector $\mathbf{W}\left(m^{(l)}\right)_L$ with number l^*, and the solution of problem (5) is the first n elements of a line with number l^* in matrix $\mathbf{M}\left(m^{(l)}\right)_{L\times 2n}$.

4. PROGRAM IMPLEMENTATION

From implementation point of view, the developed software is a bunch of several utilities, each one is dedicated to one task: utility of intermediate and auxiliary matrixes creation, utility for intermediate calculations obtaining (for computing solution of set of linear algebraic equations on a cluster), utility for calculation of QN's characteristics on

a cluster, optimization problem utility and also set of utilities for vectors and matrixes visualization.

Specified partition into set independent subroutines allowed to make process of characteristics finding as transparent as possible for end user. Cluster resources are used more effectively since temporary results are not computing several times and could be taken from some intermediate storage. The only user interface with developed utilities is command line, all parameters are specified by command line arguments.

The given set of developed utilities could be used not only on a supercomputer but on regular computers driven by UNIX or Linux operating system with PETSc installed, for example when dimension of task being solving is rather insignificant.

5. THE EXAMPLE

Let's consider queueing network consisting of $n = 5$ QS, the number of service lines in each system can vary from 1 to 10, so the number of possible searches equal $L = 160807$. Probabilities of messages transition between systems are equal $p_{ii+1} = 1,\ i = \overline{0,\ n-1}$, $p_{n0} = 1$, all remaining probabilities in transition matrix are equal to zero. The vector of cost coefficients is $E_{2\mathrm{n}} = (22,\ 31,\ 11,\ 21,\ 13,\ 15,\ 21,\ 43,\ 54,\ 32)^T$.

Solution of optimization problem (5) took about 53 mins on regular AMD64 Linux-based machine in single-processor mode (1 core was used). It took about 17 mins on the same machine with 2 cores dedicated. Obtained results are: $W\left(m^{(l^*)}\right) = 198.275$, $m^{(l^*)} = (2, 1, 2, 2, 3)$, $N\left(m^{(l^*)}\right) = (0.07, 0.953, 0.052, 0.126, 0.009)$.

REFERENCES

1. *Ivnitsky V.A.* Theory of queueing networks, Fizmatlit, Moscow, 2004.
2. *Matalytsky M.A., Tikhonenko O.M. and Pankov A.V.* Queuing theory and its application, GrSU, Grodno, 2008.
3. http://www.mcs.anl.gov/petsc/petsc-as/
4. *Toporkov V.V.* Models of the distributed calculations, Fizmatlit, Moscow, 2004.