Jackson-network model for routing optimization in volunteer computing

Ruslan Krenzler
Universität Hamburg

Alexander Rumyanzev
Institute of Applied Mathematical Research, Karelian Research Centre of RAS

October 15, 2014

Contents

We provide a mathematical model for optimal routing in a volunteer computing. The model expresses the process of computing in a Desktop Grid in terms of Jackson network. An optimization problem for finding the maximum workunits generation rate with constraints on the node utilization is considered. A numerical experiment is stated, based on data from a BOINC project, an optimal solution is obtained. Applicability of results is discussed.

Introduction

The problem of optimal routing in volunteer computing is evidently one of the main problems the researcher has to solve while planning a new project or running the existing one. A typical application in a Desktop Grid environment is split into workunits, each of them provides (one or more) results sent to hosts (nodes of the GRID) by a server. In case of an error in getting the result back from host (due to deadline violation or computation error) the server has to resend the result to (same or another) host, until the maximum number of attempts/errors is reached. The hosts are configured in such a way to prevent decrease in users quality of experience. More exactly, the volunteer computing project can not take more computational power fraction than the constraint (which may be modified by user of the host). However, as the server does not have information about the busy/idle periods of hosts, it may generate tasks too intensively, that leads to increase in erroneous results, which in term increases the server load (due to need in re-sending the results). This may provide overall decrease of performance of the GRID, which may be expressed in terms of the total error rate. One of the solutions of the problem is to provide an optimal rate together with an optimal routing scheme.

The rest of the paper is organized as follows. In section 1 the model is presented. In section 2 the optimization problem is exposed. In section 3 the numerical experiment is discussed. As the numerical experiment is done over the BOINC system, in the rest of the paper we consider this type of volunteer computing environment.

1 Jackson network model

We have a system with a BOINC server and N hosts. The whole system must process a large task splitted in (infinitely) many workunits. The server feed the hosts with results, each of them correspond to a single workunit. The hosts process the results, and send them to the BOINC server. Occasionally the results are erroneous.

We model a system as a Jackson-Network [1] with $J = N + 1$ nodes, Poisson input rate λ, routing matrix r and service rates $\boldsymbol{\mu} = (\mu_1, ..., \mu_{N+1})$. The service times are considered to be exponentially distributed, they are independent from each other and from the Poisson input stream.

The customers of the queueing system are workunits, when they enter the server $N+1$, and they are results, when they leave the hosts $1,..,N+1$. The workunits leave the virtual node 0 and enter the system in the node $N + 1$ with a rate λ (this is the intensity of generation of new workunits). The node $N + 1$ models a part of the scheduler of the BOINC server and network infrostructure from server to the clients. We assume that the customers are processed in FCFS-order. The service rate μ_{N+1} models the inverse of time required for scheduling, routing and sending a result to host. The server sends each result to host j with probability p_j. These decisions are i.i.d. We define probability vector $\boldsymbol{p} = (p_1, \dots, p_N) \in [0, 1]^N$ with $\sum_{j=1}^{N} p_j = 1$.

The client service rates μ_i $j \in \{1, \dots, N\}$ desribe the average rate required for a client j to process a result, either erroneously or successfuly. Each service rate μ_j already includes daily, weakly and similar fluctuation of availability of the host j during the processing of the large task.

Sometimes a host j fails to process a workunit correctly or on time. We assume that the probability of the workunit to be faulty processed at the host j is i.i.d with probability s_j. In this case the results will be returned to the server $N + 1$ for next processing. We define failure probabilities vector $\boldsymbol{s} = (s_1, \dots, s_N) \in [0, 1)^N$. The successfully processed workunits are sent to the virtual node 0 and leave the system.

The routing matrix r of the system is

$$r = \begin{array}{c|c}
 & \begin{matrix} 0 & 1 & 2 & \dots & N-1 & N & N+1 \end{matrix} \\ \hline
\begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ N-1 \\ N \\ N+1 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1-s_1 & & & & & & s_1 \\ 1-s_2 & & & & & & s_2 \\ \vdots & & & & & & \vdots \\ 1-s_{N-1} & & & & & & s_{N-1} \\ 1-s_N & & & & & & s_N \\ 0 & p_1 & p_2 & \dots & p_{N-1} & p_N & \end{pmatrix}
\end{array}$$

See Figure 1 for system discription.

The server $N+1$ and the clients $j \in \{1, \dots, N\}$ are modeled as queueing systems with infinite number of waiting places, which is required by the Jackson-network model. We bound the individual arrival rates η_j in consideration of the service rates μ_j. We define a vector $\boldsymbol{b} = (b_1, \dots, b_{N+1}) \in (0,1)^{N+1}$ and require

$$\eta_j \leq b_j \mu_j \qquad \forall j \in \{1, \dots, N+1\}. \tag{1}$$

Note that this consideration is natural to the BOINC system, as the hosts may define the maximum fraction of time that the host may be used for computation of the project.

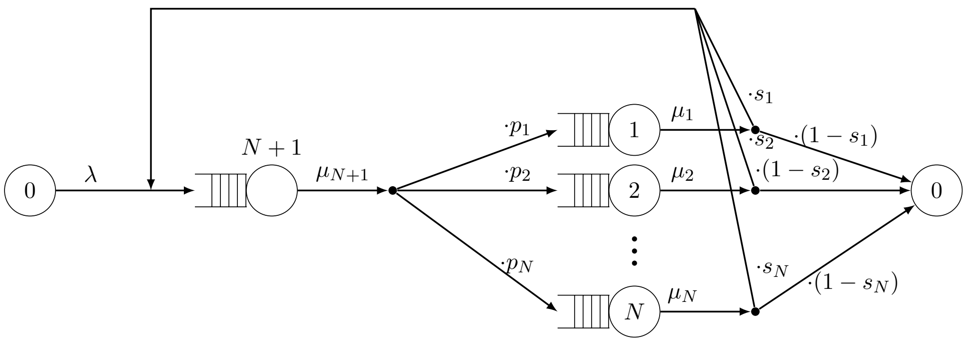

Figure 1: The Jackson-network model of the BOINC system

Mathematically, we describe the model as stochastic process $X = (X_j(t) : j \in \{1, \dots, N\}, t \in \mathbb{R}_0^+)$ where $X_j(t)$ describes number of customers on the node j at time t.

Proposition 1.1. *The solution of the traffic equation*

$$\eta_j = \lambda r(0,j) + \sum_{i \in J} \eta_i r(i,j), \qquad \forall j \in J$$

is

$$\begin{aligned} \eta_j &= \eta_{N+1} p_j, && \forall j \in \{1, \dots, N\} \\ \eta_{N+1} &= \frac{\lambda}{1 - \sum_{j=1}^N p_j s_j} \end{aligned}$$

Proof. For our particular system, the grid network, the traffic equation (**??**) is

$$\begin{aligned} \eta_j &= \eta_{N+1} p_j, \quad \forall j \in \{1, \dots, N\} \\ \eta_{N+1} &= \lambda + \sum_{j=1}^{N} \eta_j s_j. \end{aligned} \tag{2}$$

Therefore it holds

$$\begin{aligned} \eta_{N+1} &= \lambda + \sum_{j=1}^{N} \eta_{N+1} p_j s_j \\ &= \lambda + \eta_{N+1} \sum_{j=1}^{N} p_j s_j \\ \Longrightarrow \eta_{N+1} &= \frac{\lambda}{1 - \sum_{j=1}^{N} p_j s_j} \end{aligned}$$

□

2 Optimization with constant service rates μ_j

Problem 2.1. We have to find an optimal input rate λ and distribution vector $p \in [0,1]^N$ such that constraints for host utilization hold. The optimization problem is therefore to minimize

$$\begin{aligned} f : D &\longrightarrow \mathbb{R}^+ \\ (\lambda, p) : &\mapsto f(\lambda, p) := -\lambda \end{aligned}$$

on $D := (\mathbb{R}^+)^{N+1}$, under inequality constraints

$$\eta_j \le b_j \mu_j \qquad \forall j \in \{1, \dots, N+1\}$$

and under equality constraints

$$h(\lambda, p) := \sum_{j=1}^{N} p_j - 1 = 0$$

Proposition 2.2. *The equality constraints can be written as*

$$g_j(\lambda, p) \le 0$$

with

$$g_j(\lambda, p) = \lambda p_j + b_j \mu_j \sum_{j=1}^{N} p_j s_j - b_j \mu_j \qquad \forall j \in \{1, \dots, N+1\}$$

$$g_{N+1}(\lambda, p) = \lambda + b_{N+1} \mu_{N+1} \sum_{j=1}^{N} p_j s_j - b_{N+1} \mu_{N+1}$$

Proof.

$$
\begin{aligned}
&\eta_{N+1} p_j \leq b_j \mu_j \\
\iff &\frac{\lambda}{1-\sum_{j=1}^{N} p_j s_j} p_j \leq b_j \mu_j \\
\iff &\lambda p_j \leq b_j \mu_j (1 - \sum_{j=1}^{N} p_j s_j) \\
\iff &\underbrace{\lambda p_j + b_j \mu_j \sum_{j=1}^{N} p_j s_j - b_j \mu_j}_{:=g_j(\lambda,p)} \leq 0 \qquad \forall j \in \{1, \ldots, N+1\}
\end{aligned}
$$

and

$$\eta_{N+1} \leq b_{N+1} \mu_{N+1}$$

$$
\begin{aligned}
\iff \frac{\lambda}{1-\sum_{j=1}^{N} p_j s_j} &\leq b_{N+1}\mu_{N+1} \\
\iff \lambda &\leq b_{N+1}\mu_{N+1}(1 - \sum_{j=1}^{N} p_j s_j) \\
\iff \underbrace{\lambda + b_{N+1}\mu_{N+1} \sum_{j=1}^{N} p_j s_j - b_{N+1}\mu_{N+1}}_{=:g_{N+1}(\lambda,p)} &\leq 0
\end{aligned}
$$

□

Corollary 2.3. *Gradient functions*

$$\mathrm{grad} f(\lambda, p) = (-1, 0, ..., 0)$$

$$
\begin{aligned}
\mathrm{grad} g_j(\lambda, p) &= (p_j, b_j\mu_j s_1, \ldots, b_j\mu_j s_{j-1}, \lambda + b_j\mu_j s_j, b_j\mu_j s_{j+1}, \ldots, b_j\mu_j s_N) \\
\mathrm{grad} g_{N+1}(\lambda, p) &= (1, b_{N+1}\mu_{N+1}s_1, \ldots, b_{N+1}\mu_{N+1}s_{j-1}, b_{N+1}\mu_{N+1}s_j, \\
&\quad b_{N+1}\mu_{N+1}s_{j+1}, \ldots, b_{N+1}\mu_{N+1}s_N), \qquad j \in \{1, \ldots, N\}
\end{aligned}
$$

$$\mathrm{grad} h(\lambda, p) = (0, 1, \ldots, 1)$$

3 Numerical results

3.1 BOINC database information extraction

The main information on the hosts parameters is spread over three tables, namely `host, workunit, result`.

The service intensity μ_j for each host may be counted from the sum of `result.elapsed_time` divided by the number of records for the particular host. The probability of a host error s_j may be counted as the number of erroneous results for the selected host, divided by the total number of results dedicated to that host by server. The probability of routing is counted from the total number of results sent to the host by server, divided by overall number of results. The value b_j is the characteristic of a particular host and may be taken as the value `host.active_frac`.

The dataset was taken from a real BOINC project which used no replication technology (`workunit.target_nresults=1`), and the deadline was equal to 3 days. In order to clean the data, the table `workunit` was cleared from records with `id`$<$30000 (the setup period of the project). The table `result` was cleared from records having no workunits. The table `host` was creared from records on hosts which didn't provide any results. The total number of workunits involved in experiment were 927346. 75869 of them were erroneous, which is relatively high (the total number of attemts that could be made for each workunit for that particular project was 6). The SQL statements for data import using R code is as shown below:

```
dbConnect(MySQL(), dbname="test")
host=dbGetQuery(con, "select id as host,active_frac as active
from host where id in (select hostid from result);")
mean_service=dbGetQuery(con, "select sum(elapsed_time)/count(id) as ES,
hostid as host from result where outcome=1 group by hostid;")
prob_success=dbGetQuery(con, "select hostid as host,count(id) as prob
from result where outcome=1 group by hostid;")
prob_error=dbGetQuery(con, "select hostid as host,count(id) as error
from result where outcome!=1 group by hostid;")
df=merge(prob_success,prob_error,all=TRUE)
df$p=(df$prob+df$error)/sum(df$prob+df$error)
df$s=df$error/(df$prob+df$error)
host=merge(host,mean_service)
dbDisconnect(con)
```

As a result, the dataframe `host` contained all the necessary information to define the model. The optimization problem was solved with the help of NLopt package [2]. The optimal workunit generation rate was equal to 0.67, which is much lower than the value used in the project.

3.2 Discussion

The problem don't have a unique solution. After optimal input rate λ is obtained, we can optimize the total network utilization with this fixed λ value.

Note that the total "primary" error rate for the routing plan (produced by nodes without considering re-routed traffic) in case of a saturated system (when all nodes are working at their highest available level) will be equal to

$$\sum_{j=1}^{N} p_j s_j \mu_j b_j, \tag{3}$$

which by the solution of the optimization problem would be equal to 0.006878745. This would lead to approximately 6300 "primarily" erroneous workunits, which is much less than the number of errors obtained in the project. This outlines the practical importance of the model.

One of the features of the model is that the server applies routing directly, not waiting for the clients to ask for work. This assumption may be violated in a Desktop Grid utilizing PULL technology (as BOINC does), but may be true for an Enterprize Desktop Grid. In the latter case a PUSH technology seems to be the most promissing to guarantee robust computation time for each workunit.

References

[1] J.R. Jackson. Networks of waiting lines. *Operations Research*, 5:518–521, 1957.

[2] Steven G. Johnson. The nlopt nonlinear-optimization package.