A queueing-theoretic model for resource allocation in one-dimensional distributed analytics network[*]

Nitish K. Panigrahy[†], Prithwish Basu[¶], Don Towsley[†], Ananthram Swami[‡],
Kevin S. Chan[‡] and Kin K. Leung[§]
[†] University of Massachusetts Amherst, MA, USA. Email: {nitish, towsley}@cs.umass.edu
[¶]Raytheon BBN Technologies, Cambridge, MA 02138, USA. Email:prithwish.basu@raytheon.com
[‡]Army Research Laboratory, Adelphi, MD 20783, USA. Email:{ananthram.swami, kevin.s.chan}.civ@mail.mil
[§]Imperial College London, London SW72AZ, UK. Email: kin.leung@imperial.ac.uk

ABSTRACT

We consider the problem of allocating requesters of analytic tasks to resources on servers. We assume both requesters and servers are placed in a one dimensional line: $[0, \infty)$ according to two different Poisson processes with each server having finite capacity. Requesters communicate with the servers under a noninterference wireless protocol. We consider a "Move to Right" (*MTR*) request allocation strategy where each requester is allocated to the nearest available server to its right. We start our analysis from a single resource per request scenario where each requester demands a single computational resource. We map this scenario to an M/M/1 queue or a bulk service M/M/1 queue depending on the capacity of the servers. We compare the performance of the strategy with the globally optimal strategy taking "expected distance traveled by a request" (*request distance*) as performance metric. Next, we extend our analysis to two resources per request scenario. We show that it can be transformed into an equivalent fork-join queue problem. Numerical results are presented to validate the claim.

1. INTRODUCTION

Past few years have witnessed a significant growth in the use of distributed network analytics involving agile code, data and computational resources. In many of such networks, for example, Internet of Things (IoT) [3], a large number of computational and storage resources are widely distributed in the physical world. Thus the spatial distribution of resources plays an important role in determining the overall performance of the analytics network.

In a distributed analytics network, the requesters generating analytic tasks and the servers providing services to the tasks are distributed over a geographic region. We collectively term the requesters and the servers as "devices". Each analytic task may require a set of resources: computation, code and data resources to achieve computation objectives. The resources are placed on physical devices. For successful completion of each analytic task, an algorithm needs to execute the following functions [4]: (i) *Placement of computing resources:* i.e. basically determining which devices would perform the computation for the analytic task. (ii) *Retrieval of data/code resources:* The retrieval may involve communicating with the requesters that provide data/code directly or with the data/code servers that store them. (iii) *Single/Multi-hop routing:* which involves transferring the data/code through the network to the computing devices. (iv)*Handling limited computational capacity:* As the computational devices may have limited capacity, a viable algorithm should correctly place the computing resources on devices adhering to the capacity constraints.

In this work, we consider the placement of requesters and servers in a geographic region defined by a one dimensional line $\mathcal{L} : [0, \infty)$. The requesters and servers are placed according to two different Poisson processes. We assume any communication between the devices involves non-interfering single hop wireless transmission, also known as *Direct transmission model* in the literature [2]. We focus on analyzing the following two scenarios.

First, we consider the *single resource scenario*: In this case, we place the computing resources on servers according to "Move to Right" (*MTR*) request allocation strategy i.e. the geographically nearest available server to the right of the requester is allocated to the analytic task. We assume requesters provide on spot data and code for computation to the selected servers over a wireless channel. We find that the *single resource scenario* can be modeled as an M/M/1 or a bulk service M/M/1 queue depending on the capacity of the servers where capacity denotes the maximum number of requests that can be served by a server. We relate the "expected distance traveled by a request" (*request distance*) in the analytic network to the expected sojourn time for the corresponding queuing model . Using request distance as the performance metric, we compare MTR strategy to a globally optimal strategy.

Second, we consider the *two resource scenario*: Here, we place the computing resources on the requesters while the retrieval involves communicating with two sets of servers: the data servers and the code servers. We assume both data and the code servers are distributed over $\mathcal{L}$ according to a single homogeneous Poisson process. We also assume that due to interference and large data and code size, the data and code servers can only serve one request. Again we find that this scenario can be transformed into an equivalent fork-join queue problem. We validate our claim with numerical experiments.

[*]This research was sponsored by the U.S. ARL and the U.K. MoD under Agreement Number W911NF-16-3-0001. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

2. SINGLE RESOURCE SCENARIO

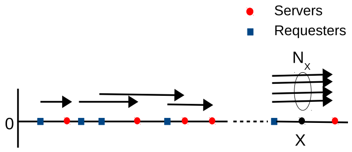

Figure 1: Poisson Placement of requesters and servers on the 1-D network with server capacity of one.

Consider a set of requesters R requesting analytics tasks of equal computation requirements and a set of computation servers S that can execute these requests. Assume that each server $j \in S$ has an execution capacity $C_j \in \mathbb{Z}^+$ measured in units of number of tasks being processed at the time. Suppose that requesters and servers are located in space $\hat{\mathcal{L}}$ equipped with a distance measure. Formally, let $r : R \to \hat{\mathcal{L}}$ and $s : S \to \hat{\mathcal{L}}$ be the location functions for requesters and servers, respectively, such that a distance measure $d_{\hat{\mathcal{L}}}(r, s)$ is well defined for all pairs $(r, s) \in R \times S$. In this section we examine the scenario where $\hat{\mathcal{L}} = \mathbb{R}^+ = \mathcal{L}$ (say), i.e., the positive real line and for the case where the locations of the requesters: $\{r(i)|i \in R\}$ and the locations of the servers: $\{s(j)|j \in S\}$ are Poisson distributed with different densities. We also assume that all servers have equal execution capacities i.e. $C_j = c\,\forall j \in S$.

Let λ be the requester density and μ the server density. The MTR rule for assigning requesters to servers is illustrated in Figure 1 where starting from the left requests are assigned to the closest available server on the right. Let $d_i, i \in R$ denote the distance between requester i and the server serving the requester. Last, define N_x to be the number of requests that traverse through point $X \in \mathcal{L}$ as shown in Figure 1. We primarily focus on deriving request distance for each request. We first focus on the case where each server has unit capacity i.e. $c = 1$.

2.1 Server capacity: c =1

When server capacity is one, the analytics network can be modeled as an M/M/1 queue. An M/M/1 queue consists of a single server with customer arrivals described by a Poisson process and customer service times by an exponential distribution. Thus the distance between two consecutive requesters in the analytics network can be thought of as inter-arrival time between customers in an M/M/1 queue. Similarly the distance between consecutive servers corresponds to a customer service time. In the analytics network, random variable d_i corresponds to the sojourn time of the i^{th} customer in the M/M/1 queue and N_x denotes the expected number of customers in the queue at time instant X. If $\lambda < \mu$, then d_i converges to a random variable $d \sim$ exponential$(\mu - \lambda)$ and N_x converges to a random variable $N \sim$ geometric$(1 - \lambda/\mu)$. Thus we can evaluate request distance as

$$\mathbb{E}[d] = \frac{1}{\mu - \lambda}. \tag{1}$$

2.2 Server capacity: c >1

When the server capacity is more than one, the analytics network maps to a bulk service M/M/1 queue. A bulk service M/M/1 queue provides service to a group of c customers. The server serves a bulk of at most c customers whenever it becomes free. The service time for the group is exponentially distributed and the customer arrivals are determined by a Poisson process. Thus similar to the previous case, the distance between two consecutive requesters in the analytics network can be thought of as inter-arrival time between customers in the bulk service M/M/1 queue. However, the distance between two consecutive servers should be mapped to a bulk service time.

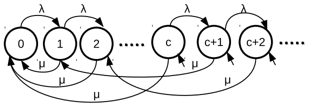

Figure 2: State space diagram for server capacity c with $c > 1$.

Having established an analogy between the analytics network and the bulk service M/M/1 queue, we now define the state space for the analytics network. Consider the definition of N_x as the number of requests that traverse through point $X \in \mathcal{L}$ under MTR strategy. In steady state, N_x converges to a random variable N provided $\lambda < c\mu$. Let π_k denote $\texttt{Pr}[N = k]$ where $k = 0, 1, \ldots$. The state space diagram for such a system is shown in Figure 2. Thus we have the following balance equations similar to that of a bulk service M/M/1 queue [5].

$$\begin{aligned}(\lambda + \mu)\pi_k &= \mu\pi_{k+c} + \lambda\pi_{k-1},\ k \geq 1,\\ \lambda\pi_0 &= \mu(\pi_1 + \pi_2 + \ldots + \pi_c).\end{aligned} \tag{2}$$

By taking the z-transform and following the procedure in [5], we obtain the steady state probability vector $\pi = [\pi_0, \pi_1, \ldots]$. By applying Little's formula, we obtain the following expression for the request distance.

$$\mathbb{E}[d] = \frac{r_0}{\lambda(1 - r_0)}, \tag{3}$$

where r_0 is the only root in the interval $(0, 1)$ of the following characteristic equation (with r as the variable).

$$\mu r^{c+1} - (\lambda + \mu)r + \lambda = 0. \tag{4}$$

2.3 Optimal allocation strategy

Next we consider formulating the optimal request allocation strategy for the analytics network taking request distance as metric. The objective is to find a function $\pi : R \to S$, neither necessarily injective nor surjective, such that the following is satisfied.

$$\arg\min_{\pi} \sum_{i \in R} d_{\mathcal{L}}(r(i), s(\pi(i))) \tag{5}$$

$$s.t. \quad \sum_{i \in R} \mathbb{1}_{\pi(i)=j} \leq C_j, \forall j \in S$$

The above capacity-constrained assignment problem can be modeled as a min-cost (single commodity) flow problem on a directed graph $G = (V, E)$, where $V = R \cup S \cup \{D_R\} \cup \{D_S\}$, where D_R is a dummy requester node and D_S is a dummy server node (note that this formulation allows a node to be both a requester and a server), and $E = R \times S \cup \{D_R\} \times R \cup S \times \{D_S\}$. Assign capacities and costs to all edges $e \in E$ as follows:

1. For $e \in \{D_R\} \times R : c_e = 1, w_e = 0$

λ	μ	c	$\overline{d}_{mtr}$	$\overline{d}_{opt}$	$\overline{d}_{opt}/\overline{d}_{mtr}$
0.9	1	1	10	4.11	0.41
0.9	1	2	1.48	0.6694	0.46
0.9	1	500	1	0.5011	0.5011

Table 1: Comparison of MTR and optimal allocation strategy where $\overline{d}_{mtr}$ and $\overline{d}_{opt}$ are the request distance in MTR (Equations (1) and (3)) and optimal strategy under single resource scenario.

2. For $e = (j, D_S), j \in S : c_e = C_j, w_e = 0$

3. For $e = (i, j) \in R \times S : c_e = 1, w_e = d_{\mathcal{L}}(r(i), s(j)$

Assign flow variables $x_e \in \mathbb{R}$ to each edge $e \in E$. The assignment problem of (5) corresponds to a min-cost flow problem where x_e obeys flow capacity constraints $x_e \leq c_e$ and flow conservation laws at each node in G, i.e., $\forall v \in V, \sum_{u:u \in \mathcal{N}_v^{in}} x_{(u,v)} = \sum_{w:w \in \mathcal{N}_v^{out}} x_{(v,w)}$, where $\mathcal{N}_v^{in}$ denotes the set of incoming neighbors of v and $\mathcal{N}_v^{out}$ denotes the set of outgoing neighbors of v.

This minimization problem can be solved by obtaining a solution to the min-cost flow problem after assigning demands to dummy variables as follows: $dem_{D_R} = -|R|$, $dem_{D_S} = |R|$. The rest of the nodes are assigned zero demand, i.e., $dem_v = 0, \forall v \in V \setminus (\{D_R\} \cup \{D_S\})$. Well-known polynomial time solutions exist for this problem, such as the network simplex algorithm with time complexities of $O(|R|^6 \log |R|)$ [7]. Moreover, since the edge capacity values c_e and demands are integral, by the flow integrality theorem [1], the optimal flow values x_e will also be integral even if the edge costs w_e are real-valued. Exploiting this property and the fact that the capacities of all edges in $\{D_R\} \times R$ and $R \times S$ are one, we get a valid 0-1 integral assignment to the corresponding flow variables x_e. That is, each request in R gets sent (assigned) to exactly one server in the optimal solution. Note that when server capacities C_j are all one, this degenerates to a minimum-weight matching problem on a weighted bipartite graph.

2.4 Performance Comparision

We compare the performance of MTR strategy with the optimal strategy obtained by solving optimization problem (5). We consider a collection of 500 requesters and 500 servers i.e. $|R| = |S| = 500$ and the results are presented in Table 1. The MTR strategy delivers performance close to a factor 2 of the optimal strategy. We also observe an increase in the performance of MTR strategy by increasing the server capacity c. Thus we have the following conjecture.

CONJECTURE 1. *Under identical conditions, the optimal to MTR approximation ratio, $\overline{d}_{opt}/\overline{d}_{mtr}$, increases with an increase in the capacity c of the servers .*

3. EXTENSION TO TWO RESOURCES

Now consider the following scenario where requesters request two resources, say data and code, which can reside on different servers. Let μ_1 and μ_2 be the densities of the data and code servers respectively. We again consider the MTR request allocation strategy. The analytics network, in this case, can be modeled as a two queue fork-join system [6]. In such a queue, each incoming job is split into two sub-jobs each of which is served on one of the two servers. After service, each sub-job waits until the other sub-job has been processed. They then merge and leave the system. In the analytics network as well, each request forks two subrequests one for data and the other for the code resource. A request is said to be completed only if it has retrieved both the resources, thus mapping it to a fork-join queue. We define the *overall request distance* to be the maximum value among data request distance and code request distance. Below we discuss a special scenario of data and code servers having the same density.

λ	μ	c	$\overline{d}_{exp}$	$\overline{d}_{th}$
1	2	1	1.4358	1.4375
0.9	1	1	14.00	13.875

Table 2: Experimental results for homogeneous two resources scenario where $\overline{d}_{exp}$ and $\overline{d}_{th}$ are the experimental and theoretical request distance.

3.1 Identical service rates ($\mu_1 = \mu_2$)

The approximated request distance for this scenario is obtained from the expression for the expected sojourn time of a fork join queue with homogeneous servers as [6]:

$$\mathbb{E}[d] = \frac{12\mu - \lambda}{8\mu(\mu - \lambda)}. \qquad (6)$$

We compare the experimental and theoretical values of request distance for a set of $|R| = |S| = 10^6$ devices in Table 2. The experimental results match with the theoretical results (obtained from Equation (6)).

4. CONCLUSION

We proposed a queuing theoretic model for analyzing the behavior of resource allocation in a one dimensional distributed analytics network. We studied two specific scenarios: single and two resources scenarios and mapped them to the corresponding queuing model. We also formulated the optimal request allocation strategy taking request distance as metric. Going further, we aim to extend our results in two ways. First we would like to extend the analysis for MTR strategy to a two dimensional geographic region. Second we would also consider analyzing different request allocation strategies such as move to right or left.

5. REFERENCES

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Inc, 1993.

[2] A. Asadi, Q. Wang, and V. Mancuso. A Survey on Device-to-device Communication in Cellular Networks. In *IEEE Commun. Surv. Tut*, 2013.

[3] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A Survey. *Computer Networks*, 54(15):2787–2805, 2010.

[4] A. Destounis, G. Paschos, and I. Koutsopoulos. Streaming Big Data Meets Backpressure in Distributed Network Computation. In *IEEE INFOCOM*, 2016.

[5] L. Kleinrock. *Queueing Systems.* John Wiley and Sons, 1976.

[6] R. Nelson and A. Tantawi. Approximate Analysis of Fork/join Synchronization in Parallel Queues. *IEEE Transactions on Computers*, 37(6):739–743, 1988.

[7] J. Orlin. A Polynomial Time Primal Network Simplex Algorithm for Minimum Cost Flows. *Mathematical Programming*, 78:109–129, 1997.