Automated Optimal Dispatching of Service Requests

Anubha Verma,
Nirmit V. Desai,
Anuradha Bhamidipaty,
Anshu N. Jain,
Jayan Nallacherry
IBM Research India
Bangalore, Karnataka 560071
Email: anubha.verma@in.ibm.com

Swapnoneel Roy*
State University of New York at Buffalo
Buffalo, NY 14220-2000, USA
Email: sroy7@buffalo.edu

Stephen Barnes
IBM Australia Ltd.
New South Wales, Australia
Email: stepbarn@au1.ibm.com

***Abstract*—In the services domain, the customers raise issues and service requests in the form of tickets. There is a pool of personnel who work on these tickets and resolve them. The problem at hand is to dispatch these tickets to the most appropriate personnel. Optimality is applied to metrics like the mean service time taken to resolve a ticket, the fair sharing of workload among the personnel, and the size and configuration of the pool. The current state of the art involves a human dispatcher for assigning incoming service requests. Though intelligent, a human dispatcher can be suboptimal with respect to the above-mentioned objectives due to the large space of parameter values to be considered. Further, there exists an opportunity to achieve high-level goals such as on-the-job training, eliminating over-production, and workload balancing among personnel through smarter dispatch decisions. For example, target skill levels of personnel can be achieved by assigning them tickets requiring those skills increasingly. Also, overproduction can be controlled by dispatching only those tickets that otherwise would be in the danger of missing deadline (SLO) constraints. Our work involves the design and implementation of an automated dispatcher which would take various characteristics of the tickets and the pool state as input and recommend an intelligent dispatching decision for the tickets, based on the above-mentioned goals and constraints.**

I. Introduction

The motivation behind this research is as follows:

- The current state of the art involves a ***human dispatcher*** for assigning incoming service requests. This results in ***subjective decision making***, and usually compromises on some of the objectives which could be potentially met through an appropriate dispatch decision.
- There exists an opportunity to ***translate high level policies*** such as ***cross learning, waste reduction/elimination, load balancing*** besides the better known objectives such as ***deadline (SLO) attainment, swing rule minimization*** into dispatch decisions.
- There is also scope for providing ***added value using analytics*** as a part of smarter dispatcher. Smarter dispatcher can provide insight into ***optimal pool size and configuration***.
- Overall, human dispatcher is typically a ***high-skilled resource*** and any assistance provided to him is ***cost/time saved***.

Our novel contribution is modeling automated dispatching process in two parts:

1) As an online dispatcher that produces real-time recommendations as the tickets arrive
2) As an offline analyzer that analyzes the historical dispatch data and recommends changes to pool configurations such as staffing levels.

Along these lines, we have a prototype that formulates the various pieces of the problem.

- Part I (Online phase): Dispatch the tickets to the personnel in order to
  1) Minimize instances where a ticket is assigned to over-skilled personnel.
  2) Minimize the disparity in the workload of any two personnel of the pool.

  Subject to the constraint that time deadlines of all the tickets should be met with, and that the assigned personnel are capable of doing the work.
- Part II (Offline analyzer): Simulate the historical workload and dispatch the tickets to the personnel in order to
  1) Minimize the resource cost of the pool by using least number of personnel, as well as minimum cost personnel for completing the work.

  Based on the simulation results, we make recommendations for, and also enable decision making for optimal size and configuration of the pool subject to the constraint that SLAs of all the tickets should be met with.

Real ticket data is used to validate the approach and implementation. We compare the automated dispatcher with the human dispatcher on the basis of mean time to resolve each ticket, load balancing, and potential benefit from any reductions in the personnel costs. In [1], [2], and [3] we observe similar problems being addressed in various different domains.

* Research supported in part by NSF grant CCF-0844796. Work done while author was an intern at IBM Research - India.

978-0-7695-4371-0/11 $26.00 © 2011 Crown Copyright
DOI 10.1109/SRII.2011.54

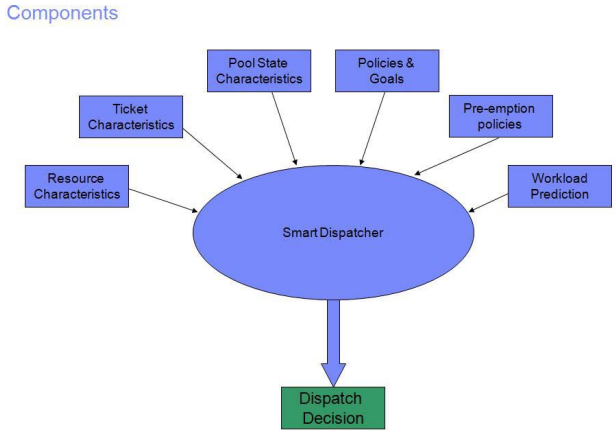

Fig. 1. The Automated Dispatcher Process Overview

We give a brief overview of the automated dispatcher we aim to design and implement in Figure 1. As mentioned earlier, the output from the system is an optimized dispatch decision of tickets to personnel, based on various inputs and constraints.

II. Business Context

This section discusses about the business values of the problem addressed. We define a few terms at this point of time.

Definition 1 (SLO): The SLO for any ticket is the deadline by which it has to be resolved. This is decided by various attributes of a ticket such as severity, priority etc.

Definition 2 (Account-affinity): The account-affinity for any personnel is the set of accounts which that personnel is adept to work on.

Definition 3 (Expected service time): The expected service time for any ticket is the expected time to be taken by any personnel to resolve the ticket. The expected service time is calculated based on historical data and would vary for personnels with different skill levels for the same ticket.

Now we list some of the business values of the problem:

- Dispatching is a critical part of ***service delivery*** and can remove a large part of the wastes in the delivery process.
- Typical pool sizes are ***increasing***, which makes the human dispatchers job difficult. For example, in $U\&I$ service line, the minimum pool size is 50 and going up.
- It is difficult for a human dispatcher to keep track of ***skills, availability, SLO, expected service time*** etc. and optimally dispatch workload.
- It is a stretch for a human dispatcher to figure out the ***optimal size and configuration of the pool***, and the basis of such a recommendation may not always be clear.

III. Approach to modelling

In the problem, the personnel in the pool are divided into 3 sets according to their skill levels:

1) **Rhythm**: The people with the least skills.
2) **Jazz**: The people with the most skills.
3) **Blue**: The people with average skills.

The tickets accordingly are also classifies into Rhythm, Blue, Jazz according to how complex they are based on certain criteria. Now ideally, we would like to have a person of a particular skill level handle a ticket of the same complexity. In our problem, we do not consider the allocation of a ticket of higher complexity to a personnel with a lower skill level.

But we do have to assign a ticket of lower complexity to a personnel with a higher skill level. As for example, a personnel of skill level Blue might be assigned to work on a ticket of complexity Rhythm. We call this the imposition of the *swing rule*. One of the important goals of the dispatcher, is to minimize the number of invocations of such swing rules. We now define the problem.

The dispatch problem has been divided into two subproblems in this work:

1) **Part I (Online phase):** The goal of this part is to dispatch the tickets to the personnel in order to
   - Minimize the invocations of the swing rules while dispatching the tickets.
   - Minimize the difference in the workload of any two personnel of the pool.

   Subject to the constraint that time deadlines of all the tickets should be met with.
2) **Part II (Offline phase):** The goal of this part is analyze and make recommendations for
   - Optimal size and configuration of the pool.

   Subject to the constraint that time deadlines of all the tickets should be met with.

We formulate the problem as a linear program (LP). We use i to index the personnels in the set (P) and j to index the tickets in the set (T).

A. Online dispatching

In this phase, we try to minimize the swings and balance the workload across each skill set of personnel. The optimization is divided into two linear programs. The first one minimizes the usage of the swings:

$$minimize \sum_j \sum_i c_{ij} x_{ij}$$

subject to

$$\sum_i x_{ij} = 1, \qquad \forall j, \tag{1}$$

$$q_{ij} x_{ij} + t_j \leq s_j \qquad ,\forall j. \tag{2}$$

(3)

Now the optimal solution value $\mathbb{S}$ of the above LP is fed into a second LP which balances the workload.

$$minimize\,(\mathbb{U}_1 - \mathbb{U}_2)$$

subject to

$$\sum_i x_{ij} = 1, \qquad \forall j, \tag{4}$$

$$q_{ij}x_{ij} + t_j \leq s_j, \qquad \forall j, \tag{5}$$

$$\left[max\left(\sum_j t_j x_{ij} + q_i + h_i\right)\right]_S \leq \mathbb{U}_1, \qquad \forall i,j,S \in \{R,B,J\} \tag{6}$$

$$\left[min\left(\sum_j t_j x_{ij} + q_i + h_i\right)\right]_S \geq \mathbb{U}_2, \qquad \forall i,j,S \in \{R,B,J\} \tag{7}$$

$$\sum_j \sum_i c_{ij}x_{ij} \leq \mathbb{S}, \forall \qquad i,j. \tag{8}$$

$$\tag{9}$$

1) x_{ij} is a *decision variable* which indicates whether ticket j has been assigned to personnel i.
2) t_j is the estimated time taken to resolve ticket j.
3) c_{ij} is the cost of assigning ticket j to personnel i. The cost is higher if there is a swing rule applied.
4) s_j is the time deadline for ticket j.
5) q_{ij} is the queue length of personnel i with respect to ticket j.
6) q_i and h_i are the sum of estimated times for the open and closed tickets respectively for personnel i.

B. Offline analysis

The motivation behind this analysis is to estimate the optimal number of people the pool could do with still meeting all the time bounds for the tickets.

$$minimize \sum_i f_i y_i$$

subject to

$$\sum_i x_{ij} = 1, \qquad \forall j, \tag{10}$$

$$q_{ij}x_{ij} + t_j \leq s_j, \qquad \forall j, \tag{11}$$

$$max(0, \bar{y}_i) \leq y_i \leq 1, \qquad \forall i, \tag{12}$$

$$x_{ij} \leq y_i, \qquad \forall i,j. \tag{13}$$

$$\tag{14}$$

f_i is the cost of using personnel i. f_i could differ based on the skill level of i e.g. $\{R,B,J\}$ and $\bar{y}_i$ tells whether personnel i was already engaged in the previous invocation of the optimizer.

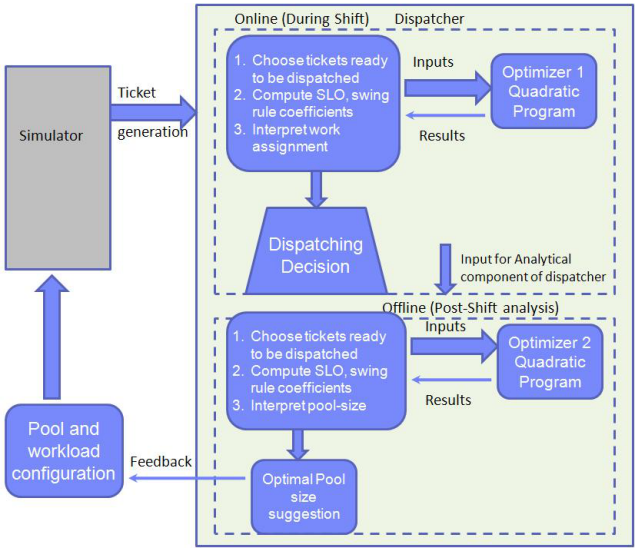

Fig. 2. The Automated Dispatcher Process

IV. Implementation & Potential Experiments

In this section, we discuss the implementation of the dispatcher and also the set of experiments we conducted with it.

A. Implementation

The prototype system has been developed. First, the tickets were generated using a simulator, which was later replaced by an actual input generator to input tickets as they came in. The tickets would be input to the dispatcher in the online phase, which would in turn invoke the optimizer portion. The optimizer which contains the mixed integer program implemented would then produce the dispatching results based on the optimal solution of the MIP.
The offline phase also works in a similar fashion. The output from the optimizer in this case is however an optimal size of the pool of personnel. Figure 2 shows the entire dispatching process of the developed system.

B. Experiments

Dispatching for both the modes have been experimented in two different ways:

1) Input all the tickets over a period for a given shift at a time to the dispatcher.
2) Input the tickets over a period for a given shift at a time to the dispatcher one by one.

1) The input data for tickets had
   - Tickets for the pool for a particular shift over a period of 2 months.
   - 118 tickets classified into Rhythm and Blue.
2) The input data for personnel had
   - 10 personnel of skill level Rhythm.
   - 3 personnel of skill level Blue.

1) The results for the online mode with all the tickets dispatched at once

- Showed a perfect balancing of workload.
- Utilized 11 out of 13 personnel in the pool for that shift.

2) The results for the online mode with the tickets dispatched one by one
   - Showed a near-perfect balancing of workload.
   - Utilized 11 out of 13 personnel in the pool for that shift.

V. Conclusion

The results of the first set of experiments looked promising. The future steps of this ongoing project are listed below.

- Objective setting with Business unit contacts.
- Modeling *business objectives as constraints* and formulating *constrained optimization problem* with the same.
- Currently developing and optimizing the offline and online algorithms, as *standalone programs* with simulation in JAVA, using *CPlex* for optimizing.
- Pilot planned with one $U\&I$ Pool as an assignment recommendation black-box.

References

[1] Hunt Guerney D.H, Goldszmidt German S., King Richard P., Mukherjee Rajat (1998) *Network Dispatcher: a connection router for scalable Internet services.* Computer Networks and ISDN Systems 30 347–357.

[2] Slovis C M, Carruth T B, Seitz W J, Celia T M, Elsea W R (1985) *A Priority Dispatch System for Emergency Medical Services.* Annals of Emergency Medicine, November 1985.

[3] Gillett B E, Miller L R (1974) *A Heuristic Algorithm for the Vehicle-Dispatch Problem.* Operations Research, Vol 22, No. 2 (Mar – Apr., 1974), pp. 340–349.

[4] Eager D L., Lazowska E D., Zahorjan J. (1996) *Adaptive Load Sharing in Homogeneous Distributed Systems.* IEEE Transactions on Software Engineering, Vol. SE-12, No. 5, May 1996.