**Server Load Balancing in WCF**

**Table of Contents**

[Introduction 2](#_Toc466539307)

[Need for Load Balancing 2](#_Toc466539308)

[Load Balancing Set up 3](#_Toc466539309)

[Load Balancer Configuration 5](#_Toc466539310)

[Output 6](#_Toc466539311)

[Summary 7](#_Toc466539312)

# Introduction

Load Balancing - Distribution of work load

Load balancing is an interesting feature in WCF to distribute the workload to the dedicated server or group of servers, which provides a good response time for the client calls. With the help of routing, we can distribute the load and get the response immediately rather than waiting for the response from one server always. Let us discuss this in detail

# Need for Load Balancing

Load Balancing aims to have the following

* Maximizing throughput
* Minimize response time
* Dedicated server for the critical operations
* Avoiding overload
* Optimizing resource

Definitely, you could experience any of the above problems in train ticket booking websites during the festival season when you try to book tickets. Since a large number of users have raised the ticket booking requests, the one server would not able to process all the requests that made the users wait for a long time to get their tickets booked. On the other hand, it might not have a dedicated server for ticket booking calls (Other calls like PNR Status, Food order also utilize the same server and makes the network traffic). When there is an increase in server load, response time to the client will also get increased. In this article, we will see how we can balance the server load in WCF by assigning a dedicated server for the critical client operations like Ticket Booking. We can also load balance the servers in a round-robin method.

# Load Balancing Set up

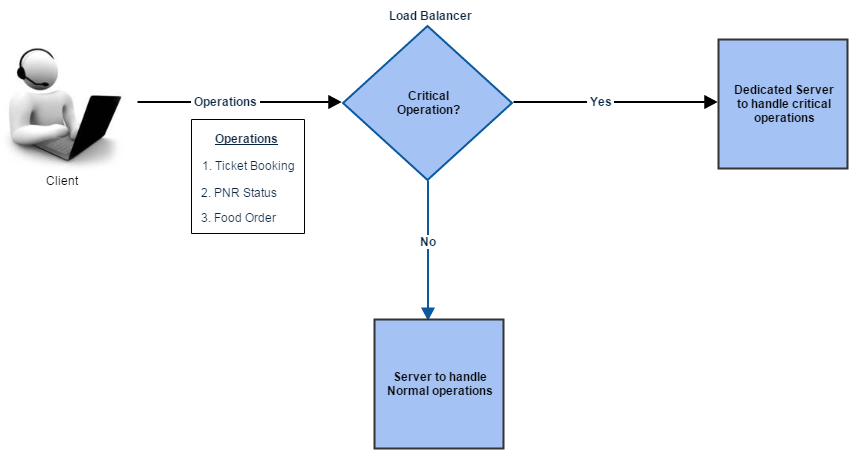
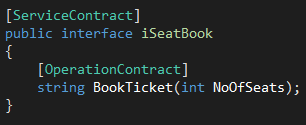
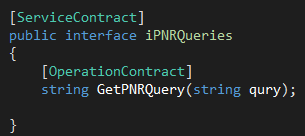


Figure1.0 – Load balancing set up

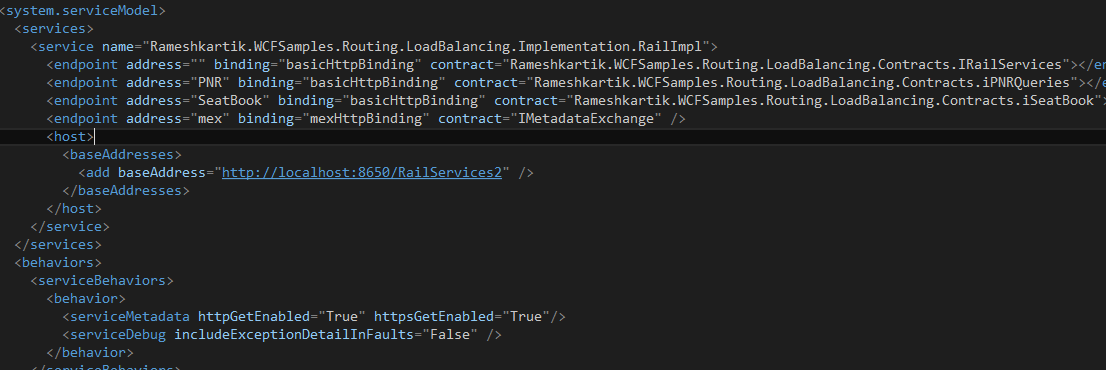
In the snapshot, given above you can find that client will make various operations. If the critical operation like Ticket booking request comes to the Load Balancer Service, it will redirect to the dedicated server assigned for critical operations to maximize the response time. This server will take care of only the ticket bookings; it will not receive any requests like, PNR status or Food order etc. These normal operations will be handled by the other service. Let us see how the same setup will be implemented using WCF.

**Services Configuration**

As a first step, you need to define two services, one to handle the critical operations and the other one to handle the normal operations. Operation Contract or Operations will be same across the 2 servers. Both servers will be capable of handling all the operations. In the snapshot, given below you can find following operations (PNR,TicketBook) will be hosted in a service 1 and service 2



Load balancer who sits between the client and the services will load balance the BookTicket requests to the dedicated server.  Since the BookTicket operation will be getting many requests by a large number of clients during the festive/tatkal season, we have hosted a dedicated server to maximize the response time. Once you implemented the contracts, create the service, endpoint and base address to host the service. Please take the below snapshot as a reference and do the hosting for the services 1 and 2.

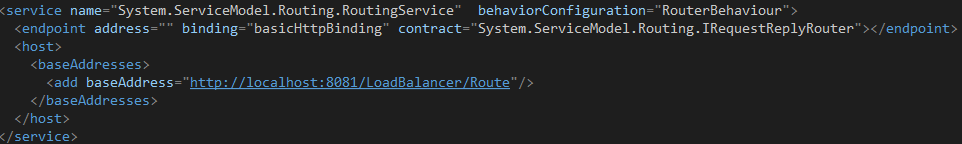


Please refer to my old articles for more details on hosting the services.

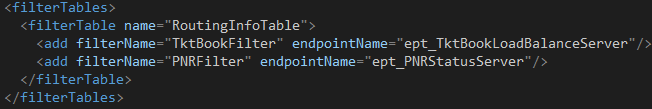
# Load Balancer Configuration

Now let us configure the load-balancing server to move the ticket booking operations on a dedicated server and the PNR operations to the normal server. In our case, service 2 will receive and process the critical operation requests (TicketBook) and service 1 will receive and process the normal operation requests (PNR). Technically, Load Balance server is none other than the Routing server, which redirects the request based on the operation (TicketBook/PNR). Please refer to my old articles for more details on Routing service configuration. As I mentioned in my routing articles please do remember this sentence "Service the FilterTable and FilterEnd", which will take you to configure RoutingServer easily.

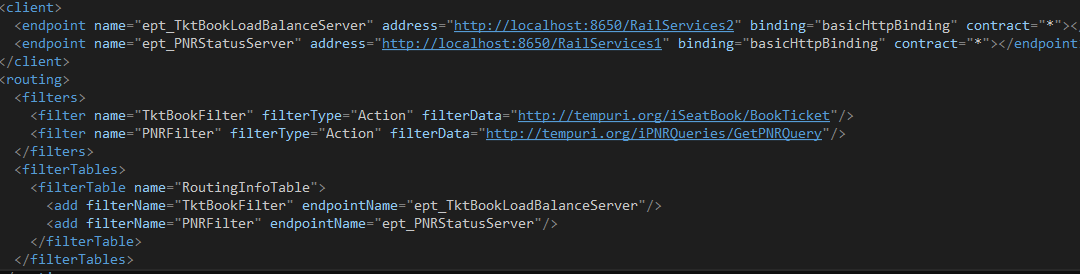
Service = Configure the Routing Service as it mentioned in the snapshot, given below, The name of the Routing service should always be "System.ServiceModel.Routing.RoutingService",and select the address, binding and contract depending upon the requirement.



Next step, Filter Table = Define the Filter table which requires Filter name and the associated endpoint name, means if the filter gets matched, redirect the message to the URL mentioned in the associated end point.



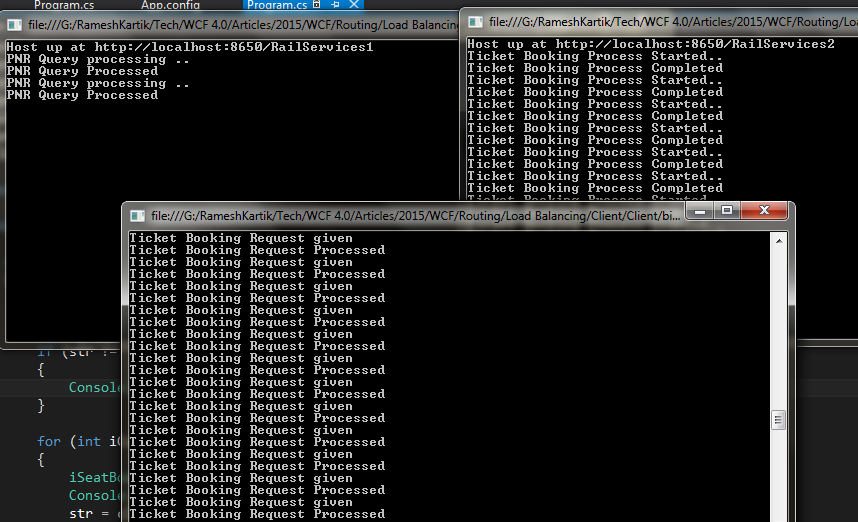
The Final one is FilterEnd, which means we need to define the filter and the end point. When we are defining the filter we must supply the filtertype and filterData, In our case, when the data(iSeatBook/BookTicket) comes to the service ,it should pass the URL mentioned in the associated end point.



In the snapshot, given above you can find that if the incoming message comes for the BookTicket operation(Look at the FilterData element) in Filter, it finds the associated endpoint in the filtertable, In our case, the associated end point of Filtername "TktBookFilter" is "ept\_TktBookLoadBalanceServer". Then it fetches the address from the endpoint element and redirects the message accordingly. In our case, the address to redirect the data (iSeatBook/BookTicket) is http://localhost:8650/RailServices2.  This logic will redirect all the ticket booking operations to the address http://localhost:8650/RailServices2 and redirect all the PNR operations to the address http://localhost:8650/RailServices1. Please download the attached source code and have a look on it.

# Output

Let’s analyze the output screens to get a better understanding



If you clearly notice the first line in the top 2 windows, it clearly mentioned the address where the service is hosted. Left side window in the top side tells that this server hosted at http://localhost:8650/RailServices1 and it processes only the PNR operations, whereas the right window in the top side tells that server hosted at http://localhost:8650/RailServices2 and it processes only the TicketBooking operations. The bottom window is a client window where it sends the requests to the service side. Please download the attached source code for more details on it.

# Summary

Load balancing feature in WCF can be attained with the help of Routing. It load balance (route) the incoming requests to the appropriate server based on the operation.