Ferramenta de monitoramento de geolocalização do sistema BRT do Rio de Janeiro utilizando Apache Spark

Cláudio Santos, Jonathan A. da Silva

Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ
Rio de Janeiro/RJ – Brasil

{csantos,jonathan}@cos.ufrj.br

Resumo. *Utilizando abordagens de Big Data e o* framework *Apache Spark, foi desenvolvida uma ferramenta de coleta, análise e visualização em tempo real de dados de geolocalização dos ônibus do sistema BRT do Rio de Janeiro, através da qual foi analisada a resiliência e escalabilidade do* framework *utilizado. Provou-se seu potencial para operações em grandes volumes de dados, gerados em fluxo contínuo, sobre os quais é necessária análise e agregação, em tempo real e também contínua.*

1. Introdução

O sistema BRT (em inglês, Bus Rapid Transit) é um sistema de transporte feito por ônibus através de corredores exclusivos e, por isso, é uma alternativa mais rápida de viagem para os passageiros. Este modelo de mobilidade existe em mais de cem países. No Rio de Janeiro, cerca de 450 mil pessoas são transportadas por dia através deste sistema, que tem uma frota de 440 ônibus distribuída em três corredores exclusivos (TransOeste, TransCarioca e TransOlímpica). Seu Centro de Controle Operacional (CCO) planeja e controla a operação do modal de forma centralizada graças a informações e comunicação avançadas, como câmeras nas estações e ao longo do corredor, e dispositivos de GPS embarcados em todos os veículos da frota, permitindo o controle de partidas dos ônibus, dos tempos de viagem, bem como gerar indicadores e relatórios do sistema (BRT Rio, 2015).

As informações de posição e velocidade dos veículos da frota do BRT estão disponíveis publicamente na Internet em http://data.rio/dataset/brt-gps, no formato indicado pela tabela 1 abaixo. Através desses dados, é possível executar análises sobre o funcionamento do sistema.

O *dataset*, que é fornecido constantemente pelo consórcio operador do serviço, proporciona o acúmulo de uma grande quantidade de dados em médio e longo prazo; esses dois aspectos, *volume* e *velocidade*, somados ao *valor agregado* da informação e seu histórico, possibilitam uma análise dos dados de acordo com a abordagem de Big Data e tecnologias que têm sido desenvolvidas nos últimos anos, e cada vez mais utilizadas em uma ampla gama de aplicações e áreas de conhecimento, como planejamento, transporte inteligente, economia de energia, entre outras (JAGADISH et al., 2014).

Campo	*Descrição*	*Tipo*	*Tamanho*
Codigo	Identificação alfanumérica do veículo	STRING	7
Linha	Linha do BRT	STRING	7
Latitude	Latitude do ônibus na coleta (GPS, WGS84)	FLOAT	11
Longitude	Longitude do ônibus na coleta (GPS, WGS84)	FLOAT	11
DataHora	Data e hora da coleta do dado	DATETIME	23
Velocidade	Velocidade do ônibus na hora do coleta do dado	FLOAT	6
Sentido	"Ida" ou "Volta"	STRING	5
Trajeto	Locais de origem e destino da linha	STRING	50

Tabela 1. Descrição do dataset *contendo informações sobre os veículos do sistema BRT*

Nesse contexto de uma nova fronteira tecnológica, foi proposta uma nova estrutura de computação distribuída em *clusters*, com suporte a distribuição automática de tarefas e tolerância a falhas, facilitando o cálculo distribuído através de um conjunto rico de operadores. O chamado *Resilient Distributed Dataset* (RDD) fornece uma interface baseada em transformações, aplicando a mesma operação a um grande conjunto de dados distribuídos, suportando tolerância a falhas através do registro do *pipeline* de transformações em vez dos dados reais (ZAHARIA et al, 2012). Essa interface é a estrutura básica do *framework* Apache Spark, através do qual o programador escreve um programa *driver* que é enviado a um cluster composto por máquinas *workers*, conforme a figura 1 a seguir. O programa deve conter a definição de um ou mais RDDs e as ações a serem executadas.

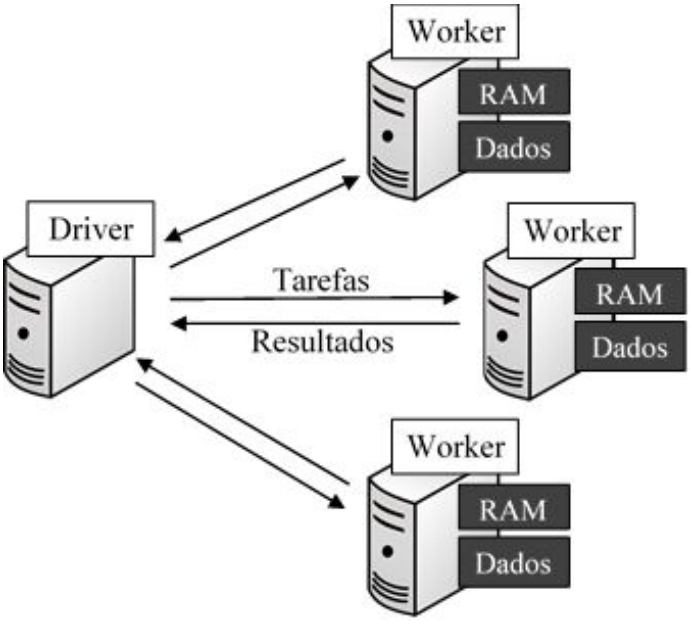

Figura 1. Arquitetura do framework *Apache Spark. Adaptado de ZAHARIA et al., 2012*

2. Objetivo

Numa abordagem de Big Data, o processo de análise é iterativo, envolvendo diversas fases, como aquisição de dados; extração e limpeza da informação; integração, agregação e representação de dados; modelagem e análise; interpretação (JAGADISH et al., 2014). Neste trabalho, será analisado o fluxo contínuo de dados de posição e velocidade dos veículos do sistema BRT, com o objetivo de gerar visualizações em tempo real, bem como agregações e cálculos baseados no histórico acumulado desses dados.

Os dois principais públicos-alvo do trabalho proposto são os próprios usuários do sistema BRT e o Centro de Controle Operacional. Para aqueles, o principal objetivo é fornecer para uma plataforma onde seja possível acompanhar o posicionamento de cada carro do BRT em tempo de atualização do *feed*, evitando assim medições imprecisas ou errôneas de previsão de chegada de um carro – o atual sistema apenas fornece o tempo previsto de chegada em cada estação baseado na posição geográfica do veículo e sua velocidade instantânea, considerando apenas a última atualização do *feed* e, assim, desconsiderando dados históricos de velocidade média no trecho a percorrer até a estação em questão. Com isso, impede que o usuário tenha uma real clareza sobre os fatos para decidir se opta por utilizar o serviço ou procurar outro meio de transporte. Outro principal objetivo é auxiliar o CCO no monitoramento da via. Constantemente algumas vias do BRT sofrem carência de manutenção. Para diminuir o tempo de resposta de possíveis corredores que possam estar sendo afetados com atrasos e aumento do tempo de viagem por problemas na via, foi proposto o uso de dados históricos para o monitoramento da via. Como dito anteriormente, o *feed* de dados do BRT possui informações como a posição do ônibus em latitude e longitude, o código da linha na qual ele pertence, a sua velocidade naquele instante da atualização e o *timestamp* em que essa informação foi gerada. Com essas informações foram construídos quatro gráficos e um mapa de calor. Os gráficos contém informações como velocidade média no período de 24 horas agrupado por hora, velocidade média no período de 7 dias agrupado por dias, número de carros disponíveis no período de 24 horas agrupado por hora e número de carros disponíveis no período de 7 dias agrupados por dia, sendo que todos os gráficos foram gerados para cada corredor: TransOeste, TransOlímpica e TransCarioca. O mapa de calor utiliza a latitude e a longitude de cada carro e a velocidade do carro nos últimos 5 minutos. O objetivo do mapa de calor é analisar de forma mais profunda como está o comportamento da via em determinados trechos.

Esses objetivos serão atingidos com o uso do *framework* Apache Spark enquanto ferramenta de ETL (*Extraction, Transformation & Load*), implementada de maneira eficiente e transparente através da interface *Spark SQL*, que incorpora métodos de carregamento, consulta e persistência de dados estruturados ou semiestruturados usando

consultas estruturadas, expressas em linguagem SQL ou uma API estruturada e declarativa chamada *Structured Query DSL* (LASKOWSKI, 2017a). Uma evolução desta interface desenvolvida recentemente, chamada *Structured Streaming*, provê suporte ao processamento de fluxos de dados contínuos, disponibilizando esta mesma interface declarativa de alto nível de modo a suportar a execução incremental contínua da consulta (LASKOWSKI, 2017b).

3. Solução proposta

O *dataset* é disponibilizado na Internet através de uma URL única (http://webapibrt.rio.rj.gov.br/api/v1/brt) no formato JSON. Assim, de modo a manter um histórico desse conjunto de dados, e também permitir a execução distribuída do *framework* com replicação de dados entre os diferentes *workers* (cf. Figura 1 acima), é utilizado HDFS (Hadoop Distributed File System) como ferramenta de armazenamento, salvando o conjunto de dados a cada 1 minuto e nomeando o arquivo com o *timestamp* correspondente ao dia/horário de *download*.

O programa recebe continuamente os dados armazenados e executa o pré-processamento a seguir:

- Exclusão das linhas sem número ou nome associado, ou não pertencentes ao sistema BRT (como linhas de ônibus alimentadores) (campo *Linha*, cf. Tabela 1)
- Associação das linhas com seu respectivo corredor exclusivo (TransOeste, TransCarioca ou TransOlímpica)
- Descarte de dados duplicados, considerando como código-chave o número de ordem do veículo (campo *Ordem*, id.) e a marca de data/hora correspondente (campo *DataHora*, id.)

Deste modo, é possível gerar os seguintes dados para análise:

- Posição e velocidade dos veículos do sistema, em tempo real
- Média histórica de velocidade dos veículos agrupada por corredor, em períodos de 1 hora e 1 dia (24 horas)
- Contagem do número de veículos agrupada por corredor, em períodos de 1 hora e 1 dia (24 horas)

Os dados gerados pela ferramenta são disponibilizados a um banco de dados através da implementação da interface *ForeachWriter* da biblioteca *Structured Streaming*, que permite executar de maneira distribuída um método personalizado para exportação dos dados. Nesta implementação, cada um dos *workers* executa comandos SQL INSERT para armazenar os dados gerados por sua instância de execução em um banco de dados MySQL para permitir consultas futuras a partir de um *frontend* com interface gráfica.

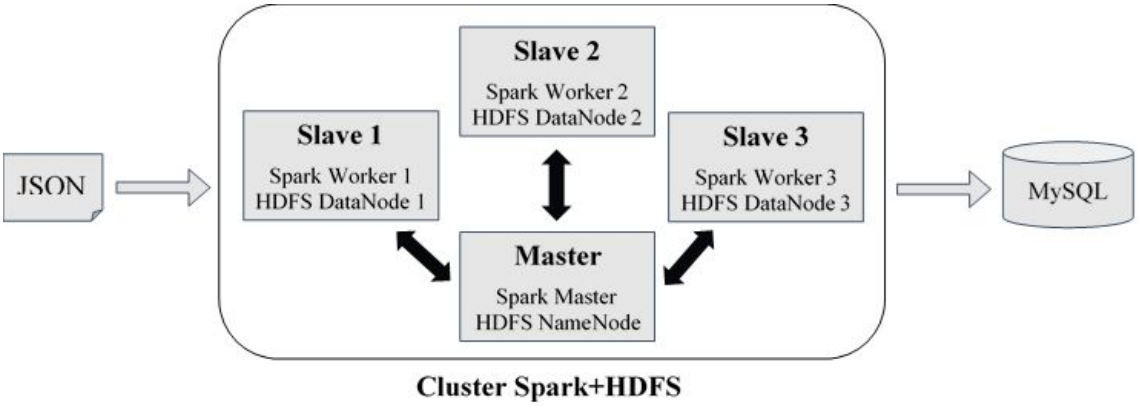

Figura 2. Arquitetura básica da ferramenta ETL desenvolvida

A partir disso, os dados armazenados em servidor MySQL ficam disponíveis para consulta e análise. Para este passo, foi desenvolvida uma aplicação web cuja arquitetura é apresentada na figura 3.

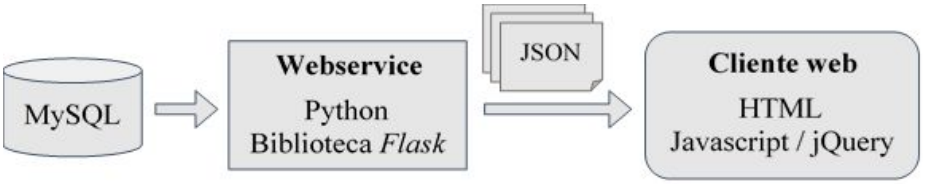

Figura 3. Arquitetura básica da interface de visualização desenvolvida

Na figura 4 é apresentada a aplicação *web* onde o usuário do sistema BRT pode manter-se atualizado do posicionamento do ônibus na qual deseja embarcar para chegar a um determinado destino. Essa aplicação possui uma opção de filtragem pelo corredor desejado ou visualizar todos os ônibus ativos na linha. Há também a possibilidade de visualizar um *heatmap*, ou seja, um mapa de calor na qual a variação de cor é dada pela velocidade do ônibus. Esse mapa de calor pode ser utilizado para identificação de trechos onde os ônibus estão trafegando com uma velocidade muito baixa, possivelmente por irregularidades na via (como tráfego irregular de pedestres), problemas na infraestrutura da via (como má qualidade do asfalto, cruzamentos com alta densidade de tráfego), ou mesmo a ausência de um corredor exclusivo para os veículos do sistema.

4. Resultados obtidos

A figura 4 apresenta o resultado do processamento dos dados em tempo real, essa integração do processamento com a aplicação web gera um sistema com o posicionamento em tempo real dos ônibus do BRT em tempo real, cada corredor tem uma marcação de bandeira diferente que representa o posicionamento do ônibus, assim como cada estação é representado por uma bandeira de cor mesclada, são elas:

simboliza uma estação do sistema BRT

simboliza ônibus do corredor da TransOlímpica

simboliza ônibus do corredor da TransOeste

simboliza ônibus do corredor da TransCarioca

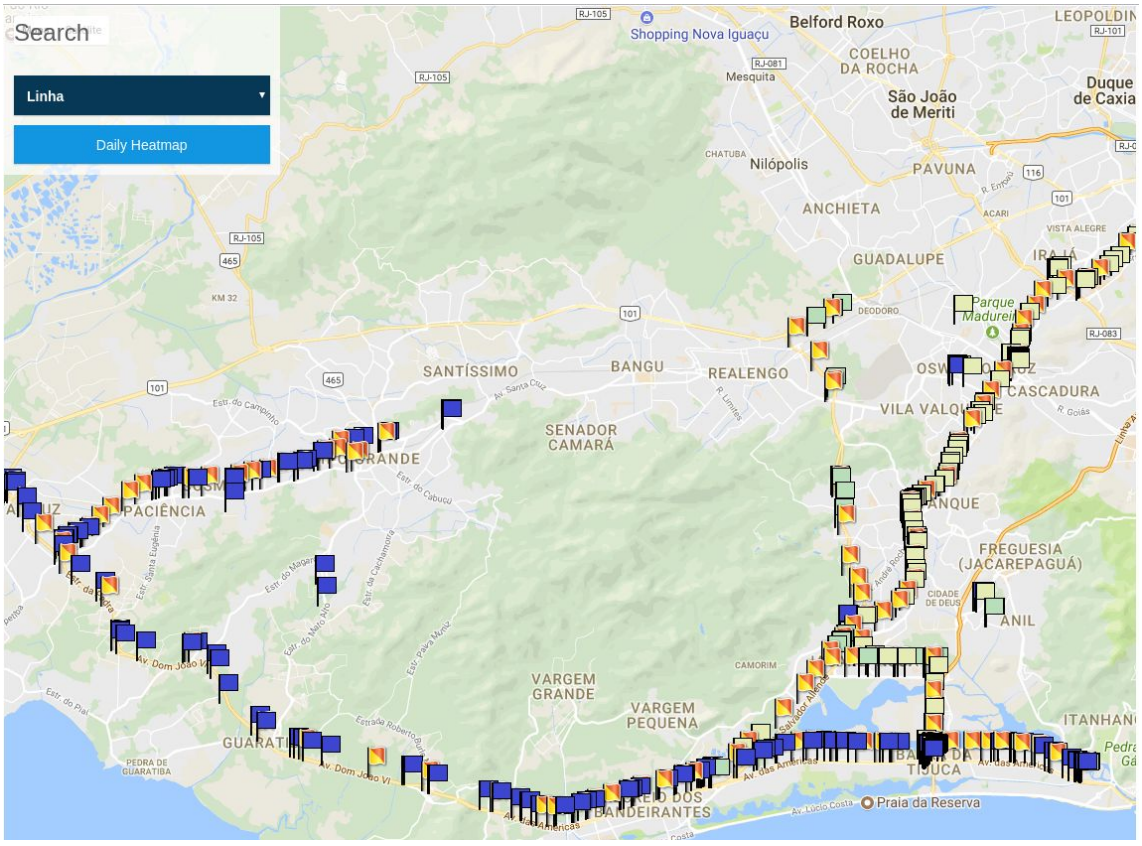

Figura 4. Acompanhamento em tempo real

As figuras 5, 6, 7, 8 e 9 apresentam o mapa de calor gerado pelo sistema. As figuras focalizam lugares onde há um grande número de registros de ocorrências no BRT, seja por problemas ou irregularidades na via, ou mesmo pela ausência de um corredor exclusivo.

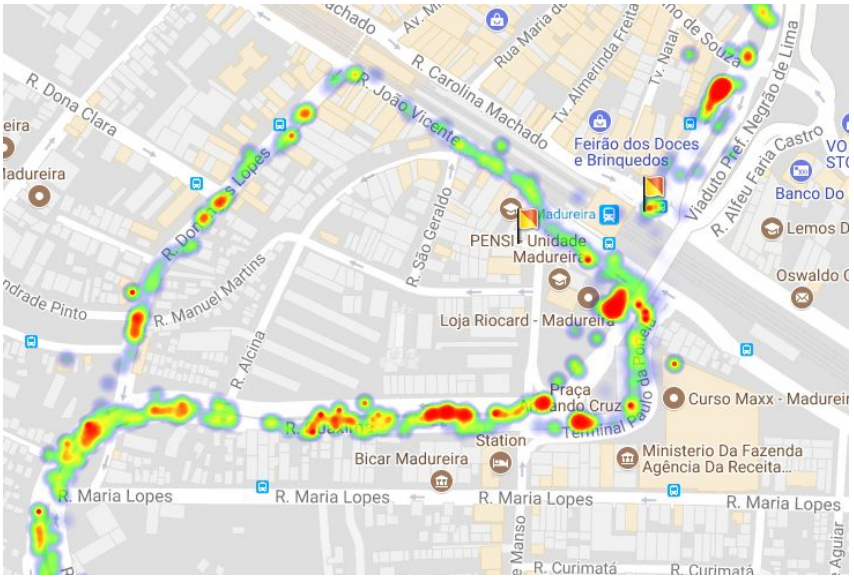

Figura 5. Entorno do terminal Paulo da Portela, onde não há corredor exclusivo

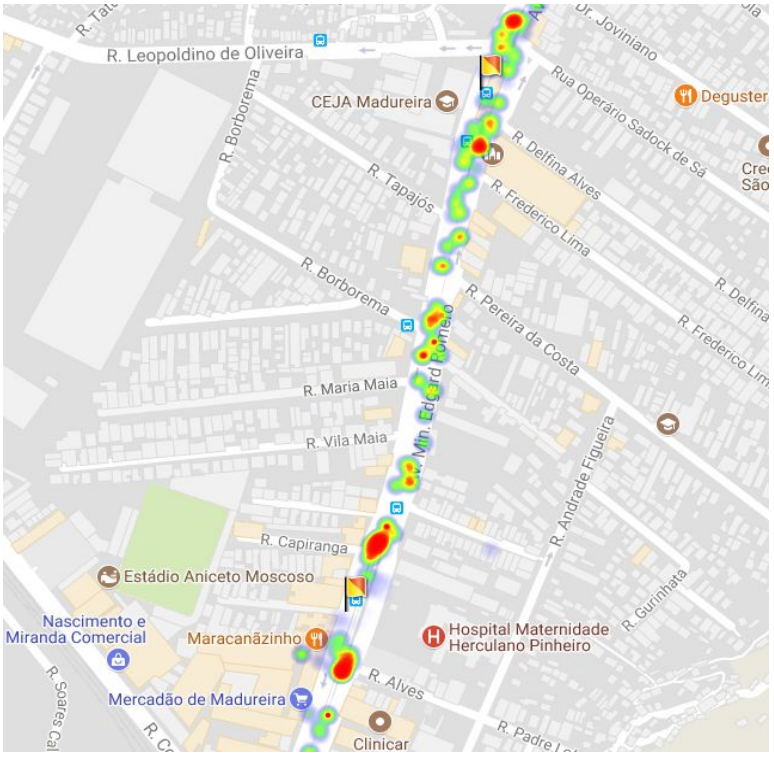

Figura 6. Trecho da Av. Edgard Romero (Madureira), maior trânsito de pessoas na via

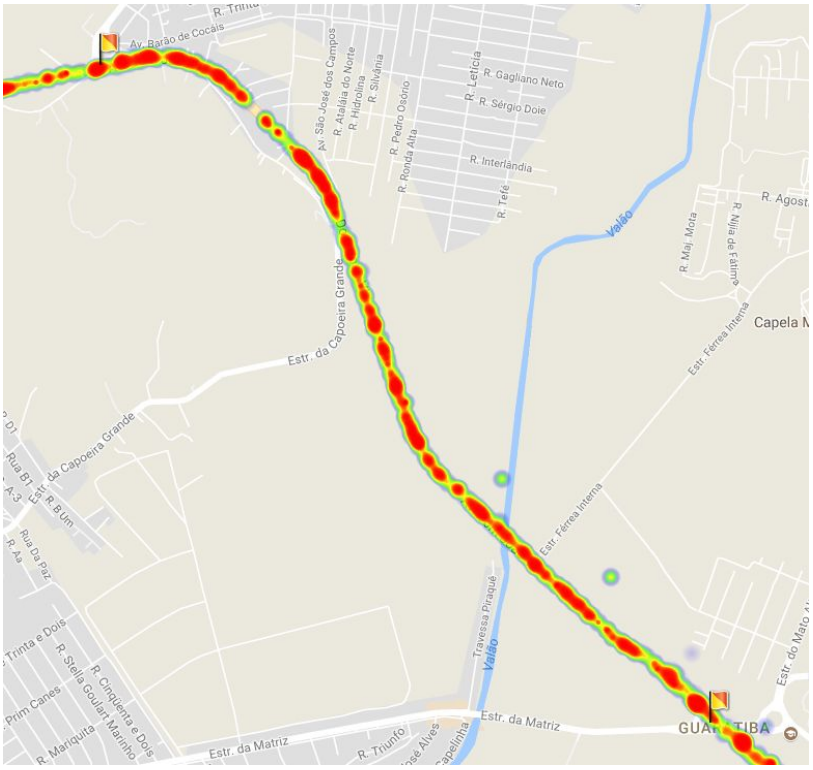

Figura 7. Trecho Mato Alto x Magarça, com más condições do asfalto

Figura 8. Trecho Magarça x Pingo D'Água, com más condições do asfalto

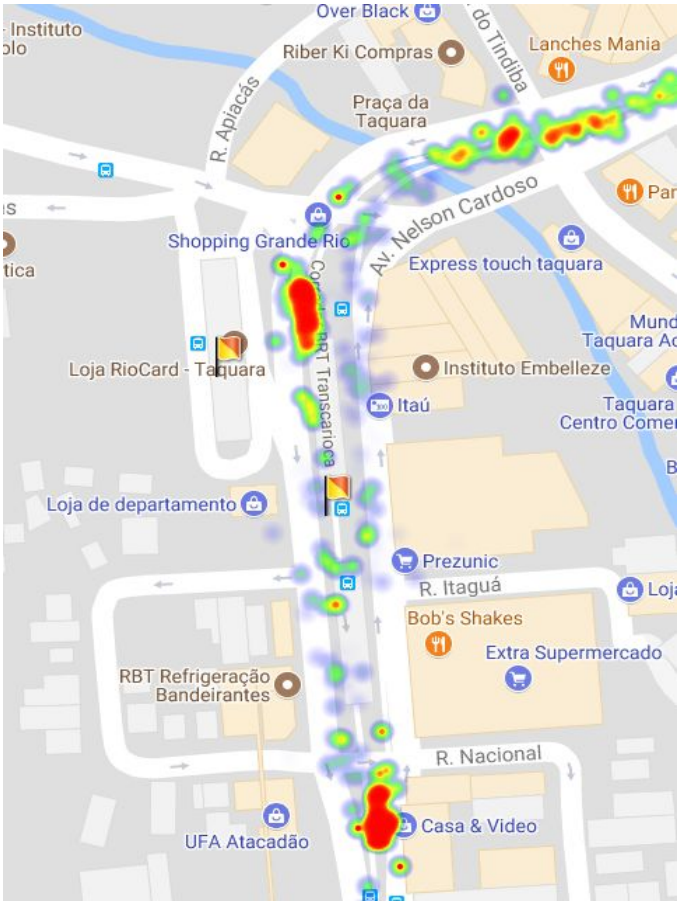

Figura 9. Entorno da estação da Taquara, com alto número de semáforos e cruzamentos

As figuras 10 e 11 apresentam gráficos relacionados a velocidade média e disponibilidade de ônibus em um período de 7 dias dividido por cada dia desse período, respectivamente. Esses gráfico são utilizados para análise da velocidade média do dia com a quantidade de ônibus disponível na via.

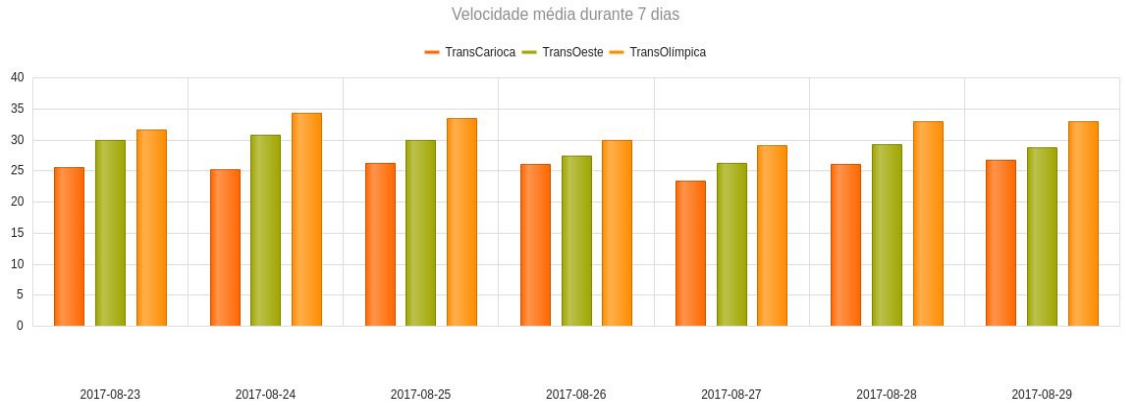

Figura 10. Gráfico de velocidade média durante 7 dias, agrupado por corredor

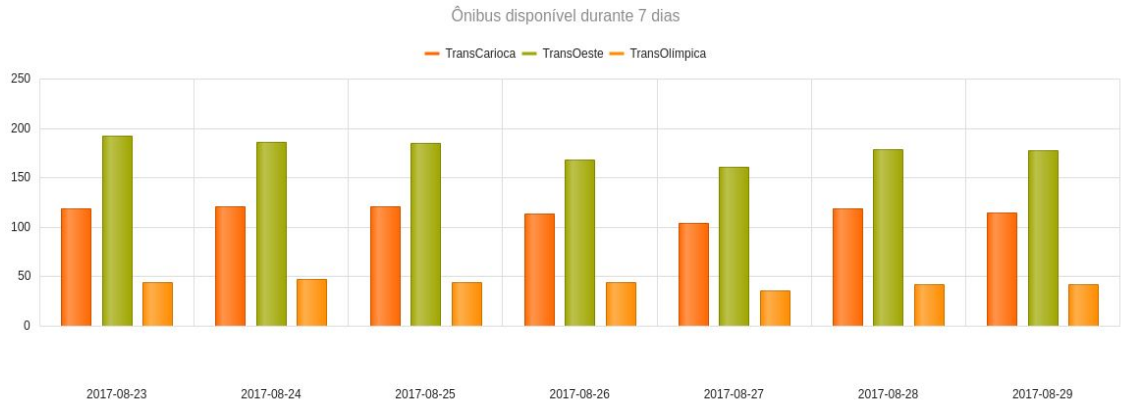

Figura 11. Gráfico Ônibus disponível durante 7 dias, agrupado por corredor

Já nas figuras 12 e 13 são apresentados gráficos relacionados a velocidade média e disponibilidade de ônibus em um período de 24 horas, respectivamente. A partir dos gráficos é possível analisar a variação da velocidade média e da disponibilidade de veículos no sistema ao longo do dia.

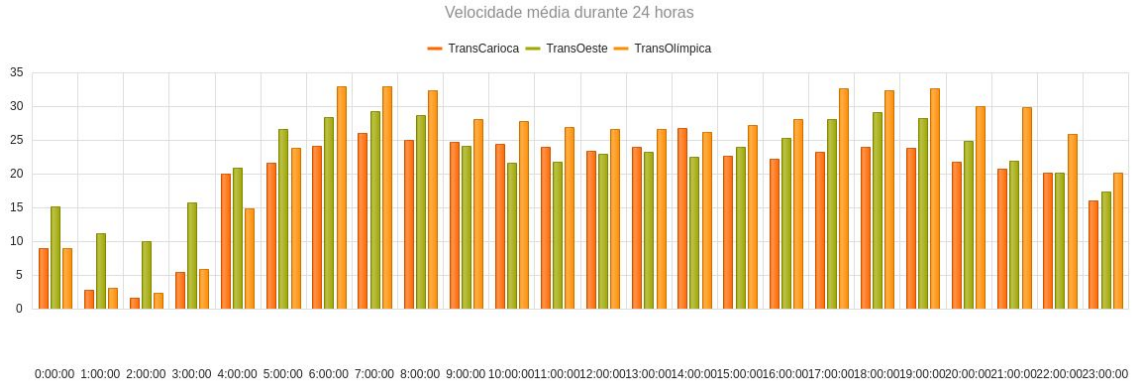

Figura 12. Gráfico Velocidade média durante 24 horas

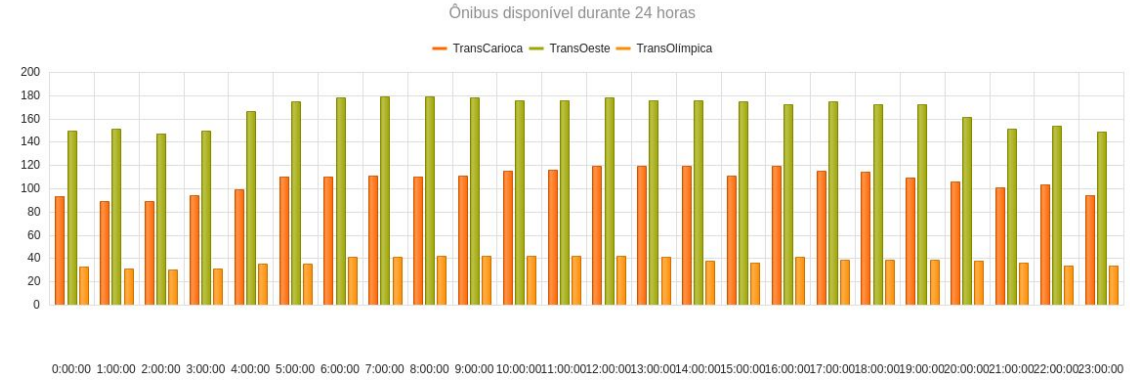

Figura 13. Gráfico Ônibus disponível durante 24 horas

5. Conclusão

A partir da análise dos dados históricos de velocidade dos veículos, foi possível observar trechos em que a velocidade média é consideravelmente menor, seja devido às condições atuais da via ou devido à inexistência de uma faixa exclusiva para circulação do sistema BRT.

Como pode ser notado nos gráficos 10, 11, 12 e 13, que foram gerados para efetuar a correlação entre a velocidade média e a disponibilidade de ônibus da via, pode-se notar que os horários em que a velocidade média é menor também há menor disponibilidade de carros (período da madrugada) e, portanto, o intervalo entre os ônibus é maior, embora não haja uma grande variação da disponibilidade de carros durante o dia.

O uso do *framework* Apache Spark possibilitou o desenvolvimento desta ferramenta de maneira ágil e eficiente, implementando extração, transformação e agregação de dados estruturados e consultas SQL utilizando paralelismo de operações e dados de maneira transparente.

Referências

Consórcio BRT Rio. **Centro de Controle Operacional celebra um ano de funcionamento.** 2015. www.brtrio.com/noticia/cco-centro-de-controle-operacional-faz-um-ano. Acessado em 01/09/2017.

Jagadish, H. V.; Gehrke, Johannes; Labrinidis, Alexandros; Papakonstantinou, Yannis; Patel, Jignesh M.; Ramakrishnan, Raghu; Shahabi, Cyrus. **Big data and its technical challenges.** Commun. ACM 57, 7 (July 2014), 86-94. 2014.

Laskowski, Jacek. **Mastering Apache Spark 2**. GitBook. 2017 (a). http://www.gitbook.com/book/jaceklaskowski/mastering-apache-spark/. Acessado em 15/08/2017.

Laskowski, Jacek. **Spark Structured Streaming (Apache Spark 2.2+)**. GitBook. 2017 (b). http://www.gitbook.com/book/jaceklaskowski/spark-structured-streaming. Acessado em 15/08/2017.

Zaharia, Matei; Chowdhury, Mosharaf; Das, Tathagata; Dave, Ankur; Ma, Justin; McCauley, Murphy**;** Franklin, Michael J.; Shenker, Scott; Stoica, Ion. **Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing.** Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012.