International Journal of Distributed Systems and Technologies
Volume 7 • Issue 1 • January-March 2016

Streaming Remote Sensing Data Processing for the Future Smart Cities:
State of the Art and Future Challenges

Xihuang Sun, Institute of Remote Sensing and Digital Earth, Chinese Academy of Science, Beijing, China
Peng Liu, Institute of Remote Sensing and Digital Earth, Chinese Academy of Science, Beijing, China
Yan Ma, Institute of Remote Sensing and Digital Earth, Chinese Academy of Science, Beijing, China
Dingsheng Liu, Institute of Remote Sensing and Digital Earth, Chinese Academy of Science, Beijing, China
Yechao Sun, China Centre for Resources Satellite Data and Application, Beijing, China

ABSTRACT

The explosion of data and the increase in processing complexity, together with the increasing needs of real-time processing and concurrent data access, make remote sensing data streaming processing a wide research area to study. This paper introduces current situation of remote sensing data processing and how timely remote sensing data processing can help build future smart cities. Current research on remote sensing data streaming is also introduced where the three typical and open-source stream processing frameworks are introduced. This paper also discusses some design concerns for remote sensing data streaming processing systems, such as data model and transmission, system model, programming interfaces, storage management, availability, etc. Finally, this research specifically addresses some of the challenges of remote sensing data streaming processing, such as scalability, fault tolerance, consistency, load balancing and throughput.

KEYWORDS

Data Processing System, Data Streaming, Real-time Processing, Remote Sensing Processing, Smart Cities

1. INTRODUCTION

With the fast development of satellite technology, satellite reception capability and the decrease in the costs of computer memory, remote sensing (RS) data processing has grown rapidly. The RS data transmitted in Institute of Remote Sensing and Digital Earth (RADI) reached 412.97GB/day in 2012 (RADI, 2012) and continues to grow rapidly. Furthermore, the RS data in EOSDIS grew to 8.5TBper day in 2012 and its total archived dataset reached 10PB (EOSDIS, 2012).

In addition to the data explosion, the needs of real-time processing and concurrent access of RS data processing has also increased as more time-strict applications are applied.RS data processing plays an important role in disaster monitoring, climate prediction, and many other city-related applications. Real-time processing of RS data can encourage the development of smart disaster management, pre-disaster risk assessment, and post-disaster risk assessment big data applications with more effective solutions and fewer energy costs. Increasing the speed of RS data processing of massive data and real-time analysis will benefit a lot in building future smart cities.

RS data are the earth-observing data continuously obtained from space borne and airborne sensors and other data acquisition measurements. As RS data is mainly transmitted as data streams

DOI: 10.4018/IJDST.2016010101

from sensors but stored as data files for offline processing, research based on data files cannot achieve thorough real-time processing since they cannot fill the gap between data transmission and data processing. In fact, problems such as mixed programming across different hardware, IO bottlenecks caused by external disk storage, and coordination between distributed datacenters still cannot be fixed perfectly. Streaming RS data processing treats RS data as continuous data streams and completes data transmission and computing in memory. It may reduce the processing units by eliminating the disk storage of intermediate data and build a pipeline where sub part of processing can make interactions overlap in time. RS data streaming processing brings no extra burden to processing hardware and can break the parallel IO bottleneck with memory.

In this paper we introduce current research in RS data processing. Then we summarize mainstream data streaming processing frameworks. We also discuss aspects of building RS data streaming processing frameworks. Finally we discuss the challenges of future RS data streaming processing, such as scalability, fault tolerance, consistency, load balancing and throughput.

2. RELATED WORKS

2.1. Current Works in RS Data Processing

To boost the speed of RS data processing, many research have been studied based on different architectures. These research mainly focus on offline RS data processing. Algorithms and processing patterns (Hailiang & LU, 2010; Jun-jun & Gang, 2010; Ke-fei, 2010; LIU et al., 2014) take the advantage of different hardware architectures such as GPU and FPGA. High performance clusters (Wei, Liu, & Wang, 2014; Xue-ping, 2013) enhance the RS data processing performance by using parallel computing. Distributed computing environments (Sekiguchi et al., 2008) can increase the data-storage capacity and expand the diversity of processing services.

There are also works based on cloud computing and RS big data computing. G-Hadoop (Wang et al., 2013) applied MapReduce across distributed data centers. Almeer(2012) employed Hadoop MapReduce framework to implement parallel processing of RS images and built an experimental 112-core high-performance cloud computing system. Wang (2014) estimates the statistical characteristics of RS big data in the wavelet transform domain. Task scheduling strategies (W. Zhang, Wang, Ma, & Liu, 2014) across data centers were also studied. Wang (2015) discusses some research issues about processing distributed internet of things data in clouds.

Apart from these works, Research on RS data streaming processing mainly focuses on data model and data querying. StreamInsight (Kazemitabar, Demiryurek, Ali, Akdogan, & Shahabi, 2010) support RS data querying by adding the Microsoft SQL server spatial library. Deng (2015) discusses the parallel processing of dynamic continuous queries over streaming data flows. Although system design is one of the most important areas in data streaming processing, there is still no mature data streaming processing system designed for RS data processing.

2.2. Data Streaming Processing Frameworks

There are several data streaming processing frameworks designed by researchers at IT companies and institutes, such as Yahoo!'s S4(simple scalable streaming system)(Neumeyer, Robbins, & Kesari, 2010), Storm(Toshniwal et al., 2014) from Twitter, Data Freeway and Puma(Zheng, 2011) from Facebook, Samza(Smaza, 2014)from LinkedIn, Microsoft's Time Stream(Qian et al., 2013) system, IBM's Stream Base(StreamBase, 2006), Berkeley's Spark(Zaharia, Chowdhury, Franklin, Shenker, & Stoica, 2010), Google's Mill Wheel(Akidau et al., 2013), and Esper(Esper, 2013) for complex event processing.

We briefly introduce three mainstream data streaming processing frameworks: Storm, Storm and Samza in this sub-section and discuss their design concerns and challenges in detail in follow sections.

Related Content