Analysis and Testing of the Underlying Communication Architecture of ZeroMQ-Based Observation Control Systems (Postprint)

Authors: Qian Jin, Deng Hui, Mei Ying, Shi Congming, Wei Shoulin, Dai Wei, Wang Feng

Date: 2018-12-04T00:00:00+00:00

Abstract

Observation Control System (OCS) is currently a research hotspot, with underlying communication being a critical component of the OCS architecture. However, traditional OCS implementations generally employ raw socket technology, lacking a unified transmission control mechanism; delays frequently occur during intensive data communication, affecting the need for real-time control. Additionally, the absence of broadcast and multicast mechanisms limits OCS system design. This paper addresses OCS requirements by systematically studying and analyzing different communication models based on ZeroMQ and their corresponding astronomical control patterns, and evaluating the availability of various communication control architectures. Building upon this foundation, experiments are conducted to test the corresponding communication models, verifying their usability in distributed control of astronomical instruments. The test results further demonstrate that constructing the underlying communication architecture of OCS using ZeroMQ is feasible and capable of meeting the diverse control requirements of OCS for equipment.

Full Text

Analysis and Test of Underlying Communication Architecture of Observation Control System Based on ZeroMQ

Qian Jin[1], Deng Hui[1,2]*, Mei Ying[2,3], Shi Congming[1], Wei Shoulin[1], Dai Wei[1], Wang Feng[1,2,3]

[1]Computer Technology Application Key Lab of Yunnan Province, Kunming University of Science and Technology, Kunming 650500, China
[2]Center for Astrophysics, Guangzhou University, Guangzhou 510006, China
[3]Yunnan Observatories, Chinese Academy of Sciences, Kunming 650011, China

Abstract: Underlying communication constitutes a critical component of Observation Control System (OCS) architecture, which has emerged as a prominent research focus in recent years. However, traditional OCS implementations typically rely on bare socket technology, lacking a unified transmission control mechanism that frequently introduces delays during intensive data communication, thereby affecting real-time control requirements. Additionally, the absence of broadcast and multicast mechanisms constrains OCS design flexibility. This paper systematically investigates and analyzes various ZeroMQ communication models and their corresponding astronomical control modes based on OCS requirements, evaluating the availability of different communication control architectures. Through experimental testing of these communication models, we verify their applicability in the distributed control of astronomical instruments. The results demonstrate that constructing OCS underlying communication architecture with ZeroMQ is feasible and capable of meeting diverse device control requirements in OCS.

Keywords: Observation Control System; ZeroMQ; Communication Architecture

1 Introduction

With continuous advancements in astronomical observation technology and increasingly sophisticated observational equipment capabilities, control requirements have grown substantially, rendering traditional manual observation modes progressively impractical and highlighting the urgent need for Observation Control Systems (OCS) [1]. Drawing upon experiences from automated and intelligent construction of advanced telescopes abroad, developing an efficient and scalable OCS holds significant importance for next-generation astronomical telescope development.

Astronomical telescopes typically comprise multiple subsystems, including domes, equatorial mounts (frames), spectrometers, and CCD terminals. OCS must coordinate control across these diverse device types while maintaining communication among them. Throughout observation processes, issues such as sequential control and exception recovery must be addressed, making a robust underlying communication architecture fundamental to telescope control system design. Current OCS implementations, whether classical ACS or the open-source RTS2 [2-3], adopt distributed control principles. Particularly in widely-used RTS2, native Socket [4] programming requires each client to maintain simultaneous connections with multiple terminals, resulting in a complex mesh underlying communication structure that complicates system development and implementation. Furthermore, OCS frequently requires synchronous control of multiple devices during actual observations, such as combined CCD observations and synchronized exposures [5]. RTS2 can only send commands to multiple devices sequentially through looping, which introduces significant communication delays and risks message loss. Consequently, developing OCS based on bare socket technology proves overly complex and

不利于 system design and maintenance. Alternative OCS implementations employing HTTP, CORBA, DCOM, and related protocols exhibit limitations at different levels [6], failing to fully satisfy the diverse communication model requirements of telescope control system underlying communication.

In recent years, message middleware technologies have continuously emerged, particularly ZeroMQ [7], which provides robust technical support for collaborative control in distributed environments. More importantly, ZeroMQ supports multiple communication patterns and can flexibly combine basic patterns to address different control scenarios, enabling one-to-one (1:1) and one-to-many (1:N) communication that meets various communication model requirements in OCS. Whether such new technologies as ZeroMQ can be adopted in astronomical observation control system design and implementation has become a common concern.

2.1 Communication Pattern Analysis

ZeroMQ features two important communication patterns: request-reply and publish-subscribe [8]. Combined with the requirements of astronomical observation control systems, these patterns can be flexibly applied to meet astronomical equipment control needs. The specific analysis is as follows.

2.1.1 Request-Reply Pattern

Point-to-point control represents a link-based connection mode where a single message sender directly connects to a single receiver, establishing communication through a question-and-answer mechanism. This pattern finds extensive application in actual observations, such as dome (DOME) switch control and equatorial mount operations. Request-reply is a direct point-to-point communication mode where message senders and receivers employ (Request-Reply) socket pairs to implement message transmission through question-and-answer exchanges. Its main characteristics include: 1) Connection-oriented communication, where the sender must establish direct communication with the receiver before subsequent message transmission can occur; 2) Tight coupling between sender and receiver, requiring both parties to be simultaneously operational (temporal tight coupling) and to know each other’ s address information in advance (spatial tight coupling). The pattern’ s primary advantage lies in its simple implementation and reliable transmission, while its disadvantage is the substantial limitation on system communication flexibility—whenever the receiver changes, the sender’ s application must be modified accordingly [9].

2.1.2 Publish-Subscribe Pattern

Point-to-multipoint control represents a broadcast or multicast communication mode where a message sender establishes communication with multiple receivers, enabling simultaneous control of multiple receivers by a single sender. ZeroMQ’ s publish-subscribe pattern is a one-to-many message broadcasting mode suitable for point-to-multipoint control scenarios in telescope observations, such as simultaneous multi-CCD exposures or combined observations.

This pattern employs (Publish-Subscribe) socket pairs, where the PUB endpoint only needs to encapsulate messages into topics for publication, and SUB endpoints retrieve published content based on subscribed topics. Its main characteristics include: 1) Communication between publisher and subscriber occurs through common topics without requiring direct link establishment; 2) Loose coupling between publisher and subscriber, as neither party needs to know the other' s address (spatial independence), both need not be simultaneously operational (temporal independence), and message production and consumption occur without blocking (process independence); 3) Support for point-to-point, point-to-multipoint, and multipoint-to-multipoint communication. The pattern' s primary advantage is achieving loose coupling between publishers and subscribers, while its fatal flaw lies in the publisher' s unidirectional data distribution without concern for whether all information reaches subscribers, making message loss likely.

2.2 Reliability Analysis

In complex and variable astronomical observation control environments, the status and behavior of senders and receivers are unpredictable. During OCS operation, network interruptions or software/hardware failures may cause message loss, directly impacting astronomical observations. To enhance OCS message transmission reliability, safeguards must be implemented from both sender and receiver perspectives to ensure smooth operation after failures.

Our team conducted extensive experiments on reliability issues and determined that to prevent single-point failures at message endpoints, the socket design should adopt the ZeroMQ dual-star pattern [10]. At any given moment, one node acts as the primary server, receiving all client and device connection requests, while another node serves as a backup. The two nodes maintain mutual heartbeat monitoring, with the primary server periodically sending device connection information to the backup via PUB sockets. When the primary server disappears from the network, the backup immediately assumes its position. Device terminals constitute core components in telescope systems for receiving messages. When devices experience communication anomalies or delays exceeding maximum thresholds, they first employ ZeroMQ timeout retransmission with sequential polling to reassess each device' s online status. To prevent message loss, devices must implement breakpoint resumption functionality [11] after brief recovery periods. Breakpoint resumption enables devices to continue transmission tasks after short-duration connection interruptions, ensuring message integrity [12]. This functionality primarily utilizes the ZeroMQ clone pattern, with PUB-SUB sockets as the core message pattern. When the system receives device feedback indicating connection anomalies, it employs ZeroMQ multi-frame message classes to timestamp and temporarily store messages. Upon recovery, devices immediately obtain current status via REQ sockets and retrieve all messages prior to the status timestamp, ensuring reliable message transmission after fault recovery.

2.3 Efficiency Analysis

Bandwidth resources always present potential challenges in network communication, and achieving reliable communication under low-bandwidth conditions represents a likely scenario. In OCS design, message transmission overhead and latency directly affect overall system performance, with message length being a crucial factor—particularly for observation-generated image data characterized by real-time requirements and large volumes. In this regard, ZeroMQ supports real-time data compression technology employing the LZ4 lossless compression algorithm, which compresses data at high speed before transmission and decompresses it at the receiving end, maximizing lossless message lightweighting and reducing bandwidth requirements—highly beneficial for astronomical observations.

Simultaneously, an efficient message communication pattern serves as important assurance for improving transmission efficiency. Image data achieves efficient point-to-point transmission through ZeroMQ request-reply sockets, while control commands can effectively control multiple devices via broadcast sockets on separate ports. Multi-threading technology at nodes enhances message transmission and processing concurrency, reducing transmission delays caused by sequentially forwarding control commands to different devices point-to-point. These methods effectively ensure the efficiency of OCS underlying communication message transmission.

3.1 Basic Architecture of OCS

[Figure 1: see original paper] illustrates the OCS architecture adopted by our team in developing a new-generation observation control system. OCS resides at the top control level, comprising four components: scheduling subsystem, management subsystem, user interaction subsystem, and status recording system. OCS interacts with other subsystems through interface calls, commanding the next-level terminal devices via telescope control system (TCS), instrument control system (ICS), and real-time data processing system (DHS). Evidently, underlying communication availability and reliability constitute key success factors for implementing a reliable OCS.

From the underlying communication pattern perspective, point-to-point and point-to-multipoint patterns already satisfy OCS design requirements. However, the critical issues become whether these patterns' reliability, transmission latency, and multi-task transmission performance can meet actual communication requirements.

Experimental Environment

The experimental environment utilized five servers with identical configurations, each equipped with gigabit network cards. One server served as the sender, while the others functioned as receivers. The primary server configuration is shown in Table 1 , with all subsequent tests conducted in this environment.

Table 1 Experimental Hardware Environment - OS: CentOS Linux release 7.3.1611 - CPU: Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz - Network: 1.0 Gbps - ZeroMQ Version: (version not specified in original text)

3.2.2 Latency Test

Communication latency for device command reception represents a crucial metric for system communication performance. Therefore, we conducted comparative tests on ZeroMQ communication latency for transmitting messages of different lengths in the same environment. Test 1 simulated a point-to-point control scenario in OCS, with the sender running on one server and the receiver on another. The test method employed REQ-REP sockets to send messages of specific sizes from one server to another. We tested 10,000 control messages of 256 bytes and 1024 bytes respectively, comparing the time difference between sending and receiving each message. The test results are shown in Figure 2 [Figure 2: see original paper]. The results indicate higher transmission delays during initial connection establishment, followed by stabilization with occasional peak values, all within reasonable ranges. When data size increased from 256 bytes to 1024 bytes, latency increased correspondingly, but overall communication latency remained low and suitable for OCS control of devices like domes or telescopes in point-to-point scenarios.

3.2.3 Multi-point Synchronization Control Performance Test

This test simulated a point-to-multipoint control scenario in OCS, with the sender running on one server and receivers on five other servers. The test method utilized PUB-SUB sockets on the publisher to send 10,000 control messages of 256 bytes, with five subscribers receiving messages based on topics. We tested the latency of message reception at each of the five subscriber endpoints, with the underlying communication latency results shown in Figure 3 [Figure 3: see original paper]. Since each message' s communication latency represents discrete data, we analyzed device message reception synchronization from both central tendency and dispersion perspectives to better understand reception performance. As shown in Table 2 , Sub1-Sub5 represent five device endpoints. Comparative analysis reveals that average message reception latency for all devices remains below 10 s with equivalent delay fluctuation levels, where average latency is significantly smaller than standard deviation.

Experimental results indicate that during high-level OCS development and design, control instruction transmission frequency must be limited with reference to underlying communication latency data to further improve control synchronization precision. In actual control scenarios, ensuring control design within the time range of average latency plus three standard deviations meets control precision requirements and sufficiently satisfies OCS point-to-multipoint control scenarios for multiple CCDs and other devices.

Table 2 Experimental Data Analysis

Index	Average Delay (s)	Standard Deviation (s)
Sub1	(value not specified in original)	(value not specified)
Sub2	(value not specified)	(value not specified)
Sub3	(value not specified)	(value not specified)
Sub4	(value not specified)	(value not specified)
Sub5	(value not specified)	(value not specified)

3.2.3 Multi-channel Transmission Availability Test

In astronomical observations, OCS frequently interacts with real-time data processing systems (DHS), where DHS feeds observation data processing results back to OCS for scheduling decisions, and OCS transmits control commands at high speed to DHS or other subsystems. In RTS2, this is typically implemented using multiple sockets.

We designed an experiment to further investigate OCS availability for point-to-point and point-to-multipoint control in intensive data communication environments. The sender generated image data of 4MB per image, converted it to binary data streams, encapsulated it as messages, and transmitted 400MB of data to the receiver via request-reply mode, simulating intensive communication environments for large-volume image data transmission between OCS and DHS.

The test method comprised: 1) Simultaneously running with image transmission, establishing new communication ports using REQ-REP and PUB-SUB socket pairs to continuously send control message packets of 256 bytes from one server to others; 2) Recording image transmission time upon completion of 400MB data transfer and immediately stopping the control message transmission program; 3) Recording control message communication latency, sent message count, and received message count during image transmission. Test results are shown in Table 3 .

Table 3 Multi-channel Transmission Test Data

Metric	REQ-REP	PUB-SUB
Image Transmission Time	7.3s	7.3s
Transmission Rate	55Mb/s	55Mb/s
Sent Messages	(count not specified)	(count not specified)
Received Messages	(count not specified)	(count not specified)
Average Control Message Latency	267.0 s	62.1 s

The results show that 400MB of image data can be transmitted within 7.3 seconds, with a 55Mb/s transmission rate sufficient for high-speed transmission of large-volume image data. Simultaneously, using REQ-REP and PUB-SUB socket pairs in ZeroMQ to achieve real-time concurrent transmission of control commands demonstrates that both patterns can transmit messages without packet loss, ensuring reliable control command transmission. Compared with standalone control command transmission scenarios, both patterns are affected by network bandwidth and communication latency under intensive data communication, resulting in reduced overall transmission rates. However, transmission performance remains more efficient and stable than traditional Socket communication, sufficiently meeting real-time and stable control command transmission requirements. Repeated testing confirms that implementing multi-task, multichannel transmission with ZeroMQ is feasible, with data transmission methods, rates, and stability all meeting OCS underlying communication requirements. This approach not only enables data transmission between OCS and DHS but also facilitates information exchange among terminals and interconnection control among subsystems, addressing the critical issue of independent, uncoordinated operation among current astronomical telescope terminals and subsystems.

4 Discussion and Conclusion

This paper has discussed the advantages and disadvantages of existing ZeroMQ communication patterns, analyzed primary astronomical control modes in OCS, and proposed different communication control architectures based on ZeroMQ, providing reference methods for addressing communication issues such as latency and exception recovery. Test result analysis demonstrates that constructing OCS underlying communication architecture using ZeroMQ technology is feasible, capable of meeting point-to-point and point-to-multipoint communication control requirements while resolving common underlying communication problems of high overhead and latency, making it highly suitable for distributed, low-latency applications like astronomical telescope observation control.

Our team aims to design and implement a hybrid communication pattern-based observation control system in subsequent work, enabling driving of all telescope terminal devices to complete actual observation control operations.

Acknowledgments

We sincerely thank the National Key R&D Program (2018YFA0404603, 2016YFE0100300), the Key Program of the Joint Fund of Astronomy of the National Natural Science Foundation of China and Chinese Academy of Sciences (U1831204), the Joint Fund of Astronomy of the National Natural Science Foundation of China and Chinese Academy of Sciences (U1531132, U1631129), the National Natural Science Foundation of China (11773012), the "Innovation Strengthening University Project" of Guangzhou University (2017KZDXM062), the Yunnan Provincial Applied Basic Research Project (2017FB001), and the CERNET Next Generation Internet Technology Innovation Project (NGII20170204). We also appreciate the support from the National Astronomical Observatories-China Alibaba Cloud Astronomy Big

Data Joint Research Center.

References

[1] Zhao, Yongheng. The observation control system of LAMOST[J]. Acta Astrophysical Sinica, 2000, (B12): 89-95.

[2] Kubánek P. RTS2 - the Remote Telescope System[J]. Advances in Astronomy, 2010, 2010:9.

[3] Kubánek P, Jelínek M, French J, et al. The RTS2 protocol[C]// Advanced Software and Control for Astronomy II. International Society for Optics and Photonics, 2008:70192S-70192S-12.

[4] Wei, Shoulin. Rearch on the Key Technologies of Distributed Computing and Control for Astronomy[D]. University of Chinese Academy of Sciences(Yunnan Observatory of Chinese Academy of Science),2017.

[5] Luo A-li, Tian Yuan, Song Jing, et al. Design and Implementation of LAMOST Observatory Control System[J]. e-Science Technology & Application, 2012, 3(4): 76-85.

[6] Deng Hui, Zhong Wenjie, Fu Yingxue,et al. The design of communication framework for a new generation of telescope autonomous control system based on ZeroMQ[J]. Astronomical Research & Technology, 2018(3).

[7] Hintjens P. ZeroMQ: Messaging for Many Applications[J]. Oreilly Media, 2013.

[8] Pu Fengping, Chen Jianzheng. Distributed system based on ZeroMQ[J]. Electronic Test, 2012(7): 24-29.

[9] Pan Huifang, Zhou Xingshe, Yang Gang. Design and Implementation of Messaging Middleware Based on Hybrid Communication Model[J]. Computer Engineering, 2006, 32(3): 116-118.

[10] Hintjens P. ZeroMQ: Messaging for Many Applications[J]. Oreilly Media, 2013.

[11] Chen Zengqiang, Guo Jialin, Liu Zhongxin, et, al. Design and Key Technology of the File-transfer System with the Function of Broken-point Continuingly-transferring[J]. Computer Engineering, 2002, 28(12): 14-16.

[12] Zhao, Yifan. Traffic control data sharing based on ZeroMQ message communication library exchange platform for the research and application[J]. China New Telecommunications, 2016, 18(6): 50-52.

Note: Figure translations are in progress. See original paper for figures.