Study on Real-time Processing of Spacecraft Telemetry Data

Zhong-wen LI1,*, Li YANG1 and Tian-tian SHEN2

1College of Information Science and Engineering, Hunan Normal University, Changsha, 410081, China
2College of Engineering and Design, Hunan Normal University, Changsha, 410081, China
*Corresponding author

Keywords: Real-time processing, telemetry data, spacecraft.

Abstract. In spacecraft automatic testing, massive telemetry data that generated by spacecraft are received by the testing system. The real-time processing of such testing data becomes critical. This paper analyses the characteristics spacecraft telemetry data, then builds up a mathematical model for real-time data processing with multi-machine. Experimental verification shows that the scheduling strategy can achieve real-time processing of telemetry data.

Introduction

In integration testing, data are generated by all kinds of sensors on spacecraft. The various values, called telemetry data, measured by satellite-borne devices are sent back to the ground via satellite-borne computers. Spacecraft testing process produces a large number of telemetry data, for example, spacecraft downlink telemetry data is amounted to about 600M per minute. In a multi-spacecraft parallel test environment, the strong data acquisition ability of the front end leads to great increase on load of data processing of the back-end. If these data is not processed in time, the test task will not be able to get data in time, or it will get invalid data, making the testing process invalid [1]. Therefore, it is necessary to use a more efficient data processing method to meet the requirement of real-time spacecraft testing data processing.

Characteristics of Telemetry Data

Spacecraft transmits telemetry data in the same way during a test as when it is in-orbit, which means telemetry subsystem receives telemetry data as continuous bit stream, and telemetry station decodes these signals into 8-bit words. Another job of a telemetry station is to assemble these separate words into “telemetry frames”. A frame of telemetry data is actually a block of data that consists of many 8-bit words, and how many words a telemetry frame contains is determined by model of spacecraft. A spacecraft is made up of many different subsystems (such as power supply subsystem, attitude control subsystem and so on). Telemetry data comprises interior data (data relating to status of spacecraft operation) and scientific data (various values produced by equipments, such as calculation value, pulse value and so on).

Downlink telemetry data transmitted during the testing process has the following characteristics:

Periodicity

In a spacecraft testing system, telemetry data is transmitted down as data flow in accordance with a certain period. Therefore, the real-time processing herein mentioned, means that the response time of data processing should be within one frame’s time interval, when telemetering is carried out on basis of frame. For a telemetric format that is processed on basis of package format, the response time should be within time interval of packet format. As long as the processing program completes within this time interval, it is considered to be real-time processed.
Linearly Separable

Processing telemetry data is mainly about analyzing the original data flow and converting the source codes in the data flow into engineering values, where the source codes with different parameters may correspond to different data processing algorithms, however, the computational complexity of data conversion and resource consumption in different algorithms can be approximately considered the same, therefore, the processing load is approximately proportional to the number of source codes in the telemetry data.

Analysis on Real-time Processing of Telemetry Data

As data processing algorithms with more superior performance are applied to spacecraft testing process, the performance of spacecraft data processing system is increasingly improving, but capability of single machine is limited, the traditional way of single-processor data processing are not able to meet the demand of real-time data processing of multi-spacecraft parallel testing, therefore multi-processing-machine parallel data processing has become an essential means for real-time spacecraft data processing [2]. At present, many more achievements have been made on research of load distribution algorithms of multi-processing-units parallel processing [3,4], and moreover, each load for load distribution in general are not decomposable, belonging to NP problems, so researches are more concentrated on using intelligent searching algorithms or heuristic algorithms for second-best solution [5].

Being different from traditional load distribution problems, in satellite-observed data processing, treatment for each data section in downlink data flow is relatively independent, therefore, in data parallelization [6], the total load can be decomposed in accordance with each independently processed data blocks, and each part of data treatment can be approximately seen as equal amount of load, so the total load can be approximately considered as linear separable. On the other hand, the network between each processing node that consists of multi-machine real-time processing environment for observed data and load distribution host is heterogeneous, so impact of network latency on load distribution is also need to be considered.

In the process of spacecraft testing, downlink telemetry data from spacecraft is in large quantity, small downlink cycle, and moreover, the original downlink data must be converted into effective data for the test task by some treatment, so using a single machine to real-time process this huge amount of data have become obviously unrealistic, multi-machine cooperative processing becomes a feasible solution, but in traditional multi-machine task scheduling, it often assumes that time for tasks to be transmitted from machine to machine can be ignored and realizes multi-machine cooperative work by balancing the loads between each machine. However, in the process of dealing with such a huge amount of data transferred down from spacecraft, the network transmission delay of sending the data processing tasks to multiple processing machines clearly cannot be ignored. Therefore, how to establish a multi-machine scheduling policy for real-time processing massive data with regard to the characteristics of telemetry data has become another key issue.

In the process of spacecraft testing, the basic idea of multi-machine real-time telemetry data processing is: to use a controlling host for allocating data processing loads, and on each processing machines’ completion of data processing, to transfer the processed data back to the host; to maintain an image of the real-time values of all the spacecraft parameters in the host. In the test task end, real-time values of various spacecraft state parameters that are needed by all the test tasks are obtained from the image.

Several factors need to be considered in multi-machine real-time processing of telemetry data are listed below:

Network Heterogeneous

A host of the center node takes the responsibility of receiving telemetry data in spacecraft testing. Every other data processing machine connects to the host through its own network and constitutes an
analogous star network structure. The bandwidth of networks that connect different processing machines to the host may be different.

**Network Transmission Delay**

Because the telemetry data are in large quantity, network transmission delay of allocating the loads between processing machines cannot be ignored.

**Dynamic Heterogeneity of the Processing Nodes**

Each data processing machine dynamically joins in the data processing system by configuration, and each data processing machine has a different capacity of load processing.

**Modeling**

In a spacecraft real-time data processing, with regard to the load distribution model (in the following identifiers, i=1,2,…,N):

- $p_0$: controlling host; $p_i$: each data processing nodes; $w_i$: calculation time required of data processing node $i$ for processing a load unit; $z_{i,j}$: time required for network transmitting a load unit between data processing node $i$ and controlling host; $D$: the total calculated load; $C$: transmission load of the tasks; $<l_i,p_i>$: denotes that $p_0$ connected to $p_i$ through $l_i$, where $i=1,2,…,N$, then the network structure is $Net(p_0)={<l_1,p_1>,<l_2,p_2>,…,<l_N,p_N>}$, assuming that when $P_0$ carries on the load distribution in accordance with the order of $Net(p_0)$. $\Psi$: total time consumed from the controlling host starting load distribution to each data processing nodes finishes load processing; $T_i$: time consumed from the controlling host starting load distribution to data processing node $i$ finishes load processing; $a_i$: assigned load to processing node $i$.

In a specified load distribution sequence, when all the processing nodes complete load computing at the same time, the total load processing time is minimized. So we can get below Eq. 1.

\[
\begin{align*}
    a_1w_1D &= a_2z_2C+a_2w_2D \\
    a_2w_2D &= a_3z_3C+a_3w_3D \\
    a_{N-1}w_{N-1}D &= a_Nz_NC+a_Nw_ND \\
    \sum_{i=1}^{N} a_i &= 1
\end{align*}
\]

make $\gamma_{i,j} = \frac{w_jD}{z_jC+w_jD}$

then the solution of the Eq.1 can be converted to:

\[
\begin{align*}
    a_1 &= \frac{1}{1+\sum_{j=1}^{N-1} \prod_{i=j}^{j+1} \gamma_{i,j+1}} \\
    a_i &= \prod_{j=1}^{i-1} \gamma_{j+1,j+1}(1+\sum_{j=1}^{N-1} \prod_{i=j}^{j+1} \gamma_{i,j+1}) \\
    &\text{, i=2,3,…,N}
\end{align*}
\]

The total load processing time is:

\[
\Psi = T_i = \frac{(z_iC + w_iD)}{(1+\sum_{i=1}^{N-1} \prod_{j=1}^{j+1} \gamma_{i,j+1})}
\]

given that the telemetry data downlink cycle is $\mu$, in order to meet real-time data processing requirements, the total load processing time $\Psi \leq \mu$. Namely
\[(z_i C + w_i D) / (1 + \sum_{j=1}^{x-1} \prod_{j=1}^{i} \gamma_{j,j+1}) \leq \mu \] (4)

When the processing capacity of each processing node is the same, namely \(w_1 = w_2 = \ldots = w_N = w\), and bandwidth of the networks between the controlling host and processing nodes are the same, namely \(z_1 = z_2 = \ldots = z_N = z\), from Eq. 4 we have:

\[
\frac{zC}{1 - (wD / (zC + wD))^N} \leq \mu
\]

\[\Rightarrow N \geq \frac{\lg(1 - zC / \mu)}{\lg(wD / (zC + wD))} \] (5)

Eq. 5 can be used to calculate the minimum number of processing nodes required with the peak telemetry data, under specific homogeneous network environment, for the purpose of real-time data processing.

**Load Distribution Strategy**

If there are \(N\) data processing nodes, make their processing speeds in a descending order \(\{p_1, p_2, \ldots, p_N\}\), and make all the nodes connected to controlling host in a descending order of network transmission speed \(\{l_1, l_2, \ldots, l_N\}\), as the network connection relationship \(\langle l_1, p_1 \rangle, \langle l_2, p_2 \rangle, \ldots, \langle l_N, p_N \rangle\), then the controlling host makes load distribution to each processing nodes by such given order. With this load distribution strategy the total data processing time will be the shortest.

**Experiment**

1. Multi-node processing: The experimental environment is composed of 6 PCs which forming a small local area network. Make controlling node read the whole archive file, allocate load among 5 processing nodes according to the above mentioned load distribution strategy, record the time from the controlling node starting to allocate to all the processing nodes finish processing. In this experimental environment, the processing capacity of the processing nodes is the same, and the speed of network transmission between controlling node and the processing nodes is also the same.

2. Single-node processing: Make one data processing node process the whole archive file. Read the archive file, and record the processing time from start to finish.

![Figure 1. Different processing time under different processing methods.](image-url)

Respectively make above two kinds of processing on 10 archive files with different sizes, each archive file is processed 20 times, and take the average time as the time for processing this archive file.
under different processing methods. The results are shown in Fig. 1. The proposed multi-node parallel data processing strategy is preponderance over than the common single-node processing.

Acknowledgement
This work was supported by the Scientific Research Fund of Hunan Provincial Education Department (Grant No. 17B161) and the National Natural Science Foundation of China (Grant No. 61803152).

References
[1] Li Z, Ye G, Ma S, et al. The study of spacecraft parallel testing [J], Telecommunication Systems, 2013, 53(1):69-76.
[2] Satzoda RK, Suchitra S, Srikanthan T. Parallelizing the Hough Transform Computation, IEEE Signal Processing Letters, Volume:15, 2008: 297~300.
[3] Chen G L, Sun G Z, Zhang Y Q. Study on parallel computing, J Comput Sci Tech, 2006.
[4] Yunquan, Zhang, Guoliang Chen, Guangzhong Sun, Qiankun Miao. Models of parallel computation: a survey and classification [J], Frontiers of Computer Science in China, 2007, 1(2):156~165.
[5] Ampazis N, Perantonis S J. Two highly efficient second order algorithms for training feedforward networks [J]. IEEE Transactions on Neural Networks, 2002, 13 (5): 1064~1073.
[6] V Bharadwaj, D Ghose, T Robertazzi. Divisible Load Theory: A New Paradigm for Load Scheduling in Distributed Systems [J], ClusterComputing, 2003, 6:7~17.