GreenSSCM: Green Software for Space Control Missions *

Ulisses Costa[1], Luis Teixeira[1], Miguel Lordelo[1], Tiago Carvalho[1], Orlando Belo[2,3], Victor Fonte[2,4], Pedro Henriques[2,3], António Ribeiro[2,4], and João Saraiva[2,4]

[1]VisionSpace, Porto
[2]Departamento de Informática, Universidade do Minho
[3]ALGORITMI [4]HASLab/INESC TEC
[1]{ucosta,luis.teixeira,miguel.lordelo,tiago.carvalho}@visionspace.com,
[2]{obelo,vff,prh,anr,jas}@di.uminho.pt

Abstract. This extended abstract briefly describes the techniques and tools developed in the context of GreenSSCM project to analyze energy consumption in software for space control missions. This project is being developed at VisionSpace in collaboration with Universidade do Minho.

1 Motivation

Since the very beginning of the space exploration, energy consumption plays a pivotal role in any space mission. In fact, hardware was always carefully designed and optimized in order to consume the minimum energy required to the success of the mission. Although hardware is still being carefully designed in terms of energy consumption, the intensive use of computing resources in the aerospace industry is also a main concern in the space industry.

In the space sector the most demanding software in terms of energy consumption are the *mission control system* (MCS) that receives communications directly from a satellite, performs the necessary computations, and transmits back to the satellite instructions to be performed. All the data transmitted to/by the satellite during a space mission is stored in complex and large data centers, which are analyzed in order to perform scientific computation relevant to this and future missions.

In the GreenSSMC we have developed techniques to monitor and analyze the energy consumption in those two contexts: First, we developed techniques to automatically instrument the source code of the *mission control software system* used by the European Space Agency (ESA), so that during the execution of the instrumented code it monitors energy consumption. To monitor energy consumption our techniques support both the use of Running Average Power Limit (RAPL) estimation framework provided by Intel, and and external data

* This work is partly funded by the Innovation Agency, SA, Northern Regional Operational Programme, Financial Incentive Grant Agreement under the Incentive Research and Development System, Project No. 38973.

acquisition device. Then, we developed a methodology, adapted from well-known software fault localization techniques[1], to relate and classify the energy consumed at runtime with the source code fragment (for example, a function, a loop, etc). We have defined a green/yellow/red classification of the source code fragments that are automatically marked with those colors, where green is an energy efficient fragment and red and energy inefficient one. As a result, the mission control software developers can immediately detect in their code which fragments of their software are responsible for an abnormal energy consumption.

The MCS used by ESA is implemented using C++ and compiled using GCC, so we decided to study, using the techniques referred above for monitoring the energy consumption, the energetic impact of the spacial and temporal optimizations of the GCC compiler. This study allowed us to determine what optimizations are more energy effective and to create a optimization suite designed to reduce the energy consumption of the MCS.

The analysis of large data collected by space missions is also a major concern in terms of energy consumption. In the context of the GreenSSCM project we have developed energy consumption plans for database queries execution in a typical data center environment [2]. These query execution plans aim to reduce as much as possible the energy consumption of such computing platforms without affecting its performance on the satisfaction of a regular querying process.

2 Conclusions

The energy optimization of software is still an area not well understood, and presents several challenges. From the measurement of the energy consumed, where the usage of more precise tools is key, to the identification of the impact of transformations in the energy consumption, where most of the times high level transformations do not have any impact and low level transformations which increase energy savings but require a case by case analysis and can not be fully automated yet.

With the work done so far we achieved some promising results regarding the reduction of the energy consumed by the MCS. Showing that this area has potential and room for further improvements.

In the future we plan to integrate all of the techniques and tools described above in order to create a platform that is capable of helping developers to create a more energy efficient software.

References

1. Carção, T.: Spectrum-based Energy Leak Localization. Master's thesis, University of Minho (supervisor: João Saraiva), Portugal (December 2014)
2. Goncalves, R., Saraiva, J., Belo, O.: Defining energy consumption plans for data querying processes. In: 2014 IEEE Fourth International Conference on Sustainable Computing and Communications, SustainCom 2014. (Dec 2014) 641–647