Journal of Network and Computer Applications 35 (2012) 1709–1719

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Energy-aware performance analysis methodologies for HPC architectures—An exploratory study

Shajulin Benedict*

HPCCLoud Research Laboratory, SXCCE, Anna University, 629003, India

ARTICLE INFO

Article history:
Received 24 March 2012
Received in revised form
18 July 2012
Accepted 7 August 2012
Available online 23 August 2012

Keywords:
HPC
Performance analysis
Tools
Energy monitoring

ABSTRACT

Performance analysis is a crucial step in HPC architectures including clouds. Traditional performance analysis methodologies that were proposed, implemented, and enacted are functional with the objective of identifying bottlenecks or issues related to memory, programming languages, hardware, and virtualization aspects. However, the need for energy efficient architectures in highly scalable computing environments, such as, Grid or Cloud, has widened the research thrust on developing performance analysis methodologies that analyze the energy inefficiency of HPC applications or their associated hardware. This paper surveys the performance analysis methodologies that investigates into the available energy monitoring and energy awareness mechanisms for HPC architectures. In addition, the paper validates the existing tools in terms of overhead, portability, and user-friendly parameters by conducting experiments at HPCCLoud Research Laboratory at our premise. This research work will promote HPC application developers to select an apt monitoring mechanism and HPC tool developers to augment required energy monitoring mechanisms which fit well with their basic monitoring infrastructures.

Contents

* Corresponding author. Tel.: +91 9443543746.
E-mail address: shajulin@sxcce.edu.in
URL: http://www.sxcce.edu.in/shajulin

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.08.003

1. Introduction

As we enter the era of high scale computing, recently, energy conscious design of HPC, Grid, or Cloud applications (Jing et al., 2011; Lin et al., 2007; Song et al., 2009) has motivated the minds of application developers (Dong, 2012). HPC researchers, application developers, and architecture designers got interested in GreenTop 500 or GreenTop100 lists of HPC applications instead of traditional Top 500 lists for HPC applications. The sole reasons are due to increased electricity billing and carbon emissions. According to a report submitted to US congress on Server and Data Centre Energy Efficiency in 2007, the energy consumption of US data centres was 61 billion kilowatts-hour in 2006 totaling dollar 4.5 billion (EnergyStar, 2012). It is predicted that the energy billing could increase in forthcoming years if precautions are not practiced in all levels including operating system, kernel, and application.

Although application developers are aware that they are responsible for reducing electricity bills and carbon emissions in environment, they find difficult in pointing out the exact code regions which lead to intolerable energy consumption. It is a known fact that majority of HPC applications has poor energy efficiency due to hefty wait times occurred either in pipelines or caches (Karkanis et al., 2002). A few major known causes of increase in energy consumption of HPC applications can be listed as follows:

1. Lack of hardware-level energy efficient designs. Although there have been some efforts such as dynamic voltage scaling or state changes, they are too complex—profiling individual hardware devices using special hardware equipments is not cost-effective.
2. Lack of energy-efficient scheduling algorithms in all levels including operating system level.
3. HPC application developers are not aware of how much energy might be consumed if an application is executed in HPC, Grid, or Cloud architectures. Energy consumption analysis by itself is challenging, i.e, Energy-based details are mostly hidden by hardware vendors—the required information is kernel specific or not supported by hardware.

Energy consumption analysis, hence, has insinuated in the minds of researchers recently. Existing energy analysis methodologies that are available in the market levies different technological flavors—even without the support of specialized hardware equipments to monitoring energy measurements: Choi et al. (2009) have estimated energy consumption for mobile parallel applications using the data obtained by performance events such as number of cache misses; Thanh et al. (2012) have estimated power consumption using resource utilization strategy; and Song et al. (2011) have modeled power consumption using iso-energy efficiency concept.

There is a need for a more comprehensive understanding about energy consumption of HPC applications, impact of the analysis, and available tools that could pinpoint the locations of massively parallel codes which create energy-based performance issues.

This paper presents an exploratory study of existing energy-aware performance analysis methodologies for HPC applications. This study would promote application developers to utilize the beneficiaries of sophisticated tools; hardware designers to benchmark their designs; and researchers to directly elect apt performance analysis methodologies.

The remaining sections of this paper are organized as follows. Section 2 describes the available energy-based performance analysis methodologies. Section 3 presents the available energy-based performance analysis tools which were productized. Section 4 validates the exploratory study of energy analysis methodologies using experiments. Section 5 concludes with some future research directions.

2. Energy monitoring and analysis methodologies

Obtaining raw electrical power usage data that are required to energize and cool HPC systems is multi-faceted in objectives. According to resource providers or owners, this is essential to

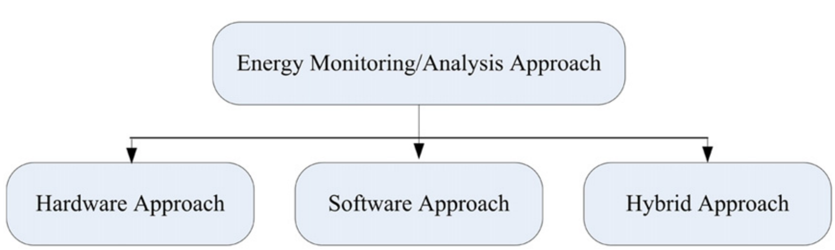

Fig. 1. Taxonomy of energy monitoring approaches in HPC architectures.