Solving Energy-Latency Dilemma: Task Allocation for Parallel Applications in Heterogeneous Embedded Systems

1. Introduction

Parallel applications with energy and low-latency constraints are emerging in various networked embedded systems like digital signal processing, vehicle tracking, and infrastructure monitoring. However, conventional energy-driven task allocation schemes for a cluster of embedded nodes only concentrated on energy-saving when making allocation decisions. Consequently, the length of the schedules could be very long, which is unfavorable or in some situations even not tolerated. In this project, we address the issue of allocating a group of parallel tasks on a heterogeneous embedded system with an objective of energy-saving and short-latency. A novel task allocation strategy, or BEATA (Balanced Energy-Aware Task Allocation), is developed to find an optimal allocation that minimizes overall energy consumption while confining the length of schedule to an ideal range. The rest of the proposal is organized as follows. In the next section we describe the motivation of this research. In Section 3, we propose the project summary. Project details are presented in Section 4. Section 5 concludes the proposal with summary and future directions.

2. Motivation

Extensive researches have been conducted to reduce overall energy consumption for a variety of embedded systems using diverse techniques [1][2][3]. In particular, most of recent researches in energy-saving for embedded systems share two common features (1) applications considered are real-time in nature with hard deadlines; and (2) energy-saving is achieved by employing DVS (Dynamic Voltage Scaling). Our work is fundamentally different from the above approaches as we focus on reducing both energy consumption and response time for soft real-time parallel applications running on heterogeneous embedded systems with no DVS available. In a heterogeneous embedded system, different processing nodes have distinct fixed energy consumption rates. Similarly, different communication channels also have various energy assumption rates. The goal of this work is to develop a task allocation strategy that not only conserves energy but also generates a short schedule, which is favorable or even necessary in some scenarios. For example, in a soft real-time embedded system such as a cellular phone [2], it must be able to encode outgoing voice and decode incoming signal during a conversation in a timely manner. Occasional glitches in conversations due to tardy response are not desired. When the response time becomes longer frequent glitches could happen, which are not tolerated at all.

Energy-saving and low-latency, however, are two conflicting objectives in the context of allocating a parallel application represented by a task graph onto a set of connected heterogeneous processing nodes in an embedded system. The dilemma arises from a multidimensional heterogeneity bearing by a heterogeneous embedded system. Specifically speaking, a processing node that provides a task with earliest finish time may not be an ideal candidate in terms of energy-saving. This is because the execution time of a task allocated on an embedded node is irrelative to the energy consumption rate offered by the node. Moreover, the computational energy consumption of a task allocated on a node is a product of energy consumption rate of the node and execution time of the task. The motivation of this work is to solve the energy-latency dilemma existed in networked heterogeneous embedded system where both energy-saving and low-latency need to be achieved. In this project, we address the issue by minimizing energy consumption while confining schedule lengths. To this end, we devised an energy-adaptive window to control the trade-off between energy consumption and response time.

3. Project summary

In this project, we address the issue of allocating tasks of parallel applications in heterogeneous embedded systems with an objective of energy-saving and latency-reducing. BEATA (Balanced Energy-Aware Task Allocation), a task allocation scheme considering both energy consumption and schedule length, is developed to solve the energy-latency dilemma. To facilitate the presentation of BEATA, we will also propose mathematical models to describe a system framework, parallel applications with precedence constraints, and energy consumption model. Extensive simulations using a real world application as well as synthetic benchmarks will be conducted to compare the performance of existing approaches with that of the BEATA scheme. The experimental results will show that BEATA significantly improves the performance in terms of

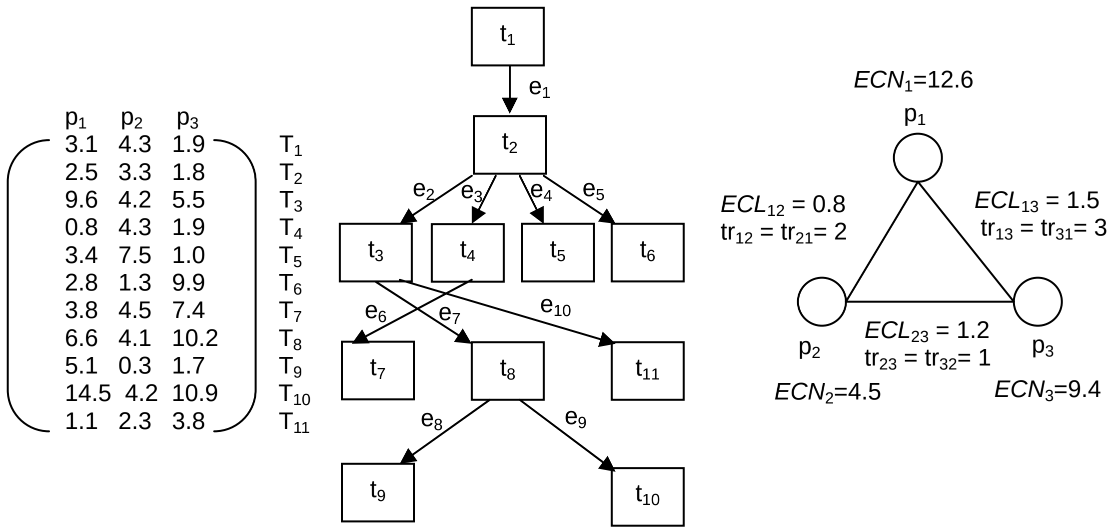

Figure 1. Example task and networked embedded system. ECN_i is the energy consumption rate of node *i*, and ECL_{ij} is the energy consumption of a link between node *i* and *j*.

energy dissipation and makespan time over two baseline allocation schemes.

4. Project details

4.1. Architecture

A networked embedded system in the most general form consists of a set, e.g., $P = \{p_1, p_2, ..., p_m\}$, of heterogeneous embedded computing nodes (hereinafter referred to as nodes or embedded nodes) connected by a single-hop wired or wireless network. The network embedded system can be represented as a graph of nodes along with their point-to-point links. In the system, an embedded node is modelled as a vertex. There exists a weighted edge between two vertices if they can communicate with each other. Each node in the system has an energy consumption rate measured by Joule per unit time. With respect to energy conservation, each network link is characterized by its energy consumption rate that heavily relies on the link's transmission rate, which is modelled by weight w_{ij} of the edge between node p_i and p_j. An allocation matrix X is an $n \times m$ binary matrix used to reflect a mapping of n tasks to m embedded nodes. Element x_{ij} in X is "1" if task t_i is assigned to node p_j and is "0", otherwise. Heterogeneity investigated in this study embrace multiple meanings. First, execution times of a task on different embedded nodes may various, since the nodes may have different processing capabilities. Second, a node offering task t_i a shorter execution time does not necessarily provide another task t_j with a shortened execution time, because different nodes may have distinct processor architectures. This implies that different nodes in a system are suitable for different kinds of tasks. Third, the transmission rates of links may be distinct. Last, energy consumption rates of the nodes may not necessarily be identical. For sake of simplicity and without any loss of generality, we assume that all nodes are fully connected with a dedicated communication system. Each node communicates with other nodes through message passing, and the communication time between two tasks assigned to the same node is negligible.

Applications with dependent tasks can be modelled by Directed Acyclic Graphs (*DAGs*) [3]. Throughout this project, a parallel application is specified as a pair, i.e, (T, E), where $T = \{t_1, t_2, ..., t_n\}$ represents a set of non-preemptable tasks, E is a set of weighted and directed edges representing communications among tasks, e.g., $(t_i, t_j) \in E$ is a message transmitted from task t_i to t_j. Figure 1 illustrates an example task graph and an example networked embedded system. The task graph has eleven tasks and the processor graph has three processors. The transmission rate and energy consumption rate of the channel between processor p_1 and p_2 are 2 and 0.8, respectively. The energy consumption rate of processor p_1 is 12.6. The matrix of execution times for each task on the three processors is illustrated as below. For example, task t_1 has execution time 3.1 second, 4.3 second, and 1.9 second on processor p_1, p_2, and p_3, respectively.

4.2. New challenges

- **Very limited power supply** In contrast to high performance computing platforms such as clusters located in data center buildings where electrical power is guaranteed, embedded systems only have very limited power supply provided by batteries. The lifetime of batteries of an embedded system directly determines the lifetime of the embedded system.

- **Stringent response requirement** Normally, embedded systems are used to collect information in a mobile environment. Quick response time is desirable or mandatory in many situations. Therefore, low latency must be achieved. Otherwise, the value of the results will be degraded or useless.
- **Achieving energy-saving and low-latency simultaneously** As we discussed in Section 2, energy-saving and low-latency are two conflicting objectives in the context of allocating a parallel application represented by a task graph onto a set of connected heterogeneous processing nodes in an embedded system. How to save energy while still keep response performance becomes more challenging in this context.

4.3. Contributions

Our main contributions include: (1) developing an energy-delay driven task allocation strategy called BEATA for collaborative applications running in heterogeneous networked embedded systems, (2) constructing an energy consumption model for quantitatively measuring energy caused by both computation and communications, (3) extending a heterogeneity model to reveal an inherent nature of a heterogeneous embedded system and its impact of system performance, and (4) simulating a heterogeneous embedded system where the BEATA strategy is implemented and evaluated.

We will use Matlab to compose a simulator, which simulates a set of connected heterogeneous processing nodes in an embedded system. In addition, our trace generator will produce synthetic workload for the simulated embedded system. For real-world application trace data, we will use DSP (Digital Signal Processing) because it is a typical parallel application. The impacts of parameters that will be examined in our experiments are:

- Number of tasks
- Energy-adaptive window
- Task execution time range Number of nodes
- Number of tasks

The performance metrics by which we evaluate system performance include:

- Makespan (the latest task completion time in the task set represented by a DAG).
- Energy consumption: total energy consumed by the task set including computation energy consumption and communication energy consumption.
- Utilization standard deviation (USD): standard deviation of nodes utilization in the simulated heterogeneous embedded systems.
- Energy standard deviation (ESD): standard deviation of the energy consumption of nodes in the simulated heterogeneous embedded systems.

4.4. Timeline

- *Paper reading (5 Feb. 2007 to 20 Feb. 2007)* extensive reading include most recent representative related work will be carried out from. A reading summary will be also conducted during this time period.
- *Simulator construction (21 Feb. 2007 to 21 March 2007)* compose a heterogeneous embedded system and synthetic workload generator using Matlab from.
- *Experiments (22 March 2007 to 8 April 2007)* conduct extensive simulation experiments from. Around 5 groups of figures with each of 4 plus two data analysis tables will be finished during this stage.
- *Paper writing (9 April 2007 to 9 May 2007)* write an IEEE format technical paper with around 8 pages (double-column) from.

5. Conclusions

In this project, we address the issue of allocating tasks of parallel applications in heterogeneous embedded systems with an objective of energy-saving and latency-reducing. BEATA (Balanced Energy-Aware Task Allocation), a task allocation scheme considering both energy consumption and schedule length, is developed to solve the energy-latency dilemma.

Future studies in this research can be performed in the following directions. First, we will extend our scheme to multi-dimensional computing resources from which energy-saving can be achieved. For now, we simply consider CPU time and network communication time. Memory access and I/O activities will be considered in the future. Second, we intend to enable the BEATA scheme to deal with real-time parallel applications, where the hard deadlines must be guaranteed.

References

[1] W.S. Conner, L. Krishnamurty, R. Want, “Making Everyday Life Easier Using Dense Sensor Networks,” *Ubicomp*, pp. 49-55, 2001.

[2] R. P. Dick, D. L. Rhodes, and W. Wolf, “TGFF:Task graphs for free,” *Proc. Int. Workshop. Hard-ware/Software Codesign*, pp. 97-101, Mar. 1998.

[3] D. Estrin, L. Girod, G. Pottie, M. Srivastava, “Instrumenting the world with wireless sensor networks,” *Proc. of the Int’l Conf. on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, May 2001.