Distributed embedded system for communication based on ant colony optimization algorithm

Article *in* Metallurgical and Mining Industry · April 2015

4 authors, including:

Yang Zhao
GuangDong University of Technology
62 PUBLICATIONS **110** CITATIONS

Distributed embedded system for communication based on ant colony optimization algorithm

Wenbo Geng [1, 2]

[1] *School of Physics and Electromechanical Engineering, Zhoukou Normal University, Zhoukou 466001, Henan, China*

[2] *Key Laboratory of Cloud Computing and Internet of Things Applications, Zhoukou Normal University, Zhoukou 466001, Henan, China*

Honghui Zhang, Quan Cheng

School of Physics and Electromechanical Engineering, Zhoukou Normal University, Zhoukou 466001, Henan, China

Yang Zhao*

Department of Electronic and Information Technology, Jiangmen Polytechnic, Jiangmen 529090, Guangdong, China

**Corresponding author is Yang Zhao, email:zhaoyang19781023@gmail.com*

Abstract

In this paper, a new UML-based distributed embedded system is prompted for communication based on ant colony optimization algorithm to solve the problem of low efficiency, error and delay in communication system. A described model is suitable for the communication of hardware and software co-design of distributed embedded system. This model, unlike well-suited for procedural specifications, is a dataflow model. Based on ant colony optimization algorithm, an algorithm is implemented in C++ and performed experiments on several examples. Communication delay is detailed analyzed under interaction of computation and communication. The model is not limited to network designs. Rather, it is aimed at modeling the general class of distributed embedded systems. The experiment shows that new system can achieve better performance.

Key words: UML, DISTRIBUTED EMBEDDED SYSTEM, ANT COLONY OPTIMIZATION ALGORITHM, COMPUTATION AND COMMUNICATION

Introduction

With the development and changes of computer industry, the research on embedded system is becoming a popular field now. The rapid developments of embedded systems play a significant role in nation economic and deeply promote the informatization and intellectualization of human society. Embedded systems often work in the harsh and complex environment which make it, especially the battered-powered systems, face serious energy consumption constraints. Furthermore, with the further evolvement of Internet and the spring up of the Internet of Things, embedded systems are increasingly threatened by security issues. How to design energy-efficient and high secure embedded systems becomes be one of the great challenges that we never faced before. From initial single chip structure to nowadays embedded structure with special operating system, the complexity and scale of embedded system is growing and extending. Consequently, how to design the complex embedded system is becoming an important task that needs to be discussed now. However, the traditional analysis and design way of embedded system has no unified standards, which makes the system unsatisfying in costs and efficiency, so it is urgent to explore a new way to develop embedded software[1].

However, there are still many problems while UML is used in requirements elicitation and modeling stage of embedded systems. One of the problems is the embedded non-functional attributes cannot be expressed in UMLdiagrams. As the standard of object oriented modeling technique [2], UML is suitable for making the design of complex system simplify. It adopts the way of object graphics oriented to describe the system, and it supports the whole process of analysis, design and modeling. As to the analysis and design for embedded system, UML has its special advantages. Through the research of the current technology of UML, combining with the existing problems in the design of embedded software, this thesis presents a solution based on UML to develop the embedded system, and applies it to a system instance.

The modeling technology based on UML can resolve a series of problems from analysis and design to verification and coding in the development of embedded software. Compared to traditional methods, this method can improve the efficiency and quality of the software development, and it can also reduce the cost and risk of the project. In addition, it not only makes software maintenance easier but also makes the reuse of analytic and design models in the similar system possible.

Overview of UML-based distributed embedded system for communication

Communication is critical for the performance and cost of distributed systems. The algorithm in the paper is based on the performance estimation algorithm and the co-synthesis algorithm, so it can be improved in a similar way. In addition, we think the following improvements are important for communication synthesis:

Cache effects can affect the interaction between CPU and communication channels. Li, Malik and Wolfe's algorithm and others which analyze caching effects can be used to adjust available bus bandwidth. A more accurate model for more communication protocols is crucial. For instance, a serial bus controlled by interrupt service routines produce several small periodic processes for a single message, where each communication process transfers one byte of data How to intelligently partition a long message into small chunks is also an interesting problem.

Many modem microprocessor has on-chip dual-port memory, on-chip cache, more than one address bus and data bus, etc. The size of on-chip dual-port memory is limited. If the communication data are fewer than the dual-port memory size, the microprocessor may perform computation and communication in parallel; but if the communication data are too large, rnmputation will be suspended by communication.

A well-defined model is needed for several reasons. First, we can use automatic generation to create large numbers of generic model instances, thus permitting application of a new partitioning heuristic on many examples, providing enough data for a good evaluation of average/worst/best case heuristic performance. If instead we used real examples, we would need several weeks to develop each the example, meaning that we could only develop a few examples, making heuristic evaluation difficult. Second, we can freely communicate model instance among researchers, because the model does not represent a system's full functionality and thus does not contain proprietary information typically found in real examples, even when the model is derived from a real example. Such communication enables fair comparison of partitioning heuristics. Finally, a well-defined model provides for partitioning from a partial specification, because the designer need only summarize the information necessary to build the model (e. g., the size and parameters of a procedure rather than its contents). The paper a

new communication modeling is presented to analyze communication delay under interaction of computation and communication, allocate interprocessor communication links, and schedule communication.

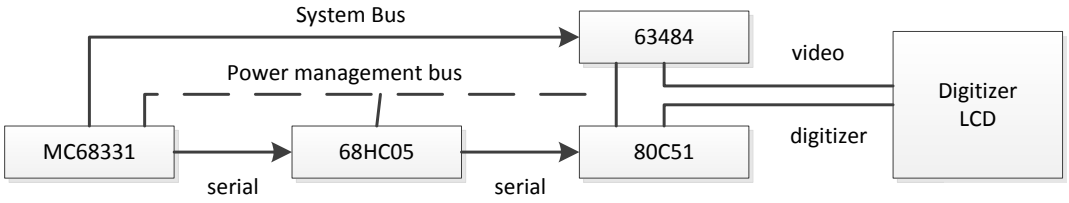

Figure 1. The pen based system

Many embedded systems use custom communication topologies and the communication links are often a significant part of the system cost .This paper describes new techniques for the communication requirements of embedded systems during co-synthesis. Most previous work uses two kinds of communication topologies; either a point-to-point communication link for each pair of processors, or a single bus/network for all interprocess communication. Embedded systems are application specific, and they can have ad hoc architecture with several uses, each with a different speed and a different number of PEs connected to it, according to the real-time constraints imposed on individual tasks. For instance, a bus connecting five CPUs may be only two CPUs for time-critical messages. Rosebrugh and Kwang described a pen-based system built from four processors of different types: a Motorola MC68331, a Motorola MC68HCOSC4, a Hitachi 63484, and an Intel 8051. There are five buses with different connections for interprocessor communication in their pen-based system: power management bus, serial bus, video bus, digitizer bus, and system bus, as shown in Figure 1.

Communication is the bottleneck in many embedded systems, because communication links add both chip and board costs, and designers frequently underestimate peak load. Design decisions based on average communication requirements may lead to an infeasible design. The communication must be scheduled and allocated to determine feasibility, and communication synthesis interacts with process scheduling and engine design. The performance estimation algorithm is extended to include communication delay to develop methods for synthesizing communication links, based on the co-synthesis algorithm. The communication synthesis algorithm selects the number of buses, the messages transferred on each bus, and schedules the bus communication. This paper gives the results of experiments with the algorithms.

Characteristics of UML-based methodology

UML uses a simple and intuitionistic graphics mode to describe all the problems and details in a system design. With the simple UML symbols, the designers with different technological background can communicate and design together easily. UML supports the visual modeling. Syncretizing some new ideas, new methods and new technology in the domain of software engineering, UML supports not only the OOA (Object-Oriented Analyze) and OOD (Object-Oriented Design), but also the whole process of software development starting from the requirement analysis [3].

For these characteristics of UML, using UML to model and write the documents in the embedded system design can get twice the result with half the effort. multi-parameter monitor is widely used in operating room, ICU sickroom, CCU sickroom and other places where ward patients. It can display three kinds of wave which are ECG wave, respire wave and SP02 wave, and display 8 parameters which are heart rate, systolic pressure, diastolic pressure, mean pressure, saturation oxygen, pulse rate, breathing rate and temperature. We can adopt Use Case diagram to model in function of a system when do the requirement analysis. For the mufti-parameter monitor, the actor is the user of monitor. The mufti-parameter monitor can supervise some physiological parameters such as heart rate, systolic pressure, diastolic pressure, mean pressure, saturation oxygen, and pulse rate. breathing rate and temperature, and can display in LCD in mode of wave, parameter and list. The mufti-parameter monitor also have functions of setting the color of wave and parameter, playback the record of wave, show the trend diagram of

every parameter in 96 hours, setting the higher limit and lower limit, and alarming when the parameter surpass the limit, which can help patient accept cure in time. The Static Model described the system by the modelled static structure. Static Model has defined the class of objects, the amibutes of a class, the relationship between classes and each kind of operation in a class .Take the Class diagram as an example, to introduce the processes of establishing a Class diagram in the multi-parameter monitor system.
It also has some measuring modules in this system, like ECG Model, Class diagram can make a model of these real world equipments. Multi-parameter monitor is a complex embedded system whose dynamic behavior model is described by Statechart Diagram, Collaboration Diagram, Activity Diagram and Sequence Diagram, taking into account the length, the paper only gives tow important UML diagrams.

Statechart Diagram is at the state of idle at first. After the user choosing demonstration or the real-time monitoring mode, the system starts to wait for the output signal, if it does not accept the signal, enters the idle condition, if accepts the signal, it will enter the displaying condition, finishes displaying, the system returns to the waiting condition, if the signal value surpasses the higher or lower limit value, then enters the alarming condition, after the user canceling the alarm, then enters the state of waiting for output signal.

Implementation of ant colony optimization algorithm

We call d_{ij} the length of the path between towns i and j. In the case of Euclidean TSP, d_{ij} is the Euclidean distance between i and j (i.e., $d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$). An instance of the TSP is given by a graph (*N*, *E*), where N is the set of towns and E is the set of edges between towns (a fully connected graph in the Euclidean TSP). Let $b_i(t)$ (*i=1, 2,..., n*) be the number of ants in town at time t and let $m=\sum_{i=1}^{n} b_i(t)$ be the total number of ants. Each ant is a simple agent with the following characteristics: it chooses the town to go to with a probability that is a function of the town distance and of the amount of trail present on the connecting edge to force the ant to make legal tours. When it completes a tour, it lays a substance called trail on each edge (*i, j*) visited.
Let $\tau_{ij}(t)$ be the intensity of trail on edge (*i, j*) at time t .Each ant at time t chooses the next town, where it will be at time t+1 .Therefore, if we call an iteration of the ACO algorithm the m moves carried out by the m ants in the interval (t, t+1), then every n iterations of the algorithm (which we call a cycle) each ant has completed a tour. At this point the trail intensity is updated according to the following formula [4]:

$$\tau_{ij}(t+n)=\rho\tau_{ij}(t)+\Delta\tau_{ij} \quad (1)$$

where ρ evaporation is a coefficient such that (1-ρ) represents the of trail between time t and t+n

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \quad (2)$$

Where $\Delta\tau_{ij}^{k}$ is the quantity per unit of length of trail substance (pheromone in real ants) laid on edge (i, j) by the *k*-th ant between time t and t+n .It is given by

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_K} & \text{if k – th ant uses edge (i, j) in its tour (between time t and t+n)}\\ 0 & \textit{otherwise}\end{cases} \quad (3)$$

where Q is a constant and Lk is the tour length of the k-th ant. The coefficient ρ must be set to a value $\rho<1$ to avoid unlimited accumulation of trail. In our experiments, we set the intensity of trail at time 0, $\tau_{ij}(0)$, to a small positive constant c.

The tabu list is then emptied and the ant is free again to choose. We define tabuk the dynamically growing vector, which contains the lobo list of the *k*-th ant, tabuk the set obtained from the elements of tabuk and tabuk(s) the s-th element of the list (i.e., the s-th town visited by the k-th ant in the current tour) [5].

The visibility η_{ij} is called the quantity η_{ij}. This quantity is not modified during the run of the ACO algorithm, as opposed to the trail, which instead changes according to the previous formula (1). We define the transition probability from town i to town j for the k-th ant as

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\cdot\eta_{ij}^{\beta}}{\sum_{s\in allowed_k}\tau_{is}^{\alpha}(t)\cdot\eta_{is}^{\beta}} & j\in allowed_k\\ 0 & \textit{otherwise}\end{cases} \quad (4)$$

Where

$$allowed_k=\{N-tabu_k\} \quad (5)$$

And where α and β are parameters that control the relative importance of trail versus visibility [6].

Communication result

Figure 2 shows how a sending process and a receiving process may be inserted into an

edge in a task graph for various cases. Dash boxes represent PEs. A small solid box a dual-port buffer for a PE. If two processes connected by an edge are allocated to the same PE, or either of the processes is a dummy process, no communication process needs to be created for this arc. When two processes connected by an edge are allocated to different PEs, at least one communication process needs to be created for the corresponding message. If there is a dual-port buffer for the process receiving the message, the receiving process is not necessary. If there is a dual-port buffer for the process sending the message, but no dual-port buffer exists for the receiving side, the sending process can lie deleted. The existence of a communication process depends on the allocation of other processes. When there is more than one edge leaving a process in a task graph, and no dual-port buffer is available, a communication process may be shared by different edges. Such cases are described in Figure 3. Dash boxes and dash line represent PEs and communication links in the current allocation respeclively. The destinations of two edges leaving the same source process are allocated to the same PE, both the sending process and the receiving process can he shared, because once the same message is read into the local memory for the first process, the second process can also read it without accessing the bus again. If the destinations of two edges are on two different PEs which use the same bus used by the PE containing the source process, the sending process can be shared, although the receiving processes must be separated. The sharing of sending process, a bus might provide better performance than point-to-point communication links [7].

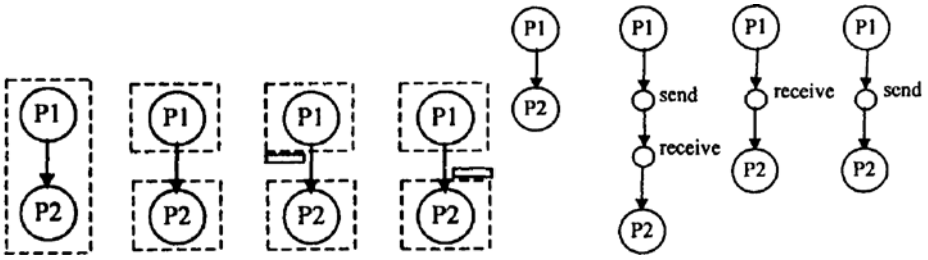

Figure 2. The creation of communication processes for various situations

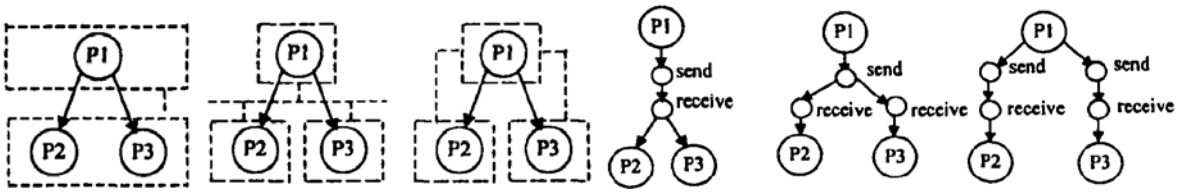

Figure 3. The sharing of communication processes for various situations

Conclusion

The cost of the system is dominated by the cost of the processors selected; each PE can perform computation and communication in parallel Prakash and Parker's run times were measured on a Solbourne 5c/900 (similar to Sun SPARC 4/490). Our run times were measured on a Sun Sparc 20 Workstation. We compare our results with theirs for three designs with different hard cost constraints. Our result for (the task delay is close to their result, but the efficiency of our algorithm is better. Their algorithm is unable to handle more complex communication models, such as the cases where the cost of buses are significant, computation and communication have resource conflict, or there are multiple buses. We also combine Prakash and Parker's two examples together and assign the periods as well as the deadlines of 7 and 15 to the two tasks. This example demonstrates how our algorithm can co-synthesize from multiple disjoint task graphs for commumcation. Our modeling is not limited to network designs. Rather, it is aimed at modeling the general class of distributed embedded systems.

Acknowledgements

This work was supported by the Natural Science Foundation of Education Department of Henan Province (Grant 2011B510020,2011B510022) and the Key Technologies R & D Program of Henan Province (Grant 132102210101,142102210580);the Research Projects of Henan Province under Grant (142400411058,132300410276).

References

1. Akhavan A, Samsudin A, Akhshani A (2011) A symmetric image encryption scheme based on combination of nonlinear chaotic maps. *Journal of the Franklin Institute-engineering and Applied Mathematics*, 8(2), p.p. 1797-1813.
2. Hongliang Chen, Shaohua Zhou (2013) Location-based Personalized Recommendation in M-Commerce. *Statistics and Decision*, 21(6), p.p. 21-25.
3. Rosebrugh, Kwang E K.(1992) Multiple microcontrollers in an embedded system. *Dobbs journal*, 22(5), p.p. 48-57.
4. Tai Jiang-zhe, Meleis W, Zhang Jue-min (2013) Adaptive Resource Allocation for Cloud Computing Environments under Bursty Workloads, Boston. USA, p.p. 978-987.
5. Tirado J M, Higuero D, Isaila F, (2011) Predictive Data Grouping and Placement Ivor Cloud-based Elastic Server Infrastructures. *Proc. Conf. On Cluster, Cloud and grid Computing*, p.p. 281-294.
6. Dagdeviren, M., Yuksel, I.(2008) Developing a fuzzy analytic hierarchy process (AHP) model for behavior-based safety management. *Information Sciences*, 178(6), p.p.1717-1733.
7. Yen T Y, Wol W., (1995). Sensitivity-driven synthesis of distributed embedded systems. *Proc. Conf. on System Synthesis*, p.p. 167-171.