ENHANCED RESOURCE LEVELING INDYNAMIC POWER MANAGEMENT TECHNIQUE OF IMPROVEMENT IN PERFORMANCE FOR MULTI-CORE PROCESSORS

Hamayun Khan¹, Anila Yasmeen², Sadeeq Jan³, Usman Hashmi¹
Sheeraz Ahmed⁵*, M.Yousaf Ali khan⁶, Irfan-ud-din⁷

¹, ⁴, ⁷Superior University, Lahore, Pakistan
²Hamdard University, Islamabad Campus, Pakistan
¹, ⁶Gomal University, D. I. Khan, Pakistan
³Department of CSIT, UET Peshawar, Pakistan
⁵Department of Computer Science, Iqra National University, Peshawar, Pakistan

*e-mail: sheeraz.ahmad@inu.edu.pk

Abstract

The criteria to judge the capacity of computational systems is changing with the advancement in technology. Earlier, they were judged only on the basis of computational capacity but now a day, power and energy optimization is one of the key parameters for their selection. The purpose of energy optimization is to prolong the battery life of all the battery operated devices especially in embedded systems. An Offline Scheduling Algorithm technique is proposed that migrate task load to the core that has less thermal values in response to a threshold temperature this technique also considers other thermal problems which affect the power, reliability and performance of multi-core system. Hardware technique on their own is insufficient so it must be combined with other software techniques to decide when and where optimization policies are applied to minimum energy consumption. This paper focuses on most popular optimization techniques Dynamic Voltage and Frequency Scaling (DVFS), Dynamic Power Management (DPM) and Dynamic Thermal Management (DTM) and their extensions. The paper also includes the thermal issues which are raised due to high temperature in multi-core platforms. It also highlights that how energy efficient techniques can be used beyond simple energy saving. The simulation results shows that the proposed technique reduces almost 4.3°C temperatures at 17% utilization and the energy utilization is 364.58 J which is 4.14 % improved as compare to the global EDF Scheduling technique used previously.

Copyright reserved © J. Mech. Cont.& Math. Sci.
Hamayun Khan et al
Keywords: Dynamic Voltage and Frequency Scaling, Dynamic Power Management, Dynamic Thermal Management, Earliest Deadline First, Least Laxity First

I. Introduction

We are living in the world which is evolving at much faster rate than ever before. Today embedded systems are everywhere from simple toaster to a large scale telecommunication system, defense system and many others. The growth rate in embedded systems is almost 10% per year and it is predicted that till 2020 almost 40 billion embedded devices worldwide. We generally define embedded system as a sub component of a larger system. Combination of hardware and software, designed to perform dedicated functionality. While designing embedded systems, there come few challenges such as size, cost, power consumption and reliability. Most of the embedded systems are battery operated. To achieve the maximum performance while using minimum power consumption is the major design challenge in all electronic/embedded systems. According to Moore’s law number of transistors on a chip is doubling in almost every 18 months [I]. The continuous decrease in chip size, chip density and frequency has made power consumption a major issue. Although the multi-core technology delivers a lot but it faces a number of thermal and power issues. High peak temperatures imbalanced the temperature gradients and affect the reliability of the system and increasing the cooling cost. So the main purpose of this research is power optimization and we specifically focus on those techniques which minimize energy consumption with very small impact on performance. This will not only include optimization policies for embedded systems but also include system-level, thread level power management techniques for multiprocessors, multi-core processor and real time embedded systems [II].

Three techniques of power optimization are used in Global schedulers 1) Dynamic Voltage and Frequency Scaling (DVFS) 2) Dynamic Power Management (DPM) 3) Dynamic Thermal management (DTM). DVFS adjusts the voltage and frequency in such a way that minimum energy is consumed. DVFS algorithms find the best setting that saves the maximum power while sacrificing the minimum speed. DPM is a design technology which reconfigures the whole electronic system dynamically in such a way that requested services can be provided with the minimum number of active components with suitable performance levels [III], [IV],[V]. DPM technique, selectively turn off all those components which are not in use. DPM is used in several portable systems but its applications are not yet explored because of the complexity of interfacing heterogeneous components. The fundamental problem in implementation of DPM techniques is non-uniform workload during execution time [VI]. To solve this problem DPM use predictive algorithm which predicts the future workload by using different predictive models?It covers several system level DPM approaches to save energy DTM technique is used to find an optimal solution to avoid peak temperature which causes hot spots on chips. Energy is not directly affected by temperature, but when temperature increases from threshold value some cooling mechanism is required to reduce the temperature. Cooling
mechanism consume electricity to reduce the temperature so if some temperature management techniques are introduced it decreases the cooling cost [VII], [VIII].

II. Background

A. Power Dissipation Basics

Most of the electronics circuits aim to give maximum performance while using minimum power. Electronics circuits are viewed as combination of different Complementary Metal Oxide Semiconductor (CMOS) devices. CMOS devices are the basic building block of all computing systems. CMOS semiconductor use negative polarity metal oxide semiconductor (NMOS) or positive polarity metal oxide semiconductor (PMOS). CMOS circuits consume less power as compared to other devices which are using just one type of transistor. This property makes them unique for usage in embedded systems which are mostly battery operated [VIII]. Power consumed in the CMOS is divided into three major components. These components are 1) dynamic power 2) static power 3) short circuit power. Among these three components dynamic power is the most dominant component.

\[ P_{\text{Total}} = P_{\text{Dynamic}} + P_{\text{Static}} \quad (1) \]

Dynamic power is also known as switching power. Dynamic power can be calculated by using the given formula:

\[ P_{\text{Dynamic}} = C \cdot K \cdot V^{dd^2} \cdot f \quad (1.1) \]

Where C is the load capacitance, K is the average number of switches or transitions per clock cycle, V^{dd} is the supply voltage which is equal to the switching voltage in most of the cases and f is the clock frequency. In a single core processor if we increase the frequency by 50% roughly increase the power consumption two times, however in dual-core systems power consumption increases 30%. If we increase the supply voltage by keeping frequency constant power increases more rapidly because power is directly proportional to the square of supply voltage.

To optimize the dynamic power we can reduce the working frequency which saves considerable amount of power but causes the performance degradation. In the same way reducing the supply voltage, reduces the dynamic power almost four times but it has an overhead by decreasing supply voltage circuit delay is increased so the circuit cannot be operate at the same working frequency. If we decrease supply voltage and frequency dynamic power decreases cubically but increases the execution time linearly [IX], [X]. Second major component of power dissipation is static power which is due to leakage current [XI]. This power consumption occurs when circuit is not changing states. Leakage current along with power supply causes static power consumption. The sources of leakage power are sub threshold leakage, p-n junction leakage and gate leakage. But the dominant factor is sub threshold leakage which represents the power dissipated by the transistor when it is deliberately to be off. Leakage power consists of almost 20 to 40
percent of total power dissipation [XII]. Leakage power is also dependent on temperature as the temperature raises leakage power increases exponentially. Multiple efforts have been taken to reduce static power dissipation at processor and micro architectural level [XIII], [XIV].

Third component of power dissipation is short circuit power which is very low as compared to dynamic and static power. Short circuit power is the power dissipated when both PMOS and NMOS are on for a short period of time. Its means when both PMOS and NMOS are on then there is a direct path between supply voltage and ground. Due to small impact of short circuit power on total power has not drawn much attention of researchers [XX].

B. Effects of Temperature on Power Consumption

As the chip size is shrinking and numbers of transistors are increasing per unit area on chip this leads more power consumption, more chip complexity and increases the temperature of the chip. Peak temperature reduces the lifespan of the chip, affects its performance and also increases the cooling cost [XV]. To reduce peak temperatures additional cooling efforts are required, a typical cooling fan can consume 51% power budget of the server [XVI], [XVII].

Leakage power is strongly dependent on temperature. Leakage current increases the temperature which increases the leakage power consumption. Leakage power is increasing as the technology grows up according to ITRS roadmap [XIX] a graph is shown with the passage of time. In section 5 we will see how these problems are solved by using Dynamic thermal management policies.
C. Earliest Deadline First (EDF) Algorithm

The jobs using EDF scheduling can have dynamic priorities. Scheduler can select those tasks that are in ready queue whose deadlines are coming next and very useful for uni-processor. The utilization factor of the process is less than is approximately equal to 1. In hard real time systems these jobs are schedulable for preempt able task and can attain 100% utilization for a central processing unit.

By using EDF scheduling technique the utilization of central processing unit improves because of implication of dynamic priority. The priority of selecting each task depends on the fixed deadlines. [XXI].

D. Task using Dynamic Priority Scheduling

Where a task with different jobs can have different priorities for its works even if they are in running state. In other words once a job is set, there is no restriction on it to change its priorities for limited number of times either the processor finishes its current running task e.g. Least laxity first algorithm.

E. Least Laxity First (LLF) Algorithm

This Scheduling algorithm is defined as when the remaining computation time can be subtracted from the Deadline of a job. In this scheduling algorithm a task can meet its deadline with highest waiting duration but meet its deadline. Jobs that are active with least laxity can have more chances of higher priority are scheduled first. LLF is used for periodic task and having dynamic priority for each task [XXII]. It is the waiting duration of a job that can be accepted before chosen for execution it is represented as $L=D-T-(E-C)$.
The deadline of a job in which any task can complete its function is known as computation duration. In minimum laxity first only those tasks are executed first that have least priority among all the jobs but in this case priority is already assigned to each task as well if there are two processes e.g. process1 and process2 that are running simultaneously and both have same priorities then process 1 will be in running state and then switching process will be occurred and preempted the process 1 and start running process2 [XX, XI, VI]. Figure 2 illustrates the scheduling chart for task T1, T2, T3 using LLF’.

II. Proposed Technique

A. Proposed Algorithm

In the above scheduling algorithm the procedure works for all real time tasks in running state due to which all the new task in ready state can move towards scheduling phase and execute with respect to time. The scheduling technique improves overall working performance of the chip as compared to the previous techniques used in earliest deadline first as well as in shortest job first algorithm. When the tasks are ready to execute our proposed scheduler can verify and check the compatibility of the task with the scheduler because in the requirement are more than available VMs the tasks are rejected.

```
Algorithm 1: Scheduling_Task()

Input: Q1, L, Q2
Output: L, L_T
Step 1: Initialize Q1 with L number of leases
Step 2: Set CT = min{L_i(ST_i)} from Q1 and set count = 0
Step 3: if AT_{max} < ST_{max}
    go to step 6
end of if
Step 4: while Q2 is NULL or CT <= D_{max} or count <= L
    Repeat the following steps 4 to 7
Step 5: for i = 1 to L do the following
    if ST_i <= CT
        Move lease T_i from Q1 to Q2 and modify Q2
        Find GT_i using Eq. (3) and sort in ascending order in Q2
    else
        break
    end of if else
end of for
Step 6: for i = 1 to L
    if L(n) == N and L_i(GT_i) == min(GT)
        Schedule lease L_i
    else
        find a suitable pair of lease l_i and l_j where L_i(n)*l_j(n)
        <= N
        Schedule lease l_i and l_j
    else
```

Copyright reserved © J. Mech. Cont. & Math. Sci.
Hamayun Khan et al
Algorithm 2: Task_Acceptance()

Input: L, CT, ST, E and DT
Output: L_o, Q_l
Step 1: If CT<=L(ST)
Step 2: Put in Q_l
Step 3: Update Q_l
Step 4: Find similar tasks
Step 5: for each similar tasks
   do
      Find free VMs
      Invoke Scheduling_Task()
      Update execution time
   end of for
Step 6: Update CT
Step 7: Find GT
end of if
Step 8: Stop

Algorithm 3: Task_Rejection()

Input: L, CT, E, DT, n and N
Output: l_o, l_u
Step 1: Check compatibility
   if \((CT + L(E)) > L(DT) || (L(n) > N))
      Reject the task
   else
      Task_Acceptance()
   end of if
Step 2: Stop

Fig.3: Block Diagram of Temperature controller Model [VII].
Table-1: Parameters of the Scheduler

| Parameter   | Description                                      |
|-------------|--------------------------------------------------|
| $Q_1, Q_2$  | Queues to store the leases                       |
| $L$         | Number of leases to be scheduled                 |
| $L_i$       | Any lease in $L$                                 |
| $CT$        | Current time                                     |
| $AT_{max}$  | Maximum arrival time of $L$ leases               |
| $ST_{min}$  | Minimum start time of $L$ leases                 |
| $D_{max}$   | Maximum deadline of $L$ leases                   |
| $ST_i$      | Start time of lease $L_i$                        |
| $GI_i$      | Gap time of lease $L_i$                          |
| $L_i(n)$    | Number of nodes (VMs) required by lease          |

B. Flow chart

The activity chart is used to explain the operations performed on scheduler in the proposed technique for quad-core multiprocessing LEAT processor in which the scheduler will check the CT & ST of a task.
Fig. 4: Flow Chart of task scheduling

Table 2: Configurations of core

| Frequency (MHz) | Utilization Factor (%) | No of cores in running state |
|-----------------|------------------------|-----------------------------|
| 100             | 0-7%                   | 1                           |
| 3               |                        |                             |
| 100             | 7-15                   | 2                           |
| 2               |                        |                             |
| 100             | 16-23                  | 3                           |
| 1               |                        |                             |
| 100             | 24-30                  | 4                           |
| 0               |                        |                             |

Copyright reserved © J. Mech. Cont. & Math. Sci.
Hamayun Khan et al

964
### Table-3: Workload Information

| Workload | No. of VMs (max (N)) | No. of leases | No. of Similar Leases | No. of leases Not Served |
|----------|----------------------|---------------|-----------------------|--------------------------|
| Workload 1 | 4                    | 5             | 2                     | 0                        |
| Workload 2 | 4                    | 7             | 4                     | 0                        |
| Workload 3 | 4                    | 10            | 4                     | 0                        |
| Workload 4 | 5                    | 18            | 8                     | 0                        |
| Workload 5 | 5                    | 27            | 12                    | 5                        |
| Workload 6 | 6                    | 30            | 15                    | 8                        |
| Workload 7 | 7                    | 36            | 18                    | 10                       |
| Workload 8 | 8                    | 40            | 16                    | 13                       |
| Workload 9 | 8                    | 50            | 14                    | 15                       |
| Workload 10 | 10                   | 55            | 15                    | 18                       |

### Table-4: Allocated Slots and Wastage Slots [slots are in minutes]

| Workload | Backfilling | EDF | GEDF | Backfilling with AHP | Proposed Mechanism |
|----------|-------------|-----|------|-----------------------|--------------------|
| Workload 1 | 54.54       | 54.54 | 69.00 | 69.00                | 69.00              |
| Workload 2 | 70.31       | 70.31 | 59.10 | 59.10                | 59.10              |
| Workload 3 | 68.18       | 68.18 | 80.26 | 80.26                | 80.26              |
| Workload 4 | 75.65       | 75.65 | 86.18 | 86.18                | 86.18              |
| Workload 5 | 98.81       | 98.81 | 72.69 | 72.69                | 72.69              |
| Workload 6 | 50.77       | 50.77 | 71.31 | 71.31                | 71.31              |
| Workload 7 | 50.68       | 50.68 | 79.40 | 79.40                | 79.40              |
| Workload 8 | 52.38       | 52.38 | 68.62 | 68.62                | 68.62              |
| Workload 9 | 55.20       | 55.20 | 64.64 | 64.64                | 64.64              |
| Workload 10 | 61.20      | 61.20 | 65.31 | 65.31                | 65.31              |

### Table-5: Comparison of Resource Utilization in

| Workload | T | Backfilling | EDF | GEDF | Backfilling with AHP | Proposed Mechanism |
|----------|---|-------------|-----|------|-----------------------|--------------------|
| Workload 1 | 550 | 300        | 250 | 300  | 300                   | 310                |
| Workload 2 | 580 | 460        | 120 | 460  | 120                   | 120                |
| Workload 3 | 800 | 600        | 240 | 600  | 240                   | 240                |
| Workload 4 | 1520 | 1150      | 370 | 1150 | 370                   | 370                |
| Workload 5 | 2140 | 1280      | 860 | 1280 | 860                   | 860                |
| Workload 6 | 2580 | 1310      | 1270 | 1310 | 1270                 | 1270               |
| Workload 7 | 2940 | 1400      | 1400 | 1400 | 1400                 | 1400               |
| Workload 8 | 3370 | 1870      | 1700 | 1910 | 1860                 | 1860               |
| Workload 9 | 4130 | 2280      | 1850 | 2280 | 1850                 | 1850               |
| Workload 10 | 4900 | 3040      | 1520 | 3110 | 1850                 | 1850               |

### C. XML File

---

Copyright reserved © J. Mech. Cont. & Math. Sci.

Hamayun Khan et al
<SIMULATION duration="150000">
  <SCHED className="storm.Schedulers.EDF_P_Scheduler"></SCHED>
  <CPUS>
    <CPU className="storm.Processors.LEATProcessor" name="CPU Core1" id="1" />
    <CPU className="storm.Processors.LEATProcessor" name="CPU Core2" id="2" />
    <CPU className="storm.Processors.LEATProcessor" name="CPU Core3" id="3" />
    <CPU className="storm.Processors.LEATProcessor" name="CPU Core4" id="4" />
  </CPUS>
  <TASK id="14" period="6" activationDate="4" deadline="6" WCET="2" priority="5" />
  <TASK id="15" period="9" activationDate="2" deadline="9" WCET="3" priority="6" />
  <TASK id="16" period="4" activationDate="4" />
</SIMULATION>

Fig.5: Overview of STORM simulator and operating procedure.

Copyright reserved © J. Mech. Cont. & Math. Sci.
Hamayun Khan et al
III. Experimental Results

In this section we discuss our experimental results that illustrates the temperature variation between the curves of proposed EDF and Global EDF. Proposed EDF considers reliability and performance parameter so only those cores are in running state that is in working condition. In the beginning exponentially temperature on chip rises and then arrive at a steady state level. At 17% utilization factor the global EDF has 4.3°C more temperature on chip as compare to our proposed approach. Proposed EDF based on core configurations and using quad-core processor.

![Fig.6: Frequency variation at core 1](image1)

![Fig.7: Frequency variation at core 2](image2)
Fig. 8: Frequency variation at core 3

Fig. 9: CPU core 1, 2, 3 Power Consumption at 17% UF
Fig. 10: Gantt diagram for CPUs under PEDF at 17% UF

Fig. 11: Comparison of core temperature to the max Threshold frequency
Table-5: Energy improvement of Proposed EDF and Global EDF

| Utilization Factor % | Running Cores | Energy of PEDF | No. of Cores | Energy of GEDF | Energy Improvement |
|----------------------|---------------|----------------|--------------|----------------|--------------------|
| 10%                  | 2             | 230.43 J       | 4            | 212.88 J       | 5.12 %             |
| 17%                  | 3             | 376.23 J       | 4            | 364.58 J       | 4.14 %             |
| 30%                  | 4             | 470.33 J       | 4            | 447.53 J       | 6.08               |

V. Conclusion and Future Work

With ever increasing processor speed and power densities in high speed performance systems, energy optimization and power management techniques have become an active research area. The paper summarizes the important research work done in the area. It discusses the principle and significance of Dynamic Voltage and Frequency Scaling (DVFS), Dynamic Power Management (DPM) and Dynamic Thermal Management (DTM) techniques and their extensions. Furthermore it highlights the trade-offs involved in designing and implementation of different techniques along with several practical applications and new challenges. Most of the current techniques do not consider the impacts of ambient temperature which can be investigated in future work. Furthermore, most of the existing algorithms address homogeneous multi-core systems that can be extended to heterogeneous platforms. Another area which is not well explained is thermal management of 3-D integrated circuits. Most of the current techniques are based on 2-D integrated circuits. For this new plan novel thermal management techniques needs to be investigated.

References

I. A. Coates, B. Huval, T. Wang, D. Wu, B. Catanzaro, N. Andrew, Deep learning with COTS HPC systems, in: Proceedings of the 30th International Conference on Machine Learning, 2013, pp. 1337–1345.

II. C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler And T. Keller, “Energy Management for Commercial Serv- ers,” In IEEE Computer, Vol. 36, Issue 12, pages 39 – 48, 2003.
III. C.-h. Hsu and W.-c. Feng, “A power-aware run-time system for high-performance computing,” in Proceedings of the 2005 ACM/IEEE Conference on Supercomputing. IEEE Computer Society, 2005.

IV. D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of Go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.

V. D. Konar, K. Sharma, V. Sarogi and S. Bhattacharyya, "A Multi-Objective Quantum-Inspired Genetic Algorithm (Mo-QIGA) for Real-Time Tasks Scheduling in Multiprocessor Environment", Procedia Computer Science, vol. 131, pp. 591-599, 2018.

VI. H. Khan, M. U. Hashmi, Z. Khan, R. Ahmad, and A. Saleem, “Performance Evaluation for Secure DES-Algorithm Based Authentication & Counter Measures for Internet Mobile Host Protocol,” IJCSNS Int. J. Comput. Sci. Netw. Secur. VOL.18 No.12, December 2018, vol. 18, no. 12, pp. 181–185, 2018.

VII. H. Khan, Q. Bashir, and M. U. Hashmi, “Scheduling based Energy Optimization Technique in multiprocessor Embedded Systems,” in 2018 International Conference on Engineering and Emerging Technologies (ICEET). doi:10.1109/iceet1.2018.8338643, 2018.

VIII. H. Khan, S. Ahmad, N. Saleem, M. U. Hashmi, and Q. Bashir, “Scheduling Based Dynamic Power Management Technique for offline Optimization of Energy in Multi Core Processors,” Int. J. Sci. Eng. Res. Vol. 9, Issue 12, December-2018, vol. 9, no. 12, pp. 6–10, 2018.

IX. H. Khan, M. U. Hashmi, Z. Khan, and R. Ahmad, “Offline Earliest Deadline first Scheduling based Technique for Optimization of Energy using STORM in Homogeneous Multi-core Systems,” IJCSNS Int. J. Comput. Sci. Netw. Secur. VOL.18 No.12, December 2018, vol. 18, no. 12, pp. 125–130, 2018.

X. M. Bohr, R. Chau, T. Ghani, and K. Mistry, “The High-Cool Solution,” IEEE Spectrum, vol. 44, no. 10, pp. 29-35, Oct. 2007.

XI. N. Fathima, “Website: www.ijirset.com Energy Aware Dynamic Slack Allocation for Multiprocessor System,” pp. 7476–7483, 2017.

XII. Q. Bashir, H. Khan, M. U. Hashmi, and S. Ali zamin, “A Survey on Scheduling Based Optimization Techniques in Multi-Processor Systems,” in Proceedings of the 3rd International Conference on Engineering & Emerging Technologies (ICEET), Superior University, Lahore, PK, 7-8 April, 2016., 2016.

XIII. R. Ayoub, S. Sharifi and T. Rosing, “GentleCool: Cooling Aware + pages 295 – 298, 2010.

XIV. S. Shi, Q. Wang, P. Xu, X. Chu, Benchmarking state-of-the-art deep learning software tools, in: Proceedings of the 7th IEEE International Conference on Cloud Computing and Big Data, Macau, China, 2016.
XV. S. Kaxiras, Z. Hu, and M. Martonosi, “Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power,” Proc. Int’l Symp. Computer Architecture (ISCA ’01), pp. 240-251, 2001.

XVI. S. Yang, M. Powell, B. Falsafi, K. Roy, and T. Vijaykumar, “An Integrated Circuit / Architecture Approach to Reducing Leakage in Submicron High-Performance-Caches,” Proc. Seventh Int’l Symp. High-Performance Computer Architecture (HPCA ’01), pp. 147-157, 2001.

XVII. S. Dutta, R. Jensen, and A. Rieckmann, “Viper: A Multiprocessor SOC for Advanced Set-Top Box and Digital TV Systems,” Design and Test of Computers vol. 18, pp. 21-31, 2001, IEEE.

XVIII. T. J. Semiconductor, B. Doyle, M. Group, and J. Corporation “Transistor Elements for 30 nm Physical Gate Length And Beyond,” Int’l Technology J., vol. 6, pp. 42-54, 2002.

XIX. T. Horvath, T. Abdelzaher, K. Skadron, and X. Liu, “Dynamic voltage scaling in multitier web servers.

XX. V. Shinde, “Comparison of Real Time Task Scheduling Algorithms,” vol. 158, no. 6, pp. 37–41, 2017.

XXI. Xu, J. & Parnas, D., “Scheduling Processes with Release Times, Deadlines, Precedence and Exclusion Relations,” IEEE Transactions on Software Engineering. Vol. 16(3), pp. 360-369, 1990.