IMPACT-2009

Load Balancing Issues in Automotives

Rajeshwari Hegde[1], K S Gurumurthy[2]

[1]*BMS College of Engineering, Email:rajeshwari.hegde@gmail.com*
[2]*UVCE, Bangalore, Email:drksgurumurthy@gmail.com*

***Abstract*—Electronic Control Units (ECUs) are widely used to improve the comfort and reliability of vehicles. It has become the fundamental building block of any automotive subsystem and is interfaced with electro mechanics counterpart. To meet the system wide requirements, these ECUs are interconnected using the communication infrastructure. Although the communication infrastructure in terms of, predominantly, the CAN based vehicle network took its birth to enable ECUs to work in a coordinated manner in order to support system wide requirements, during the past decade, this infrastructure was also viewed as a potential means to incorporate extensibility in terms of addition of newer ECUs which are built for implementing additional requirements. With this paradigm, the number of ECUs started growing in a steep manner, uncontrolled and as a result, today, it is not hard to see a high segment automotive housing ECUs as large as 75-80. Hence, load balancing mechanisms are needed to ease ECU integration and for efficient utilization of CPU power in ECUs. In this paper, we explain the concept of load balancing on the basis of CPU utilization across ECUs.**

Keywords: CPU, CAN, ECU, Load Balancing,

I. INTRODUCTION

It is forecast that 90% of the automotive innovations will be based on electronics development, and 80% of that amount will be based on software development, henceforth [1]. Looking back, this estimate is based on the growth of electronics content in the vehicle. The transformation that has occurred over the last two decades in the automotive industry from mechanically oriented companies to electronically or software oriented companies will continue to accelerate. Regarding the growing complexity of vehicle networks, this also becomes apparent. Since the first usage of the CAN protocol in production-vehicles in 1989 with three ECUs, the number of ECUs increased up to more than 70 ECUs. This was accompanied by an increasing degree of distribution of more and more complex functions. Today the various functions are distributed on many ECUs, due to increasing demands on comfort and reliability of a vehicle as well as on reduced emissions and fuel consumption. Thus the perception of a vehicle is increasingly determined by software. The ability of developing software cost-saving, fast and with high quality is therefore an essential prerequisite for successful innovations. In addition, the number of vehicle variants rose significantly during the last years. All automotive OEMS plan on further increasing this variety, shortening development time and time to market for new model, and reducing costs drastically. As automotive OEMs gained the capability to build complex ECUs and integrate them within automotives, the number of ECUs started growing in a steep manner, uncontrolled. This has resulted in the following setbacks:

1. Management of complex network of ECUs - a formidable task.
2. Proprietary nature of the ECU owners.
3. Overall cost of the ECUs and the associated infrastructure becoming a non- trivial fraction of the vehicle cost

Industry initiatives are addressing these setbacks and AUTOSAR (Automotive Open System Architecture) is a recent consortium, which is responsible for the standardization of subsystem design and implementation for future vehicle generations. AUTOSAR architecture, as a prime objective, inherently, features mechanisms to reduce the number of ECUs by exploiting the CPU power of the ECUs. Reduction in the number of ECUs reduces the overall cost, improves the reliability of the system, reduces complexity and exercises better controllability of the ECU market.

Moving forward, AUTOSAR compliant subsystems shall replace the existing subsystems in automotives. However, it is envisaged that, in future, any addition or refinement to the existing subsystems shall view the load sharing, load balancing, hot spots in terms of computational power, hot spots in terms of communications, etc. as mandatory performance requirements to be met. The paper is organized as follows. Section 2 explains the role of ECUs in Automotives. Section 3 gives an overview of load balancing; Section 4 looks at the load balancing concepts, Section 5 deals with the need for load balancing across ECUs; Section 6 explains an optimal dynamic load balancing Algorithm; Section 7 shows the simulation results. The paper is concluded in section 8.

II. ROLE OF ECUS IN AUTOMOTIVES

Importance of electronics is increasing day by day in modern automotive and the same has lead to proliferation of ECUs (Electronic Control Units) in it. The comfort, safety and performance of a modern car depends largely on the ECUs. An ECU also called a control unit, or control module, is an embedded system that controls one or more of the electrical systems or subsystems in a vehicle.The development of a new ECU is a joint effort of the car manufacturer and the supplier. The car manufacturer writes a specification of the desired functionality for which the software and hardware is delivered by the supplier.[12]. The development of ECUs is part of the development of a whole car. ECUs, the fundamental electronic building block of any automotive subsystem, used to be relatively simple, hardware-oriented systems. Today, they are multi-purpose, multi-chip computer systems where more functionality often is delivered in software than hardware. The most complex ECUs operate the powertrain. Simpler ones operate functions such as power windows, power seat, mirror adjustment system, etc. But even these ECUs need to be networked so that those specific features can be exploited both from the view of power management and such other critical co-ordinations or for enhancing the utility by way of personalization, etc. Having numerous subsystems like these, a medium to high complexity automotive contains as many as 70 ECUs which are interconnected by up to five different buses which themselves are connected by proper gateways. This trend of increasing automotive electronic content is the direct result of many new features that will greatly increase both safety and comfort but that will require more sophisticated ECUs with a large embedded software component. The safety features include steer-by-wire, brake-by-wire and drive-by-wire (collectively known as “X-by-wire”), automatic lane-following, drowsy driver detection, intelligent cruise control and airbag systems that can adjust deployment based upon passenger weight and the specific nature of an accident. Improving fuel efficiency is an important goal: hybrid (internal combustion and battery) and fuel-cell electric drive place high demands on software. Telematics and in-car entertainment will further increase the electronic content of cars, requiring the combination of such technologies as wireless connectivity, global positioning, digital radio and Internet access, all with hands-free voice activation whenever possible. [13]. For example, Antilock brake systems (ABS), which were introduced in 1978 and became widespread in the '80s, marked the first of the car-overrides-driver technologies and ABS laid the foundation for all the systems that have followed. Essentially, ABS works by using sensors to keep a central electronic control unit apprised of the rotating speed of each wheel. The processor regulates hydraulic pressure to apply maximum braking force without causing a skid. Building on that data and hardware, engineers increased the capabilities of the computer controls to create traction control in the '90s and, soon after, electronic stability control (ESC). By applying pressure to individual brakes, ESC keeps the vehicle from either over steering (fishtailing) or under steering (plowing) and keeps the vehicle stable. The next natural step in this progression was to equip cars with sensors and software that anticipate danger and react to it if the car decides that the driver is not doing enough to avoid an accident[14].

III. WHAT IS LOAD BALANCING?

Load-balancing, by definition, is dividing the amount of work that a computer has to do between one or more additional computers so that more work gets done in the same amount of time and, in general, all processing get done faster [4]. It is the assignment of work to processors and is critical in parallel simulations. It maximizes application performance by keeping processor idle time and interprocessor communication as low as possible. In applications with constant workloads, static load balancing can be used as a pre-processor to the computation. Other applications, such as adaptive finite element methods, have workloads that are unpredictable or change during the computation; such applications require dynamic load balancing that adjusts the decomposition as the computation proceeds. Numerous strategies for static and dynamic load balancing have been developed, including recursive bisection (RB) methods, space-filling curve based (SFC) partitioning and graph partitioning.

Consider the load balancing across ECUs. In static load balancing, the load on each ECU is known in advance. Hence the work is equally distributed across ECUs and no extra cost is required for balancing the load [6]. This can be explained using graph theory, where in, vertices represent individual ECU load and the edges represent the data dependencies [7].Dynamic Load Balancing across ECUs can improve the utilization of CPUs and the efficiency of parallel computations through migrating workload across CPUs at runtime. Workload migration can be carried out through transferring processes across nearest neighbor ECUs. Iterative strategies have become prominent in recent years because of the increasing popularity of point-to-point interconnection networks. [5]

There are many reasons to institute load balancing across ECUs. The two most popular are:

1.**Response time**- With two or more ECUs sharing the load, each of them will be running less of a load than 1 ECU alone, there by keeping the response time low.
2. **Redundancy**-If a load is balanced across 3 ECUs and one of them dies completely, then the other two can keep running and a vehicle will not even notice any downtime. Any load-balancing solution worth its salt will immediately stop trying to send traffic to the down ECU.
Usually, the load-balancing mechanism aim is to move the running tasks across the CPUs in order to insure that no CPU is idle while some tasks are waiting to be scheduled on other CPUs. It should minimize the total running time by a set of tasks. The characteristics of a load-balancing can be enumerated as follow:
_ information update policy: how to renew statistics about the entire system,
_ trigger policy: how to decide it is time to redistribute the tasks,

_ selection policy: method to select unbalanced nodes,
_ local designation policy: method to select the tasks that will move,
_ pairing policy: method to select the destination node for a given task.[16]

IV. LOAD BALANCING CONCEPTS

Modern automotive system contains as many as 75 ECUs which are interconnected by up to five different buses using proper gateways. Figure below shows the Load balancing across six ECUs on the basis of CPU utilization. For effective load balancing, status information and CPU utilization of each ECU have to be known. [15].

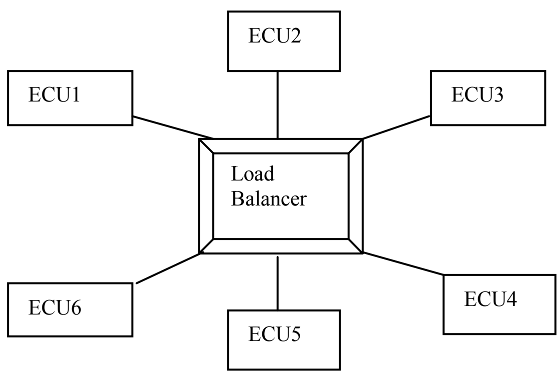

Figure1: Load balancing concepts

Load balancer attempts to ensure that loads are balanced across a group of ECUs. The figure below shows the Load balancer components. The detector detects and initiates a trigger when an overload or load imbalance occurs. Load Estimator estimates the load on overloaded and under loaded ECU and informs the mediator about the imbalance. The trigger from the detector tells the mediator to establish a load balancing session between the two entities, namely offloading ECU (ECU with the higher load doing the offloading) and the load-accepting ECU (ECU accepting load from the offloading ECU). Depending on which performance metric is to be balanced, one of the offloading algorithms is invoked. Finally, the mediator is invoked to establish the load balancing session.

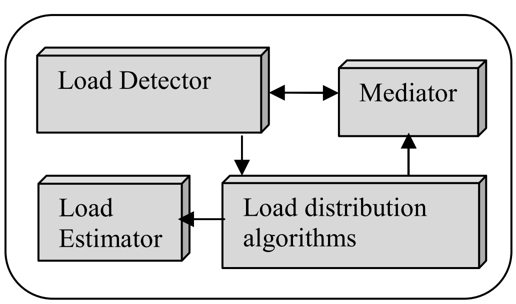

Figure2: Components of Load Balancer

V. NEED FOR LOAD BALANCING ACROSS ECUS

Distributed systems such as automotives can suffer from poor performance due to a bottleneck at overloaded ECUs. To address this performance bottleneck, an adaptive load balancing is used to distribute the load from densely loaded ECUs to scarcely loaded ECUs. Not much research has been done on keeping the load balanced across ECUs. To achieve good performance, it is essential to maintain a balanced work load among all the ECUs. Sometimes the load can be balanced statically. However, in many cases, the load on each ECU cannot be predicted a priory. Dispatching tasks from densely loaded ECUs to scarcely loaded ones to improve the overall performance of the vehicle is both logical and feasible.[3] A schedule of the work load that should be moved between any two ECUs , such that each ECU will have the same load on completion is a challenging task. One way to balance the load is to dispatch the job immediately upon arrival. The best load balancing status occurs when all ECUs are at the point of full utilization, without saturation. Each ECU's work load is proportional to its capacity. Allocating more jobs to a fully utilized ECUs might cause imbalance without improving the overall throughput. Since the data movement between ECUs incurs communication cost, the schedule should give balanced load with minimal data movement. Restricting the data movement to the neighboring ECUs might reduce communication cost. According to dimension Exchange Algorithm, the ECUs can be grouped in pairs and an ECU pair (a, b) with load l_a and l_b will exchange load, after which each will have the load $(l_a+ l_b)/2$.

VI. AN OPTIMAL DYNAMIC LOAD BALANCING ALGORITHM

Much of the load balancing problems can be described using terminology from graph theory. A graph G has two key components. The vertex set V and the edge set E. Let N be the number of ECUs. Let the ECU graph be represented by a graph(V,E), where V=(1, 2 ,3,…,N) is the set of vertices each representing an ECU and E is the set of edges. The graph is assumed to be connected. Two vertices i and j form an edge if ECUs i and j share a load. Associated with each ECU i is a scalar l_i representing the load on the ECU. The average load per ECU is

$$l_{avg}=\frac{\sum_{i=1}^{p} l_i}{p} \quad (1)$$

An edge (i,j) of the graph also has a scaler δ_{ij} associated, where δ_{ij} is the amount of load to be sent from ECU_i to ECU_j. The variables δ_{ij} are directional. i.e.

$$\delta_{ij} = -\delta_{ji} \qquad (2)$$

representing the fact that if ECU_i is to send the amount δ_{ij} to ECU_j, then ECU_j is to receive the same amount.
In reality, the workload will likely to be an integer number. A load balancing schedule should make the load on each ECU equal to the average load, that is

$$\sum_{(i,j)\in E} \delta_{ij} = l_i - l_{avg}, \; i=1,2,\cdots,N \qquad (3)$$

If N-1 equations of (3) are satisfied, the remaining one equation will be satisfied automatically.
A number of trends in computational science and engineering have increased the need for effective dynamic load balancing techniques.[1]

VII. SIMULATION RESULTS

The load on each ECU before and after load balancing is shown in figure 3.

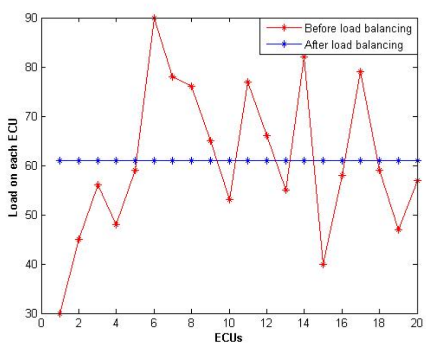

Figure 3: Load on ECUs before and after load balancing.

The graph of the amount of load to be sent from ECU i to ECU j is shown in figure4.

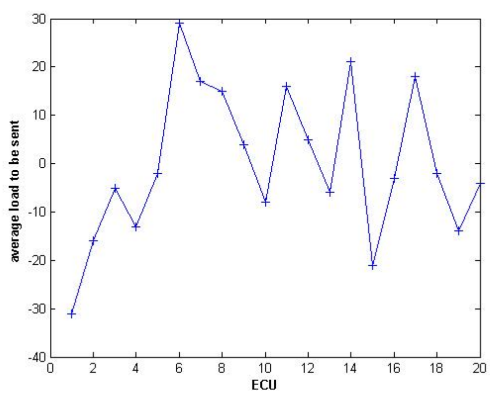

Figure 4: Amount of load sent from ECU i to ECU j.

VIII. CONCLUSION

In this paper, an attempt has been made to introduce the concept of load balancing across ECUs in automotives on the basis of CPU utilization. The load balancing approach reduces the complexity of the automotive system by equally distributing the load across different ECUs. Load balancing mechanisms in automotives ease the ECU integration.

REFERENCES

[1]. Jerrell Watts and Stephen Taylor, “A Practical Approach to Dynamic Load Balancing”, caltechcstr.library.caltech.edu/242/00/95

[2] Y F Hu, R J Blake, “An optimal dynamic load balancing Algorithm”, citeseer.ist.psu.edu/121199.html, 1995

[3]Chi-Chung Hui, Samuel T Chanson, “Improved strategies for dynamic load balancing”, IEEE Concurrency,1999

[4]http://webhosting.devshed.com/c/a/Web-Hosting-Articles/What-is-Load-balancing-and-Do-I-Need-It/

[5] Cheng-Zhong XU amd Francis C.M. Lau, “Iterative Dynamic load balancing in multicomputers”, Journal of Operation Research Society, Vol. 45, No. 7, July 1994, pp,786-796

[6].http://www.hpc2n.umu.se/events/courses/slides/programming/tsld021.htm

[7]. Can Static Load Balancing Algorithms Be Appropriate in a Dynamic Setting? http://www.dl.ac.uk/TCSC/Staff/Hu_Y_F/MEETING/TALKS/hendrickson.ps.gz.

[8]. Alex K. Y. Cheung and Hans-Arno Jacobsen , Dynamic Load Balancing in Distributed Content-based Publish/Subscribe, ACM middleware 2006.

[9] Richard Anthony, Alexander Leonhardi, Cecilia Ekelin, Dejiu Chen, Martin Törngren, Gerrit de Boer, Isabell Jahnich, Simon Burton, Ola Redell, Alexander Weber, Vasco Vollmer, “A Future Dynamically Reconfigurable Automotive Software System”, www.md.kth.se/RTC/ARTIST2/publications/ATT00014.pdf.

[10] Richard Anthony, Achim Rettberg, Dejiu Chen, Isabell Jahnich, Gerrit de Boer, Cecilia Ekelin, “Towards a dynamically Reconfigurable automotive control System architecture”, www.hni.uni-paderborn.de/publikation/

[11] Guoqing Yang, Minde Zhao, Hongli, Zhaohui Wu, “ SmartOSEK Based Design and Verification for In-vehicle Network System: A Model-Based Approach”, International Conference on Control, Automation, Robotics and Vision, IEEE 2006.

[12] Florian Wohlgemuth, Peter HoJinann,” Automotive System Design: Today And Tomorrow”,IEEE 1998.

[13] B.S.Nagabhushana, Rajeshwari Hegde, Veena Hegde,” Avenues and Technologies for Infotronics in Automotive Industry” National Conference on ICSA, Chennai, 2006.

[14].http://www.popularmechanics.com/blogs/automotive_news/4202639.html

[15].Isabell Jahnich, Ina Podolski, Achim Rettberg, “Integrated Dynamic Load Balancing Strategies into the Car-Network, 4th IEEE International Symposium on Electronic Design, Test & Applications. IEEE2008.

[16] Eric PIEL, Philippe MARQUET, Julien SOULA, Jean-Luc DEKEYSER, “Load-Balancing for a Real-Time System Based on Asymmetric Multi-Processing” LIFL Report # 2004-06, April 2004