ISSN Application No. 21650

Journal of Cloud Computing and Data Base Management

Volume No. 8

Issue No. 3

September - December 2023

ENRICHED PUBLICATIONS PVT. LTD

S-9, IInd FLOOR, MLU POCKET,
MANISH ABHINAV PLAZA-II, ABOVE FEDERAL BANK,
PLOT NO-5, SECTOR-5, DWARKA, NEW DELHI, INDIA-110075,
PHONE: - + (91)-(11)-47026006

ISSN Application No. 21650

Journal of Cloud Computing and Data Base Management

Aims and Scope

Journal of Cloud Computing and Database Management is a peerreviewed Print + Online journal of Enriched Publications to disseminate the ideas and research findings related to all sub-areas of Computer Science and IT. It also intends to promote interdisciplinary researches and studies in Computer Science and especially database management and cloud computing maintaining the standard of scientific excellence. This journal provides the platform to the scholars, researchers, and PHD Guides and Students from India and abroad to adduce and discuss current issues in the field of Computer Sciences.

ISSN Application No. 21650

Journal of Cloud Computing and Data Base Management

(Volume No. 8, Issue No. 3, September - December 2023)

Contents

A Novel Approach Design in Interference Finding System for Data Mining using Madam Id

R. Venkatesan[*], R. Ganesan[**], A. Arul Lawrence Selvakumar[***]
[*]Research Scholar/CS, CMJ University, Shillong, Meghalaya
[**]Professor, Dept of E& I Engineering, N I C E, Kumaracoil, Tamil Nadu
[***]Professor & Head, Dept of CSE, Rajiv Gandhi Institute of Technology, Bangalore, Karnataka

A B S T R A C T

Intrusions are the activities that violate the security policy of system. Intrusion Detection is the process used to identify intrusions. Network security is to be considered as a major issue in recent years, since the computer network keeps on extending dramatically. Information Systems and Networks are subject to electronic attacks and the possibilities of intrusion are very high. In order to protect the networking system, it is mandatory to update and install Intrusion Detection System (IDS) due to the expansion of network dramatically every day along with new threats and attack. We do know current IDSs are constructed with interception of Data Mining techniques and Intrusion Detection. We also used the Data Mining techniques to design this interference finding system. The objective of this paper is state our novel approach design in Intrusion Detection for Data Mining using MADAM ID. We have analyzed this novel approach using Network Flight Recorder (NFR).

Keywords: Data Mining, Intrusion Detection System (IDS), Network Security, Misuse Detection, Anomaly Detection, Classification, Clustering, MADAM ID, NFR

1. INTRODUCTION:

In recent years many researchers are focusing to use Data Mining concepts for Intrusion Detection. Data mining is a process to extract the implicit information and knowledge which is potentially useful and people do not know in advance, and this extraction is from the huge data. In the other hand intrusions in an information system are the activities that violate the security policy of the system, and intrusion detection is the process used to identify intrusions. The objective of this paper is state our novel approach design in Intrusion Detection for Data Mining using MADAM ID in NFR (Network Flight Recorder Inc., 1997)[1], a system includes packet capturing engine and N-Code programming support for specific packet filtering logic. This is an offline evaluation, but an effective ID should be in real-time to satisfy the security policy an organization.

2. INTRUSION DETECTION TECHNIQUES:

The intrusion detection techniques based upon data mining [2], [3] are generally falls into one of two categories: anomaly detection and misuse detection. The signatures of some attacks are known, whereas other attacks only reflect some deviation from normal patterns.

2.1 Anomaly Detection

Anomaly detection attempts to determine whether deviation from an established normal behavior profile can be flagged as an intrusion [4]. Anomaly detection consists of first establishing the normal behavior profiles for users, programs, or other resources of interest in a system, and observing the actual activities as reported in the audit data to ultimately detect any significant deviations from these profiles. Most anomaly detection approaches are statistical in nature.

2.2 Misuse Detection

Misuse detection works by searching for the traces or patterns of well-known attacks. Lee et al. [5] designed a signature-based database intrusion detection system (DIDS) which detects intrusions by matching new SQL statements against a known set of transaction fingerprints.

Misuse detection is considered complementary to anomaly detection.

2.3 Pros and Cons of Anomaly and Misuse Detection

Table 1: Pros and Cons of Anomaly Detection and Misuse Detection

Technique	Pros	Cons
Anomaly Detection	Is able to detect unknown attacks based on audits	High false-alarm and limited by training data.
Misuse Detection	Accurately and generate much fewer false alarm	Cannot detect novel or unknown attacks

2.4 Drawbacks of current IDS

Intrusion Detection Systems (IDS) has become a standard component in security infrastructures as they allow network administrators to detect any violations. These security violations range from external attackers trying to gain unauthorized access to insiders abusing their access. Current IDS have a number of significant drawbacks [6]:

- **False Positives** – A common complaint is the amount of false an IDS will generate.

- **False Negatives** – In this case, IDS does not create a signature or alarm, when an intrusion is actually happened.

- **Data Overload** – In this aspect, Misuse Detection cannot be related directly, however, it is very important to analyze how much data an analyst can efficiently and effectively analyze.

2.5 Need of using data mining approaches in IDS

A team in Minnesota University (1990) recognized the need for existence of standardized dataset to train IDS tool. Minnesota Intrusion Detection System (MINDS) combines signature based tool with data mining techniques. Signature based tool (Snort - freeware) are used for misuse detection & data mining for anomaly detection. The reasons for using Data Mining approaches in IDS are:

1. It is very difficult to build IDS using programming languages, which requires more explicit data and functional knowledge.
2. The reliability, compatibility and dynamic nature of machine-learning make it a suitable solution for this situation.
3. The environment of an IDS and its classification task highly depend on user-driven preferences.

3. DATA MINING APPROACHES

Data mining generally refers to the process of (automatically) extracting models from large stores of data [7]. The recent rapid development in data mining has made available a wide variety of algorithms, drawn from the fields of statistics, pattern recognition, machine learning, and database. There are several types of algorithms [7] which are particularly related to intrusion detection.

• **Classification:** classifies a data item into one of several pre-defined categories. These algorithms normally output "classifiers". An ideal application in intrusion detection would be to gather sufficient "normal" and "abnormal" audit data for a user or a program, then apply a Classification algorithm to learn a classifier that can label or predict new unseen audit data as belonging to the normal class or the abnormal class.

• **Link analysis:** determines relations between fields in the data base records. Correlations of system features in audit data, for example, the correlation between command and argument in the shell command history data of a user, can serve as the basis for constructing normal usage profiles.

• **Sequence analysis:** models sequential patterns. These algorithms can discover what time-based sequences of audit events are frequently occurring together. These frequent event patterns provide guidelines for incorporating temporal and statistical measures into intrusion detection models.

3.1. Systematic Framework

Our framework consists of data-mining programs for learning detections models, a translator for converting learned rules to real-time models, and NFR for capturing network traffic and applying the real-time N-code modules for ID. A framework has been developed, first proposed in [4], of applying data mining techniques to build intrusion detection models. This framework consists of programs for learning classifiers as well as a support environment that enables system builders to interactively and iteratively drive the process of constructing and evaluating improved detection models. The end product of this process is a set of concise and intuitive rules (that can be easily inspected and edited by security experts when needed) that can detect intrusions. The rules are then subsequently ported over to N-code as sub-routines or independent functions.

First, the network security expert needs to analyze and categorize attack scenarios and system vulnerabilities, and then we need to code the corresponding rules and patterns manually in N- code for misuse detection. In this manual development process, current IDSs including NFR have limited extensibility and adaptability. Our aim is to develop IDS which should be substantial to reduce this effort by automating:

1) The task of building intrusion detection through data mining.
2) Generating the N-code for NFR to detect intrusions via a machine translator.

3.2 Mining Data to Construct Attributes

In order to mine the data, first we must process and summarize packet-level network traffic data into "connection" records. We initially start out with the raw audit data (commonly tcpdump binary output) of the designated network we wish to monitor. This is then subsequently pre processed into individual packets/events in the ASCII format. As the packets are summarized according to their separate connections, we record their within connection features which may be deemed as "traditional attributes" of a connection record. We use the mined patterns from network connection records as guidelines to construct temporal statistical attributes for building classification models [9]. We performed pattern mining and comparisons using intrusion data of several well known attacks e.g., port-scan, ping-sweep, etc., as well normal connection records. Each of the unique intrusion patterns are used as guidelines for adding additional features into the connection records to build better classification models.

3.3 Learning Detection Rules

We apply RIPPER [10] to the connection records to generate the classification rules for the intrusions. Like other rule learning systems, this method is used for classifications problems. "count " the count of such connections rej count the count of connections that get the flag "REJ" met by a particular host S01 count the count of connections that send a SYN packet but never get the ACK packet (S0), or receive an ACK on SYN that they never have sent (s1) diff services the count of unique (different) services diff srv rate diff services / count...

A "training" period is initially required for RIPPER to gather the necessary data on the network to compute models. The purpose is two-fold:

1) Establishing "normal" traffic patterns and variants that the network may encounter to establish anomaly detection,
2) Introducing known intrusion methods and attack scripts into the network in order to inductively learn the classification models of intrusions.

An example rule has been given here for better understanding and it is used to detect known attacks. In particular, when we illustrate how to detect and recognize an attack which is categorized as denial-of service.

Rule Translation

A detection rule, for example,
pod :- wrong_fragment >= 1, protocol_type = icmp. can be automatically converted into the following N-code:

```
filter pod ()
{
if (wrong_fragment() > 1 & protocol_type () == icmp) alarm_pod ();
}
```

As long as the features, i.e., wrong_fragment and protocol type, have been implemented as N-code filter functions.

3.4. Network Flight Recorder (NFR)

The Network Flight Recorder (NFR) [1] is one such extensible system that combines data collection, analysis, and storage within a single platform. IDS would normally be located between a firewall and an Internet connection, an area aptly named the DMZ(De-Militarized Zone) . This offline analysis in NFR is accomplished by scripts based on a language called N-code, NFR's flexible language for traffic analysis. Information is displayed in NFR will be transferred to a Web-based interface with the Java support. NFR also has a real time alerting capability and a storage subsystem that allows data to be stored and transferred to other external devices [1]. We use the frequent episodes algorithms [8] for this analysis.

3.5 Generation of N-Code Filters

The attributes of connection records are implemented as subroutines that may be called upon to check the rules that were generated by machine learning. A RIPPER rule simply consists of a sequence of attribute value tests, with each attribute implemented as an N-code filter, a rule can be automatically translated into an N-code filter that consists of a sequence of unction calls to the N-code filters.

4. EFFICIENT EXECUTIONS OF LEARNED RULES

Our plan is to implement all the required features used in the RIPPER rule set as N-code "feature filters", and implement a translator that can automatically translate each RIPPER rule into an N-code "rule filter". Although often ignored in off-line analysis, efficiency is a very important consideration in real-time intrusion detection. In our first experimental implementation of N-code "rule filters", we essentially tried to follow the off-line analysis steps in a real-time environment. A connection is not inspected (i.e., classified using the rules) until its connection record is completely formulated, that is, all packets of the connection have arrived and summarized, and all the temporal and statistical features are computed. This scheme failed miserably. When there is a large volume of network traffic, the amount of time taken to process the connection records within the past 2 seconds and calculate the statistics is also very large. During, the execution many connections may have terminated (and thus completed with attack actions) when the current connection is finally inspected by the RIPPER rules. That is, the detection of intrusions is severely delayed. Ironically, DOS attacks, which typically generate a large amount a traffic in a very short period time, are often used by intruders tofirst overload an IDS, and use the detection delay as a window of opportunity to quickly perform their malicious intent. For example, they can seize control of the operating system and "kill" the IDS. MADAM ID can be used to learn the site-specific intrusion detection rules using the locally gathered audit data, and be automatically converted in to N-code "rule filters".

5. CONCLUSION

In this research paper, we studied the problem of how to automatically construct features from the mined patterns. We have applied relevant algorithms to construct the required model and also open the scope for further research in the areas of Misuse detection and IDS management in networking environment (large scale). The main challenge is how to efficiently execute the rules in a real-time & large-scale networking environment.

REFERENCES

[1] Inc. Network Flight Recorder. Network flight recorder. http://www.nfr.net, 1997.

[2] Daniel Barbara, Ningning Wu and Sushil Jajodia Detecting novel network intrusion using bayes estimators. In Proceedings of First SIAM Conference on data mining Chicago, 2001.

[3] Eric Bloedorn, Alan D. Christiansen, William Hill, Clement Skorupka, Lisa M. Talbot, and Jonathan Tivel. Data mining for network intrusion detection: How to get started.

[4] W. Lee and S. J. Stolfo. Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, January 1998.

[5] S.Y. Lee, W. L. Low and P. Y. Wong, "Learning Fingerprints for a Database Intrusion Detection System", In Proceedings of the 7th European Symposium on Research in Computer Security, Pages 264-280, 2002.

[6] (SANS: FAQ: Data Mining in Intrusion Detection) http://www.sans.org/security-resources/idfaq/data_mining.php

[7] W. Lee, S.J. Stolfo, K.W. Mok, Algorithms for Mining System Audit Data, in Proc. KDD, 1999.

[8] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences. In Proceedings of the 1st International Conference on Knowledge Discovery in Databases and Data Mining,

[9] W. Lee and S. J. Stolfo. A data mining framework for building intrusion detection models. In 1999 IEEE Symposium on Security and Privacy., Oakland, CA, May 1999.

[10] W. W. Cohen. Fast effective rule induction. In Machine Learning: the 12th International Conference, Lake Taho, CA, 1995. Morgan Kaufmann.

An Efficient Low Complexity MMSE for OFDM Systems

Pavana Shree B.K*
*Department of Electronics and Communication,
V.T.U University, Bangalore

ABSTRACT

OFDM is becoming the technique of choice for many high data rate wireless applications. OFDM systems are multi carrier systems in which the carriers are spaced as closely as possible while maintaining orthogonality, thereby efficiently using available spectrum. This technique is very use full in frequency selective channels, since it allows a single high rate data stream to be converted in to a group of many low rate data stream, each of which can be transmitted without inter symbol interference. Researchers are working at a greater extent on wireless applications. As per the literature many channel estimation technique have been proposed by the researchers.

The present work aims at the performance of channel estimation methods are investigated by MATLAB simulation. A novel MMSE channel estimation scheme is proposed and its performance is numerically confirmed for the OFDM system proposed in the IEEE 802.16 standards.

Keywords: OFDM, channel estimation, frequency-selective fading channel.

INTRODUCTION

OFDM technology is a popular technique for transmission of signals over wireless channels, due to its many advantages such as the high spectral efficiency, robustness to frequency selective fading, and the feasibility of low-cost transceiver implementations [6]. For wideband wireless communication, it is necessary to dynamically estimate the channel before demodulating the signals. There are two kind methods for channel estimation. The first is the pilot assisted estimation, the pilot signals are embedded in certain sub-carriers of each OFDM symbol [4].

At the receiver, the channel components estimated using these pilots are interpolated for estimating the complete channel. Based on the criterion of realization, it can be classified as Least Square (LS), MMSE, and maximum likelihood estimator and so on [1].The second category is the Blind channel estimation. The blind schemes avoid the use of pilots, for achieving high spectral sufficiency. This is achieved at the cost of higher implementation complexity and some amount of performance loss [5].Now propose a low complexity MMSE estimation scheme which can reduce computational complexity but cause little attenuation of performance.

LITURATURE SURVEY

1. A new derivation of least-squares-fitting principle for OFDM channel estimation.

Many channel estimation and data detection algorithms of the orthogonal frequency division multiplexing (OFDM) system have been proposed. Some of these algorithms are based on the principle of linear minimum mean-square error (LMMSE) estimation, which is theoretically optimal. There are also some algorithms developed based on the least-squares- fitting (LSF) principle, which finds a regression polynomial to it a block of tentative channel estimates in the least-squares sense. The LSF principle is a non-statistical approach, while the LMMSE algorithm is statistical

to known or estimate the channel statistics like correlation matrices and signal-to-noise ratio (SNR). This letter proposes a novel viewpoint of the LSF principle. We show that the non-statistical LSF principle can be derived alternatively from the statistical LMMSE principle by eigenvector approximation. This constructs a link between these two principles. The mean-square estimation error (MSEE) analysis shows that there are common terms in the MSEE expressions of these two principles. This further validates theconstructed link. Based on the derived link and MSEE analysis, we also give some characteristics and discussions of the LSF principle.

2. Robust channel estimation for OFDM systems with rapid dispersive fading channels.

Orthogonal frequency division multiplexing (OFDM) modulation is a promising technique for achieving the high-bit-rates required for a wireless multimedia service. Without channel estimation and tracking OFDM systems have to use differential phase-shift keying (DPSK), which has a 3 dB signal-to-noise ratio (SNR) loss compared with coherent phase-shift keying (PSK). To improve the performance of OFDM systems by using coherent PSK, we investigate robust channel estimation for OFDM systems. We derive a minimum mean-square-error (MSE) channel estimator, which makes full use of the time- and frequency-domain correlations of the frequency response of time-varying dispersive fading channels. Since the channel statistics are usually unknown, we also analyze the mismatch of the estimator to channel statistics and propose a robust channel estimator that is insensitive to the channel statistics. The robust channel estimator can significantly improve the performance of OFDM systems in a rapid dispersive fading channel.

3. Low complexity channel estimation for space-time coded wideband OFDM systems.

In this article, channel estimation for space-time coded orthogonal-frequency division multiplexing (OFDM) systems is considered. By assuming that the channel frequency response is quasi-static over two consecutive OFDM symbols, we develop channel parameter estimators based on the use of space-time block coded (STBC) training blocks. Using an STBC training pattern, a low-rank Wiener filter-based channel estimator with a significant complexity reduction is proposed. A simplified approach for the optimal low-rank estimator is also proposed to further reduce the estimator complexity while retaining accurate frequency domain channel estimation. Numerical results are provided to demonstrate the performance of the proposed low complexity channel estimators for space-time trellis coded OFDM systems.

4. Blind adaptive channel estimation in OFDM systems.

Consider the problem of blind channel estimation in zero padding OFDM systems and propose blind adaptive algorithms in order to identify the impulse response of the multipath channel. In particular, we develop RLS and LMS schemes that exhibit rapid convergence combined with low computational complexity and numerical stability. Both versions are obtained by properly modifying the orthogonal iteration method used in numerical analysis for the computation of singular vectors. With a number ofsimulation experiments we demonstrate the satisfactory performance of our adaptive schemes under diverse signaling conditions.

In July 1998 O. Edfors, M. Sandell, and J.-J. van de Beek,published “OFDM channel estimation by singular value decomposition,” In this paper we present and analyse low rank channel estimators for OFDM systems using the frequency correlation of the channel. Low rank approximation based on the discrete Fourier transform (DCT) have been proposed, but these suffer from poor performance when the channel is not sample spaced. We apply the theory of optimal rank reduction to linear minimum mean squared error(LMMSE) estimators and show that these estimators are robust.

In July 1998 Y. Li, L. J. Cimini, Jr., and N. R. Sollenberger did work on "Robust channel Estimation for OFDM systems with rapid dispersive fading channels," In this paper they proposed robust channel estimation approach to estimate the channel fading in time domain. The estimation criterion is to minimize the worst possible amplification of the estimation errors in terms of the exogenous input disturbances such as multiplicative and additive noise. The criterion is different from the traditional minimum estimation error variance criterion for the Kalman estimation algorithm, and requires no a priori knowledge of the disturbance statistics.

In Dec. 2001 M. Morelli and U. Mengali, published paper on "A comparison of pilot-aided channel estimation methods for OFDM systems, In this paper, the channel estimation methods for OFDM systems based on comb-type pilot sub-carrier arrangement are investigated.

The channel estimation algorithm based on comb type pilots is divided into pilot signal estimation and channel interpolation. The pilot signal estimation based on LS or MMSE criteria, together with channel interpolation based on piecewise-linear interpolation or piecewise second-order polynomial interpolation is studied.

In Sep. 2003Y Gong and K. Ben Letaief, published paper on "Low complexity channel estimation for space-time coded wideband OFDM systems," The paper proposes a computationally efficient, pilot aided linear minimum mean-square-error (MMSE) time- domain channel estimation algorithm for OFDM systems with transmitter diversity in unknown wireless fading channels. The proposed approach employs a convenient representation of the channelimpulse responses based on the Karhunen-Loeve (KL) orthogonal expansion and finds MMSE estimates of the uncorrelated KL series expansion coefficients. Based on such an expansion, no matrix inversion is required in the proposedMMSE estimator. Subsequently, optimal rank reduction is applied to obtain significant taps resulting in a smaller computational load on the proposed estimation algorithm.

In 2004 R. Prasad the author and proposed a paper called OFDM for Wireless Communications Systems. In his paper has explained how OFDM technique in wireless communication systems. He proposed the block diagram for OFDM system.

In July 2006 X. G. Doukopoulos and G. V. Moustakides, published paper on "Blind adaptive channel estimation in OFDM systems," The problem of blind channel estimation in zero padding OFDM systems, and propose blind adaptive algorithms in order to identify the impulse response of the multipath channel. In particular, we develop RLS and LMS schemes that exhibit rapid convergence combined with low computational complexity and numerical stability. Both versions are obtained by properly modifying the orthogonal iteration method used in numerical analysis for the computation of singular vectors. With a number of simulation experiments we demonstrate the satisfactory performance of our adaptive schemes under diverse signaling conditions.

METHODOLOGY

The modulation & demodulation method of pilot symbols and data symbols is QPSK. In order to reduce the complexity of MMSE channel estimation proposed to use an algorithm called singular value decomposition (SVD), it will inevitably cause attenuation of performance. In order to reduce the complexity of MMSE channel estimation with little or no attenuation, we use new method called diagonal matrix.

INDENTATIONS AND EQUATIONS

There are two kind methods for channel estimation. The first is the pilot assisted estimation, the pilot signals are embedded in certain sub-carriers of each OFDM symbol. At the receiver, the channel components estimated using these pilots are interpolated for estimating the complete channel. Based on the criterion of realization, it can be classified as Least Square (LS), MMSE, and maximum likelihood estimator and so on. The second category is the blind channel estimation. The blind schemes avoid the use of pilots, for achieving high spectral sufficiency. This is achieved at the cost of higher implementation complexity and some amount of performance loss.

Propose a low complexity MMSE estimation scheme which can reduce computational complexity but cause little attenuation of performance. The final scheme shows to be efficient according to our extensive computer simulations.MSE and BER performance of channel estimation methods are investigated by MATLAB simulation. The MATLAB version used isMATLAB 2011. The modulation & demodulation method of pilot symbols and data symbols is QPSK.

The simulation results demonstrate the effectiveness of the proposed scheme. The MSE of channel estimation is defined by 2

The computational complexity reduction ratio (CCRR) of the proposed MMSE over the conventional scheme is defined as

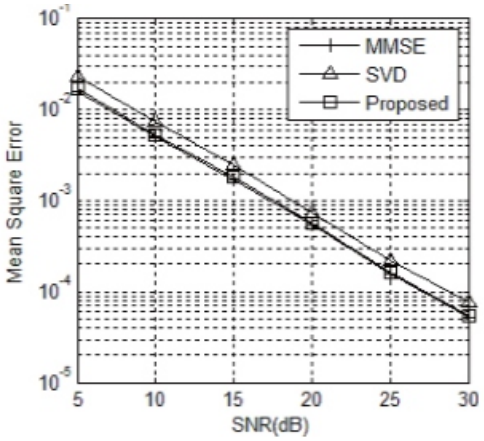
$$CCRR = 1 - \frac{\text{Complexity of the proposed scheme}}{\text{Complexity of the or iginal scheme}} \times 100(\quad)$$

The results will show that as compared with conventional MMSE, using this scheme can reduce the computational burden.

SIGNIFICANCE OF THE STUDY

Since this scheme can reduce the computational burden and suffer little attenuation of performances. Therefore, it is rather attractive for practical application in OFDM-based communication systems. OFDM technology is a popular technique for transmission of signals over wireless channels, due to its many advantages such as the high spectral efficiency robustness to frequency selective fading, and the feasibility of low-cost transceiver implementation.

FIGURES AND TABLES



MSE vs. SNR of three channel estimation methods

Computational complexity of the proposed mmse and the Conventional scheme when N=128

	Conventional	Proposed	CCRR
multiplications	81,920	33,408	59.21%
additions	16,384	128	99.22%

CONCLUSION

In this paper, a novel MMSE channel estimation scheme is proposed and its performance is numerically confirmed for the OFDM system proposed in the IEEE 802.16 standard. The results show that as compared with conventional MMSE, using this scheme can reduce the computational burden and suffer little attenuation of performances. Therefore, it is rather attractive for practical application in OFDM-based communication systems.

REFERENCES

[1] M.-X. Chang, "A new derivation of least-squares-fitting principle for OFDM channel estimation," IEEE Tr 2009

[2] Y. Li, L. J. Cimini, Jr., and N. R. Sollenberger, "Robust channel estimation for OFDM systems with rapid dispersive fading channels," IEEE Transactions on Commun., vol. 46, no. 7, pp. 902–914, July 1998.

[3] Y. Gong and K. Ben Letaief, "Low complexity channel estimation for space-time coded wideband OFDM systems," IEEE Trans. Wireless Commun., Sep. 2003.

[4] M. Morelli and U. Mengali, "A comparison of pilot-aided channel estimation methods for OFDM systems," IEEE Trans. Signal Process., Dec. 2001.

[5] X. G. Doukopoulos and G. V. Moustakides, "Blind adaptive channel estimation in OFDM systems," IEEE Trans. Wireless Commun., July 2006.

[6] O. Edfors, M. Sandell, and J.-J. van de Beek, "OFDM channel estimation by singual value decomposition," IEEE Trans. Commun., vol. 46, no. 7,pp. 931–939, July 1998.

[7] Manwinder Singh , "Block based Channel Estimation Algorithms for OFDMIEEE 802.16e (Mobile WiMAX) System" 2011 january.

[8] Ming Lei, Ismail Lakkis , Hiroshi Harada, "MMSE-FDE based on Estimated SNR for Single-Carrier Block Transmission (SCBT) in Multi-Gbps WPAN (IEEE 802.15.3c)"2008.

Query Recommendation Approach for Searching Database using Search Engine

Geetanjli Gaur*, Ashish Oberoi*, Mamta Oberoi**

*Department of Computer Engineering, M. M. University, Mullana, Ambala, Haryana, India

**Department of Statistics, MLN College, Yamuna Nagar, Haryana, India

A B S T R A C T

Search Engines generally return long lists of ranked pages, finding the desired information content from which is typical on the user end and therefore, Search Result Optimization techniques come into play. The proposed system based on learning from query logs predicts user information needs and reduces the seek time of the user within the search result list.

To achieve this, the method first mines the logs using a similarity function to perform query clustering and then discovers the sequential order of clicked URLs in each cluster . Finally, search result list is optimized by re-ranking the pages. The proposed system proves to be efficient as the user desired relevant pages occupy their places earlier in the result list and thus reducing the search space. This thesis also presents a query recommendation scheme towards better information retrieval.

Keywords: World Wide Web (WWW), Information Retrieval, Search Engine, Query processing

1. INTRODUCTION / LITERATURE SURVEY

The Query Recommendation provides an excellent opportunity for gaining insight into how a search engine is used and what the users' interests are. Query Logs prove to be important information repositories to keep track of user activities through the search results, knowledge about which can improve the performance of a search engine. In spite of the recent advances in the Web search engine technologies; there are still many situations, in which user is presented with undesired and non-relevant pages in the top most results of the ranked list. One of the major reasons for this problem is the lack of user knowledge in framing queries. Moreover, search engines often have difficulties in forming a concise and precise representation of the response pages corresponding to a user query. Nowadays, providing a set of web pages based on user query words is not a big problem in search engines. Instead, the problem arises at the user end as he has to sift through the long result list, to find his desired content. This problem is referred to as the Information Overkill problem. Search engines must have a mechanism to find the users' interests with respect to their queries and then optimize the results correspondingly. To achieve this, query log files maintained by the search engines play an important role. The logs provide an excellent opportunity for gaining insight into how a search engine is used and what the users' interests are.

The goal of this approach describes the various terms & approaches that are used in optimization of web search and provide an overview of framework used in the field of web mining

1.1 TERMS USED IN WEB MINING

Various terms that are commonly used in web mining are:

QUERY LOGS

The log keeps users' queries and their clicks, as well as their browsing activities. In the context of search engines, servers record an entry in the log for every single access they get corresponding to a query. The typical logs search engines include the following entries:

1) User (session) IDs,
2) Query q issued by the user
3) URL u accessed/clicked by the user
4) Rank r of the URL u clicked for the query q and
5) Time t at which the query has been submitted.

2. PROPOSED MODEL

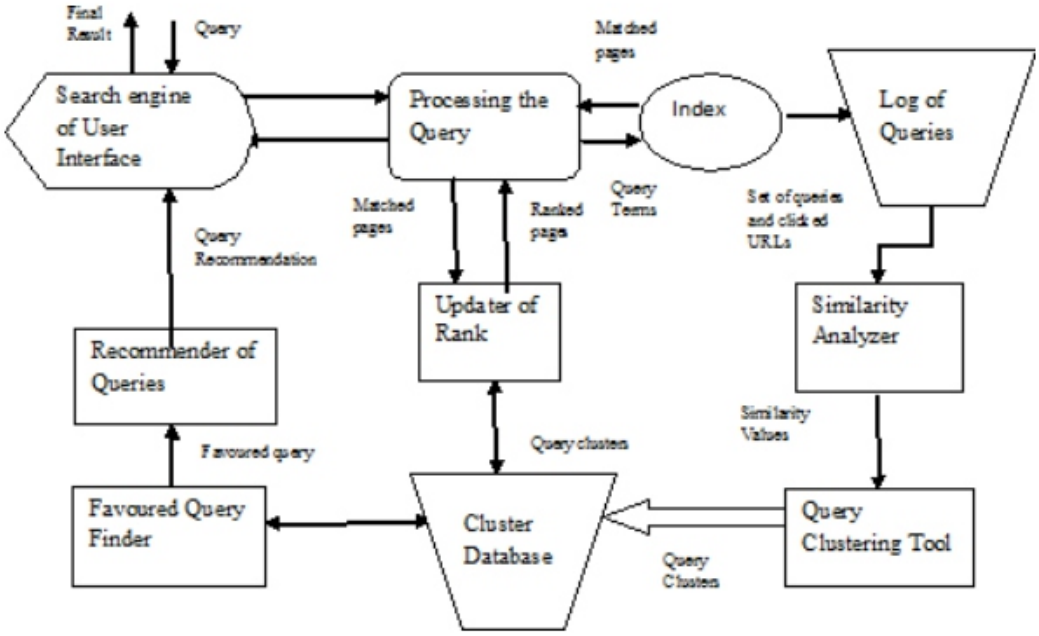


Fig.1 Proposed model

When user submits a query on the search engine interface, the query processor component matches the query terms with the index repository of the search engine and returns a list of matched documents in response. User browsing behavior including the submitted queries and clicked URLs get stored in the logs and are analyzed continuously by the Similarity Analyzer module, the output of which is forwarded to the Query Clustering Tool to generate groups of queries based on their similarities.

Favored Query Finder extracts most popular queries from each cluster and stores them for future reference. The Rank Updater component works online and takes as input the matched documents retrieved by query processor. The Query Recommender guides the user with similar queries with the most famous query.

The proposed system works in the following steps

1. Similarity Analyzer
2. Query Clustering Tool
3. Favored Query Finder
4. Updater of Rank
5. Recommender of Query

3. EXPERIMENTAL RESULTS

To show the validity of the proposed architecture, a fragment of sample query log is considered (given in Table 1). Because the actual number of queries is too large to conduct detailed evaluation, only 7 query sessions are chosen in present illustration. The following functions are tested on the 7 query sessions:

1. Keyword similarity (Simkeyword),
2. Similarity using documents clicks (Simclick),
3. Similarity using both keyword and document clicks (Simcombined)
4. Query clustering
5. Updater of Rank

Table 1. Simple Query Log

s.no	Id	User_id	Query	Clicked_id
1	2	admin	Data mining	http://www.A
2	3	admin	Data ware housing	http://www.B
3	4	admin	Data mining	http://www.B
4	5	admin	Data warehousing	http://www.A
5	6	admin	Search engine	http://www.B
6	14	admin	Database	http://www.B
7	15	admin	Data base	http://www.A

3.1 SIMILARITY AND CLUSTERING CALCULATIONS

Query Similarity Analyzer

The approach taken by this module is based on two principles:

1) Similarity based on the queries themselves and
2) Based on cross-references

.

3.1.1 Similarity based on query keywords

If two user queries contain the same or similar terms, they denote the same or similar information needs. The following formula is used to measure the content similarity between two queries.

Where kw (p) and kw (q) are the sets of keywords in the queries p and q respectively, KW (p, q) is the set of common keywords in two queries.

It is estimated that longer the query, the more reliable it is. However, as most of the user queries are short, this principle alone is not sufficient. Therefore, the second criterion is used in combination as a complement.

3.1.2 Similarity Based On Clicked URLs

A query vertex is joined with a document vertex if document has been accessed by a user corresponding to the said query. The numerical integer on each edge dictates the number of accesses to the document by distinct users for a particular query. For example a value 10 between Q1 and D1 says that 10 users have clicked on D1 corresponding to Q1. In the figure above: D1, D2, D4 are accessed with respect to Q1, thus are relevant to Q1 and D2, D3, D4 are relevant to Q2 and so on. As Q1 and Q2 share two documents D2 and D4, they can be considered similar but similarity is decided on the basis of number of document clicks.

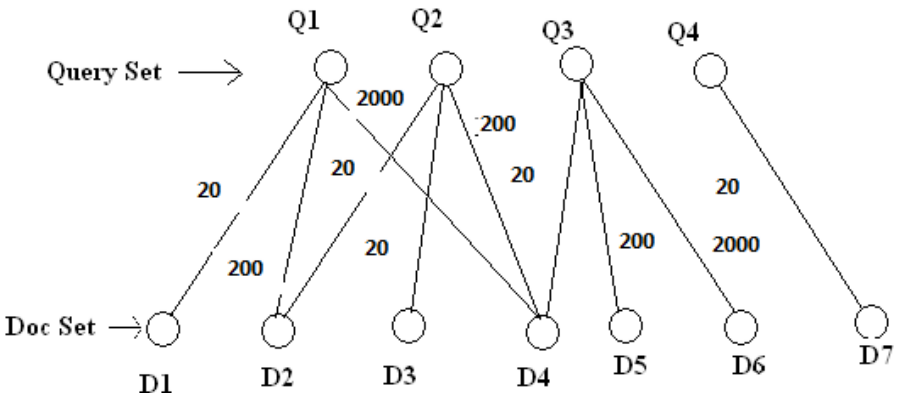


Fig. 2 Similarity Based on Clicked URLs

3.1.3 Biparite Graph of Query Log

If two queries p and q share a common document d, then similarity value is ratio of the total number of distinct clicks on d with respect to both queries and the total number of distinct clicks on all the documents accessed for both queries. If more than one document is shared, then numerator is obtained by summing up the document clicks of all common documents.

The following formula dictates the similarity function based on documents clicks.

$$Sim_{clickURL(p,q)} = \frac{\sum C(p,di) + C(q,di)}{\sum C(p,xi) + C(q,xi)}$$

Where C (p, d) and C (q, d) are the number of clicks on document d corresponding to queries p and q respectively. C (p) and C (q) are the sets of clicked documents corresponding to queries p and q respectively.

As an example illustration, Q1 and Q2 share two common documents D2 and D4, while D1, D2, D3 and D4 are accessed either by Q1 or Q2 or both. The similarity between two queries is

$$Sim_{clicked\ url}\ (Q1,Q2) = \frac{(200+20) + (2000+200)}{(20+0) + (200+20) + (0+20) + (2000+200)}$$

$$= 0.984$$

Similarly, the similarity between Q1 and Q3 is:

$$Sim_{clicked\ url}\ (Q1,Q3) = \frac{2000}{4440}$$

$$= 0.455$$

The similarity values always lie between 0 and 1. The measure given in declares two queries similar by imposing a threshold value on the similarity.

3.1.4 Combined Similarity Measure

The two criteria have their own advantages. In using the first criterion, queries of similar compositions can be grouped together. In using the second criterion, benefit can be taken from user's judgments. Both query keywords and the corresponding document clicks can partially capture the users' interests when considered separately. Therefore, it is better to combine them in a single measure. A simple way to do it is to combine both measures linearly as follows:

$$Sim_{combines(p,q)} = \alpha . Sim_{Keyword(p,q)} + \beta . Sim_{clickURL(p,q)}$$

Where α and β are constants with 0<=α(and β)<=1 and α+β=1

The values of constants can be decided by the expert analysts depending on the importance being given to two similarity measures. In the current implementation, these parameters are taken to be 0.5 each.

3.1.5 CLUSTERING ALGORITHM

Initially, all queries are considered to be unassigned to any cluster. Each query is examined against all other queries (whether classified or unclassified) by using (4). If the similarity value turns out to be above the pre-specified threshold value (ττ), then the queries are grouped into the same cluster. The same process is repeated until all queries get classified to any one of the clusters. The algorithm returns overlapped clusters i.e. a single query may span multiple clusters. Each returned cluster is stored in the Query Cluster Database along with the associated queries, query keywords and the clicked URLs.

Algorithm : Query_Clustering(Q,α,β,τ)

Given : A set of n queries and corresponding clicked url's stored in an array Q[q1,URL1…..URL m] .
1<=i<=n
α=β=0.5
Similarity Threshold τ
Output : A set C={C1,C2….Ck} of k query clusters
//Start Algorithm
K=1; // k is the number of clusters For (each query p in Q)
Set Cluster_Id(p) - Null;//Initially No Cluster is clustered For (each p € Q)
{
Cluster_Id(p) = Ck; Ck –{ p };
For each q € Q such that p ≠ q
{

$$Sim(p,q) = \frac{|KW(p,q)|}{|kw(p) \cup kw(q)|}$$

$$Sim_{clickURL(p,q)} = \frac{\sum LC(p,di)+LC(q,di)}{\sum LC(p,xi)+LC(q,xi)}$$

$$Sim_{combines(p,q)} = \alpha . Sim_{Keyword(p,q)} + \beta . Sim_{clickURL(p,q)}$$

If $(Sim_{combines(p,q)} > \tau)$

Set Cluster_Id(q) = Ck; Ck = Ck U {k};
Else Continue;
} // End For K=K+1;
} //End Outer For
Return Query Cluster Set C;

Snap shots:

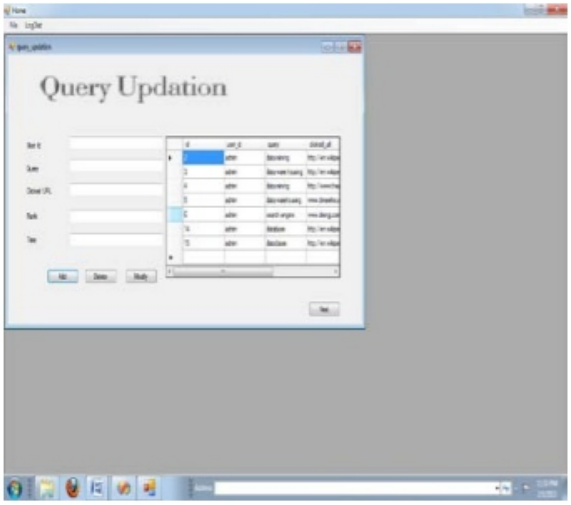


Fig 3. Query Updation

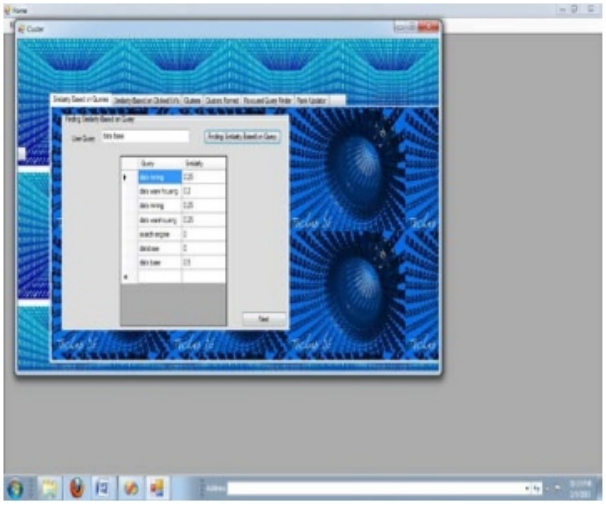


Fig. 4 Similarity form (based on keywords)

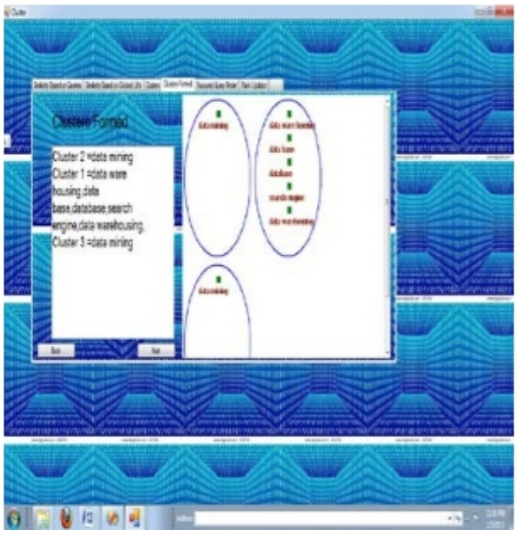


Fig 5. Cluster Formation

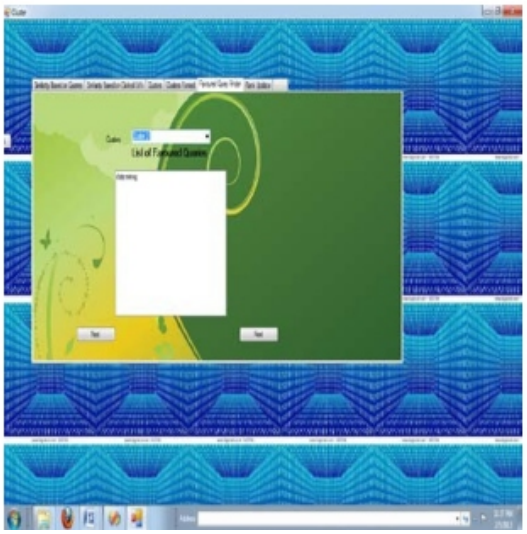


Fig. 6 Favoured Query Finder

4. CONCLUSION & FUTURE SCOPE

This approach based on query log analysis is proposed for implementing effective web search. The most important feature is that the result optimization method is based on users' feedback, which determines the relevance between Web pages and user query words. Since result improvement is based on the analysis of query logs, the recommendations and the returned pages are mapped to the user feedbacks and dictate higher relevance than the pages, which exist in the result list but are never accessed by the user. By this way, the time user spends for seeking out the required information from search result list can be reduced and the more relevant Web pages can be presented. The results obtained from practical evaluation are quite promising in respect to improving the effectiveness of interactive web search engines. Further investigation on mining log data deserves more of our attention. Further study mayresult in more advanced mining mechanism which can provide more comprehensive information about relevancy of the query terms and allow identifying user's information need more effectively.

REFERENCES

[1] A. K. Sharma, Neelam Duhan, Neha Aggarwal, Rajang Gupta. Web Search Result Optimization by Mining the Search Engine logs. Proceedings of International Conference on Methods and Models in Computer Science (ICM2CS-2010), JNU, Delhi, India, Dec. 13-14, 2010.

[2] Spirant R., and Agawam R. "Mining Sequential Patterns: Generalizations and performance improvements", Proc. of 5th International Conference Extending Database Technology (EDBT), France, March 1996.

[3] A. Birchers, J. Her locker, J. Konstantin, and J. Riel, "Ganging up on information overload," Computer, Vol. 31, No. 4, pp. 106-108, 1998.

[4] B. Amen to, L. Tureen, and W. Hill, "Does Authority Mean Quality? Predicting Expert Quality Ratings of Web Documents", In Proceedings of 23th International ACM SIGIR, pp. 296-303, 2000

[5] Beeferman and Berger A., 2000. Agglomerative clustering of a search engine query log. In Proceedings of the 6th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, (August). Acme Press, New York, NY, 407–416.

[6] D. Zhang and Y. Dong, "An Efficient Algorithm to Rank Web Resources," In Proceedings of 9th International World Wide Web Conference, pp. 449-455, 2000.

[7] J. Went, J. Mie, and H. Zhang. Clustering user queries of a search engine. In Proc. at 10th International World Wide Web Conference. W3C, 2001.

[8] Bernard J. Jansen and Undo Pooch. A review of web searching studies and a framework for future research. J. Am. Soc. Inf. Sci. Technol., 52(3):235–246, 2001.

[9] A. Aras, J. Cho, H. Garcia-Molina, A. Peace, and S. Raghavan, "Searching the Web," ACM Transactions on Internet Technology, Vol. 1, No. 1, pp. 97-101, 2001

[10] M.R. Her zinger, "Hyperlink Analysis for the Web," IEEE Internet Computing, Vol. 5, No.1, pp. 45-50, 2001.

[11] K. Bharat and G.A. Michaela, "When Experts Agree: Using Non- Affiliated Experts to Rank Popular Topics," ACM Transactions on Information Systems, Vol. 20, No. 1, pp. 47-58, 2002.

Improvement of an Efficient AES Implementations for Arm based Processor

Dr. S. Kishore Reddy*, Dr. Syed Musthak Ahmed A. Ravi Shankar*****

* Associate Professor, ECE, SR Engineering College, Warangal, AP, India

** Professor and HOD, SR Engineering College, Warangal, AP, India

*** Assistant Professor, ECE, SITS, Khammam, AP, India

A B S T R A C T

The Advanced Encryption Standard (AES) contest, started by the U.S. National Institute of Standards and Technology (NIST), saw the Rijndael [13] algorithm as its winner [11]. Although the AES is fully defined in terms of functionality, it requires best exploitation of architectural parameters in order to reach the optimum performance on specific architectures. Our work concentrates on ARM cores [1] widely used in the embedded industry. Most promising implementation choices for the common ARM Instruction Set Architecture (ISA) are identified, and a new implementation for the linear mixing layer is proposed. The performance improvement over current implementations is demonstrated by a case study on the Intel StrongARM SA-1110 Micropro-cessor [2]. Further improvements based on exploitation of memory hierarchies are also described, and the correspond-ing performance figures are presented.

Keywords: AES, ARM microprocessor, code optimisation, cache memories.

1. INTRODUCTION

The AES, developed to replace The old Data Encryption Standard (DES), is the result of a four-year effort involving the cooperation between the U.S. Government, private industry and academia from around the world. The call for algorithms made by NIST required that algorithms must be based on symmetric key cryptography and work as a block cipher. Among fifteen candidate algorithms, Rijn-dael's combination of security, performance, efficiency, ease of implementation and flexibility made it the most appropriate choice for NIST as the Advanced Encryption Standard.

With the increasing use of portable and wireless devices in the business and daily life, protecting sensitive information via encryption is becoming more and more crucial. High- performance, low-power ARM processor cores are now li-censed by many major semiconductor partners and widely used in smart cards, portable communication devices and by consumer electronics industry. Our work aims in fact at performing a comprehensive exploration of the solution space for efficient software implementations of AES on ARM processors. Efficient software implementations of AES pro-posed so far are listed in [13] [17] [12]. However, in this work, the peculiarities of the common ARM ISA suited for AES are identified and appropriately exploited, also allowing us to design a new implementation, which exhibits interesting features with respect to the known ones.

This paper is organised as follows: Sect. 2 describes the Rijndael algorithm. Sect. 3 presents the ARM ISA and the related optimisations. Sect. 4 presents Intel StrongARM SA-1110 Microprocessor architecture, real performance fig-ures on this processor, together with improvements due to ISA related optimisations, and improvements due to ex-ploitation of memory hierarchies and cache locking mech-anisms. Finally Sect. 5 summarizes our results.

2. THE RIJNDAEL ALGORITHM

Rijndael is an iterated block cipher with a variable block length and key length [14] [15]. The block length and the key length can be independently set to 128, 192 or 256 bits, whereas AES restricts the block length to 128 bits only [16].

The number of round transformations to be applied on the input blocks is either 10, 12 or 14, depending on the key length. The round transformation is composed of three dis-tinct invertible transformations, called layers: (1) the non-linear layer, (2) the linear mixing layer, and (3) the key ad-dition layer. The different transformations operate on the intermediateresult called the State. The State can be considered as a two dimensional array of bytes. The array has four rows, and the number of columns depend on the block length. The nonlinear layer makes a nonlinear byte substi-tution on each of the State bytes. The linear mixing layer rotates the rows of the state array over different offsets, and applies a linear transformation called the MixColumn trans-formation on each column of the State. The key addition layer simply XORs State bytes with subkey bytes that are generated after a keyschedule.

All the operations described above, with the exception of the MixColumn transformation, are quite straightforward to implement, therefore we will mainly concentrate on imple-mentation of the MixColumn transformation in this paper. This is also justified by the fact that in compact implementations the MixColumn transformation is responsible for about 50% of the encryption time.

2.1 The MixColumn Transformation

The MixColumn transformation makes use of arithmetic operations in the finite field GF(2n).

Table 1: Look-up tables used by different versions

Version	Encryption	Decryption
V1	Sbox	InvSbox
V1T	Sbox	InvSbox
V2	Sbox + Enc. table	InvSbox + Dec. table

Table 2: Sizes of the look-up tables in bytes

Look-up table	Size in bytes
Sbox	256
InvSbox	256
Enc. table	1K
Dec. table	1K

We assume that the reader has a basic background of Galois Fields, but for com-pleteness we recall that addition in GF(2^n) is equivalent to a simple bitwise XOR, while multiplication is obtained by reducing the result of standard multiplication (with XOR as sum) modulo a fixed polynomial. This polynomial must be irreducible to preserve the algebraic structure of field. In the MixColumn transformation, each column of the State is considered as a polynomial with coefficients in GF (2^8), and multiplied modulo x^4 + 1 with a fixed polynomial $\{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$, coprime to the modulo. Assum-ing that the column before transformation consists of the bytes (b_0 , b_1, b_2, b_3), each byte representing a polynomial in GF (2^8), the transformed column bytes (c_0, c_1, c_2, c_3) are computed as follows:

$$
\begin{aligned}
c_0 &= \{02\} \cdot b_0 \oplus \{03\} \cdot b_1 \oplus \{01\} \cdot b_2 \oplus \{01\} \cdot b_3 \\
c_1 &= \{01\} \cdot b_0 \oplus \{02\} \cdot b_1 \oplus \{03\} \cdot b_2 \oplus \{01\} \cdot b_3 \\
c_2 &= \{01\} \cdot b_0 \oplus \{01\} \cdot b_1 \oplus \{02\} \cdot b_2 \oplus \{03\} \cdot b_3 \\
c_3 &= \{03\} \cdot b_0 \oplus \{01\} \cdot b_1 \oplus \{01\} \cdot b_2 \oplus \{02\} \cdot b_3
\end{aligned}
\qquad (1)
$$

where $\cdot$ denotes polynomial multiplication in GF (2^8) de-fined by the irreducible polynomial $x_8 + x_4 + x_3$ + x + 1, and $\oplus$ denotes simple XOR at byte level. Multiplication by {02} in GF (2^8) can be implemented at byte level with a left shift followed by a conditional bitwise XOR with {1b}. Mul-tiplication by larger coefficients can be implemented with repeated multiplications by {02} and XORs with previously calculated results.

2.2 32-bit Implementation Strategies

On 32-bit processors byte-level operations can be done in parallel. In particular, the bytes within a column can be multiplied by {02} in parallel, and we will refer to this operation as the X_{time} operation. Several efficient imple-mentations of the Xtime operation have been described by Brian Gladman in [17].

The original Rijndael submission defines two main imple-mentation strategies. The first strategy, which we will call Version 1 (V1), employs byte substitution tables only (Sbox and InvSbox). The second strategy, which we will call Ver-sion 2 (V2), employs additional encryption/decryption ta-bles in order to avoid the software computation of (1). Ver-sion 1 implements the nonlinear layer using byte substitu-tion tables and implements the MixColumn transformation making repetitive calls to the Xtime operation. Version 2, on the other hand, combines the nonlinear layer, and the linear mixing layer within 4 table lookups, 3 XORs and 3 rotates allowing very fast implementations on 32 bit processors.

The third strategy proposed recently by Bertoni et al. in [12], like Version 1, employs the nonlinear byte substitution tables only, however transposes the state array before ap-plying the round transformations, and transposes again at the end. In this way, considerable amount of computation is saved in the implementation of the MixColumn and Inverse MixColumn (InvMixColumn) transformations, resulting in higher performance than V1 in general but maintaining the same low memory requirements. We will refer to this strat-egy as the Transposed State Version (V1T).

The look-up tables used by the three described strategies are given in Table 1, and the sizes of the different look-up tables are given in Table 2. The use of pre-rotated encryp-tion and decryption tables, and word extended byte sub-stitution tables has also been proposed, respectively in the Rijndael submission and in [17] by Brian Gladman. These optimisations can eliminate a number of shifts and rotates on some architectures, and can be considered among other implementation strategies.

3. OPTIMISATIONS FOR ARM PROCES-SORS

ARM is the leading provider of 32-bit embedded RISC mi-croprocessors with almost 75% of the market. ARM offers a wide range of processor cores based on a common architec-ture [9] [4], delivering high performance together with low power consumption and system cost. ARM processors implement a load/store architecture. De-pending on the processor mode, 15 general purpose registers are visible at a time. Almost all ARM instructions can be executed conditionally on the value of the ALU status flags. Load and store instructions can load or store a 32-bit word or an 8-bit unsigned byte from memory to a register or from a register to memory.

The ARM arithmetic logic unit has a 32-bit barrel shifter that is capable of shift and rotate operations. The sec-ond operand to all ARM data-processing and single register data-transfer instructions can be shifted before data process-ing or data transfer is executed, as part of the instruction. The amount by which the register should be shifted may be contained in an immediate field in the instruction, or in the bottom byte of another register. When the shift amount is specified in the instruction, it may take any value from 0 to 31, without incurring any penalty in the instruction cycle time.

Use of word extended substitution tables in Rijndael im-plementations is unnecessary and inefficient on ARM proces-sors, since the architecture supports load byte instructions. Use of pre-rotated tables cannot improve the performance neither, since the barrel shifter that can be combined with data processing instructions reduces the effective cost of ro-tate instructions to zero. Use of such tables, in

fact, in-creases the register pressure and possibility of cache misses, therefore degrading the performance. We will consider only V_1, V_2 and V_1T described in Sect. 2.2 in the rest of this paper.

3.1 The Proposed Mix Column Implementation

The MixColumn implementation described by Gladman [17] in V_1 requires 4 XORs, 3 rotates and one Xtime opera-tion, incurring 16 XORs, 12 rotates and 4 Xtime operations per AES round.

V1T by Bertoni et al. [12] eliminates the rotations, and requires 16 XORs and 4 Xtime operations per AES round. In fact, the advantage of using a transposed state is more evident in the decryption operation, because the InvMixCol-umn operation sees an important reduction in the number of XORs and Xtime operations.

We describe here a new MixColumn implementation that requires 3 XORs, 3 rotations and one Xtime, incurring 12 XORs, 12 rotations and 4 Xtime operations per AES round. However, using the ARM barrel shifter, the 12 rotations can be combined with 12 XORs without any penalty, resulting in 12 XORs and 4 Xtime operations effectively per round. The proposed MixColumn implementation, in addition to cutting down the number of logical operations, can support all block lengths multiples of 32-bits, unlike the Transposed State Version, which requires a State matrix of 128-bits.

Assuming that $b = (b_0, b_1, b_2, b_3)$ is the input column to be transformed, s and t are two 32- bit temporary variables, and $c = (c_0, c_1, c_2, c_3)$ is the result, the four steps of the MixColumn transformation are given as follows, where EOR is the ARM instruction for XOR, and ROL is the rotate left command for the barrel shifter:

1. **EOR s, b, b ROL 8**
   $s_0 = b_0 \oplus b_1$, $s_1 = b_1 \oplus b_2$
   $s_2 = b_2 \oplus b_3$, $s_3 = b_3 \oplus b_0$
2. **EOR t, s, b ROL 16**
   $t_0 = b_0 \oplus b_1 \oplus b_2$ $t_1 = b_1 \oplus b_2 \oplus b_3$
   $t_2 = b_2 \oplus b_3 \oplus b_0$ $t_3 = b_3 \oplus b_0 \oplus b_1$
3. **s = Xtime(s)**
   $s_0 = \{02\}_3 (b_0 \oplus b_1)$ $s_1 = \{02\}_3 (b_1 \oplus b_2)$
   $s_2 = \{02\}_3 (b_2 \oplus b_3)$ $s_3 = \{02\}_3 (b_3 \oplus b_0)$
4. **EOR c, s, t ROL 8**
   $c_0 = \{02\}_3 (b_0 \oplus b_1) \oplus b_1 \oplus b_2 \oplus b_3$ $c_1 = \{02\}_3 (b_1 \oplus b_2) \oplus b_2 \oplus b_3 \oplus b_0$ $c_2 = \{02\}_3 (b_2 \oplus b_3) \oplus b_3 \oplus b_0 \oplus b_1$ $c_3 = \{02\}_3 (b_3 \oplus b_0) \oplus b_0 \oplus b_1 \oplus b_2$

The final result is equivalent to (1).

4. A CASE STUDY ON INTEL STRONG-ARM SA-1110 MICROPROCESSOR

The Intel StrongARM SA-1110 Microprocessor (SA-1110) is a highly integrated 32-bit RISC microprocessor that incor-porates Intel design and process technology along with the power efficiency of the ARM architecture [10] [7]. The SA-1110 implements the ARM V4 ISA, together with a Harvard architecture memory system. There are separate instruc-tion and data caches. The instruction and data streams are translated through independent memory- management units (MMUs) [6]. The 16 Kbytes Icache has 512 lines of 32 bytes (8 words), arranged as a 32-way set associative cache. There are two logically separate data caches: the main data cache and the mini data cache (minicache). The main data cache, an 8 Kbyte write- back Dcache, has 256 lines of 32 bytes (8 words) in a 32-way set-associative organization.

Table 3: Number of clock cycles, 128-bit key length

Version	Enc. cycles	Dec. cycles
V1	1020	1605
V1T	1033	1312
Proposed	943	1605

The mini-cache provides a smaller and logically separate data cache that can be used to enhance caching performance. The mini-cache is a 512-byte write-back cache having 16 lines of 32 bytes (8 words) in a two-way set-associative organization. The Dcaches are accessed in parallel and the design ensures that a particular line entry will exist in only one of the two at any time. Both Dcaches use the virtual address generated by the processor and allocate only on loads according to the set-tings of certain bits in the MMU translation tables. Stores are made using a four-line write buffer. The performance of specialized load routines is enhanced with the four-entry read buffer that can be used to prefetch data. The Stron-gARM pipeline has 5 stages: fetch, decode, execute, buffer and writeback. Most instructions normally spend a single cycle in each stage. The StrongARM core contains a num-ber of result bypasses. These normally allow the processor to use the results of one instruction in the following instruc-tion as soon as it has been generated. In particular, almost every instruction can read its inputs from the bypasses as it enters the execute stage if these inputs are not yet in the register file in the decode stage.

4.1 Performance Figures

We worked with Intel StrongARM SA-1110 Microproces-sor Development Board, Hardware Build Phase 5. We used ARM Developer Suite (ADS) Version 1.1 and Angel Version 1.2 Revision 2.08a as the development software.

We implemented in C the three strategies explained in Sect. 2.2. We unrolled the loops that implement the series of round transformations by one, and we kept the State array explicitly in registers. We applied the standard techniques described in [8] in order to optimise the machine codes gen-erated by the ARM compiler. We calculated the clock cycle information reading the Operating System Timer clocked by the 3.6864 MHz oscillator, that also feeds the CPU phase-locked loop (PLL). In order to remove the effects of initial cache misses, we made a large number of consecutive en-cryptions or decryptions on random input blocks, and we took the averages.

Performance figures with 128-bit key length based on the three MixColumn implementations described in Sect. 3.1 are given in Table 3. We underline the fact that in Table 3 we compare the speed of compact implementations, i.e. we exclude V2 because its code does not carry out the MixCol-umn step; in V2 the Mixcolumn operation is hardwired in the bigger look-up tables.

The experimental results are parallel to the theoretical claims: encryption with the proposed MixColumn imple-mentation is the fastest. V1T encryption is slightly slower than V1 encryption due to the initial and the final trans-positions on the State matrix. Instead, V1T is winning in decryption. We conclude that the best compact implementation is obtained combining proposed encryption code and V1T decryption code; we will refer to this version as BC.

A comparison of performances between BC and V2 is given in Table 4 for different key lengths. The results show that V2 is the fastest implementation on the processor, both for encryption and decryption. We note here that the per-formance figures of V2 compare well with the StrongARM results given in

[3]. However, an exact comparison has not been possible, since the development platform and the specific processor are not specified in the latter case.

Table 4: Number of clock cycles for different key lengths

Version	Key length	Encryption	Decryption
BC	128-bit	943	1312
	192-bit	1119	1559
	256-bit	1295	1806
V2	128-bit	659	638
	192-bit	767	766
	256-bit	875	874

4.2 Memory Requirements

The three different versions have different memory re-quirements. V_1 and V_1T employ only byte substitution ta-bles, whereas V_2 employs additional encryption/decryption tables. It is clear that V2 requires more data memory com-pared with V_1 and V_1T. However, use of the look-up tables reduces its code size. Table 5 summarizes the memory re-quirements for the different versions, containing the code for keyschedule for the three different key lengths, the code for encryption and the code for decryption; RO stands for read only and ZI stands for zero initialised.

Table 5: Memory requirements for different versions in bytes

Version	Code	RO Data	ZI Data	Total
V1	3620	522	248	4390
V1T	4440	522	248	5210
V2	3148	2570	248	5966

V2 again has the largest memory requirements, when code and data are considered together, and V1 is the most com-pact implementation. Although V1T uses the same look-up tables as V1, it has larger memory requirements; this can be explained by the fact that it requires a more complex key schedule, which increases its code size.

4.1 The Mini-cache Solution and Effects of Cache Misses

The mini data cache, having a size of 512 bytes, is an ideal place to lock the byte substitution tables used by all candidate implementations. By manipulating memory man-agement translation tables, it is possible to assign the sub-stitution tables to the mini data cache, and all the other data area to the main data cache [5].

Table 6: Effects of the Mini-cache use on the per-formance, clock cycles, 128-bit key length

Version	Operation	With MC	Without MC
V1	keysch. + 1 enc.	3877	4204
V2	keysch. + 1 enc.	4355	4734
V1T	keysch. + 1 dec.	4719	5365
V2	keysch. + 1 dec.	5614	6215

This ensures that there will be no interference by other data structures, or other applications on the mini-cache, and once loaded, sub-stitution tables will always remain there. When there is an operating system running on the platform, resulting in fre-quent context switches, or when the number of consecutive encryptions or decryptions are small and must be succeeded (or preceded) by other applications, cache interference can present significant penalties on AES performance.

We tried to reduce this penalty by making use of the SA-1110 mini-cache. Table 6 describes the effects of the mini-cache use on the performance with the instruction cache and the main data cache initially empty. According to Table 6 results, a reduction of 8 to 12 percent in thenumber of clock cycles can be obtained with a single keyschedule and a single encryption or decryption by making use of the mini-cache. The improvement is more significant for decryption since it makes use of both Sbox and InvSbox (Sbox in keyschedule). Each byte substitution table, having a size of 256 bytes, oc-cupies 8 cache blocks with proper alignment. Keeping them in the mini-cache, 8 cache misses are saved for encryption and 16 cache misses are saved for decryption. However, the performance figures are still quite far from those given in Sect. 4.1. The reason is the instruction cache misses, which cannot be avoided in any way. The improvement due to the mini-cache is still convincing.

Another observation that can be made from Table 6 is that V2 encryption requires more clock cycles than V1 en-cryption, and V2 decryption requires more clock cycles than V1T decryption in case of a single encryption or decryp-tion. This is expected, since V2 is subject to additional data cache misses due to the encryption/decryption tables it uses. Moreover, V2 decryption keyschedule is more complex than others as it requires additional InvMixColumn transforma-tions applied on the expanded key. However, as the number of encryptions or decryptions are increased, we would ex-pect the speed of V2 to compensate for the cache miss or keyschedule overheads at some point.

Table 7 shows that V2 encryption is faster than BC en-cryption only if the number of consecutive encryptions is larger than 2. The decryption results, shown in Table 8, are similar: V2 decryption is faster than BC decryption when the number of consecutive decryptions is larger than 2.

We conclude that, for a particular application encrypting or decrypting messages smaller than or equal to 2 blocks (32 bytes) at a time and subject to cache block replacements between two executions, it is better strategy to use version BC instead of V2, not only from the memory requirements point of view, but also considering performance.

We may add that a reduction in the number of clock cycles probably lead also to a reduction in energy consumption, although we did not make such measurements. The use of BC is thus recommended in several use-cases, such as:

Table 7: BC encryption vs V2 encryption, clock cy-cles, 128-bit key length, substitution tables in mini-cache

Operation	BC	V2
keysched. + 1 enc.	3877	4355
keysched. + 2 enc.	4831	5021
keysched. + 3 enc.	5796	5680
keysched. + 4 enc.	6712	6328

Table 8: BC decryption vs V2 decryption, clock cy-cles, 128-bit key length, substitution tables in mini-cache

Operation	BC	V2
keysched. + 1 dec.	4719	5614
keysched. + 2 dec.	6027	6229
keysched. + 3 dec.	7390	6882
keysched. + 4 dec.	8691	7507

- The case when the running application must encrypt or decrypt and send the value of some counters or some flags, for the purpose of getting synchronized with a remote host.
- The case when data to be sent over an encrypted con-nection consist of key strokes from the keyboard or from dedicated pads (very common when using remote shells or interfaces)
- The case when small data from sensors must be pe-riodically sent to a sink, and other applications are running on the device.

5 CONCLUSIONS

In this work, the peculiarities of the common ARM ISA suited for AES are identified and appropriately exploited. A new implementation for the encryption linear mixing layer is formulated, which enhances the performance of AES on all ARM cores. The new implementation is the most compact implementation, preserves the flexibility of Rijndael and ex-hibits the highest encryption performance on ARM proces-sors, compared with similar implementations claiming low memory use. It exploits the ISA features of ARM by effi- ciently rearranging some operations and reducing their num-ber.

A case study on Intel StrongARM SA-1110 microproces-sor presents the performance figures for different candidate implementations and some ways of improving the perfor-mance by exploiting memory hierarchies and cache lock-ing strategies. Moreover, the case study demonstrates that the most compact implementations are also the fastest ones when there is potential cache interference, and identifies the smallest number of consecutive encryptions/decryptions that make implementations using additional look-up tables advantageous over the more compact ones on the specific architecture.

6. REFERENCES

[1] Arm Ltd. website. http://www.arm.com.
[2] Intel Ltd. website. http://www.intel.com.
[3] A survey of Rijndael implementations.
[4] http://www.tcs.hut.fi/~helger/aes/rijndael.html.
[5] ARM7. Data Sheet ARM DDI 0020C, ARM Limited, Dec 1994.
[6] Configuring ARM Caches. Application Note ARM DAI 0053B, ARM Limited, Feb 1998.
[7] Memory Management on the StrongARM SA-110. Application Note 278191-001, Intel Corporation, Sep 1998.
[8] StrongARM SA-110 Microprocessor Instruction Timing. Application Note 278194- 001, Intel Corporation, Sep 1998.
[9] Writing Efficient C for ARM. Application Note ARM DAI 0034A, Jan 1998.
[10] ARM Architecture. Reference Manual ARM DDI 0100D, ARM Limited, Feb 2000.
[11] Intel StrongARM SA-1110 Microprocessor. Developer's Manual 278240-003, Intel Corporation, Jun 2000.
[12] Announcing the ADVANCED ENCRYPTION STANDARD (AES). Federal Information Processing Standard FIPS 197, National Institute of Standards and Technology (NIST), Nov 2001.
[13] G. Bertoni, L. Breveglieri, P. Fragneto, M. Macchetti, and S. Marchesin. Efficient Software Implementation of AES on 32-Bit Platforms. In B. S. K. Jr., C¸ etin Kaya Ko¸c, and C. Paar, editors, Cryptographic Hardware and Embedded Systems - CHES 2002, volume 2523 of Lecture Notes in Computer Science, pages 159–171. Springer, Berlin, Aug. 2002.
[14] J. Daemen and V. Rijmen. AES Proposal:Rijndael. http://csrc.nist.gov/CryptoToolkit/aes/rijndael/, Sep 1999.
[15] J. Daemen and V. Rijmen. Efficient Block Ciphers for Smartcards. In USENIX Workshop on Smartcard Technology (Smartcard '99), pages 29–36, May 1999.
[16] J. Daemen and V. Rijmen. The Block Cipher Rijndael. In J.-J. Quisquater and B. Schneier, editors, Smart Card Research and Applications, volume 1820 of Lecture Notes in Computer Science, pages 288–296. Springer, Berlin, 2000.
[17] J. Daemen and V. Rijmen. Rijndael, the Advanced Encryption Standard. Dr. Dobb's Journal, 26(3):137–139, Mar. 2001.
[18] B. Gladman. A Specification for Rijndael, the AES Algorithm. Available at http://fp.gladman.plus.com, May 2002

Improving Perfor Mance & Power in Multi-Core Processors

Dr S. Kishore Reddy*, Dr. Syed Musthak Ahmed, K. Manohar*****
* Associate Professor, ECE, SR Engineering College, Warangal, AP, India
**Professor & Head, Department of ECE, SR Engineering College, Warangal, AP, India
***Lecturer, Electrical & Computer Engg, Wollo University, Ethiopia

ABSTRACT

Multi-core processor architecture entails silicon design engineers placing two or more processor-based execution cores, or computational engines, within a single processor. A multi-core processor plugs directly into a single processor socket, but the operating system perceives each of its execution cores as a discrete logical processor, with all the associated execution resources. A processor equipped with thread-level parallelism can execute completely separate threads of code. This can mean one thread running from an application and a second thread running from an operating system, or parallel threads running from within a single application. Multimedia applications are especially conducive to thread-level parallelism because many of their operations can run in parallel.

Although faster processors are one way to improve server performance, other approaches can help boost performance without increasing clock speed and incurring an increase in power consumption and heat. Multi-core processing continues to exert a significant impact on software evolution. Before the advent of multi-core processor technology, both SMP systems and HT Technology motivated many OS and application vendors to design software that could take advantage of multithreading capabilities. As multi core processor–based systems enter the mainstream and evolve, it is likely that OS and application vendors will optimize their offerings for multi-core architectures, resulting in potential performance increases over time through enhanced software efficiency. In a technical nutshell, multi-core processing can support several key capabilities that will enhance the user experience, including the number of PC tasks a user can do at one time, do multiple bandwidth-intensive activities and increase the number of users utilizing the same PC.

Keywords: Multi core processor, computational engines, multimedia applications, HT technology, nutshell, pc tasks.

INTRODUCTION

Core: The processing part of a CPU chip minus the cache. It is made up of the control unit and the arithmetic logic unit (ALU).

Multi-Core: "Multi Core" refers to - two or more CPUs working together on one single chip (like AMD Athlon X2 or Intel Core Duo) in contrast to DUAL CPU, which refers to two separate CPUs working together.

ADVANTAGES IN USING MULTI-CORE PROCESSORS

The proximity of multiple CPU cores on the same die allows the cache coherency circuitry to operate at a much higher clock rate than is possible if the signals have to travel off-chip. Combining equivalent CPUs on a single die significantly improves the performance of cache snoop (Bus snooping --> Bus sniffing or Bus snooping is a technique used in distributed shared memory systems and multiprocessors

aimed at achieving cache coherence. Every cache controller monitors the bus, waiting for broadcasts which may cause it to invalidate its cache line.) operations. Put simply, this means that signals between different CPUs travel shorter distances, and therefore those signals degrade less. These higher quality signals allow more data to be sent in a given time period since individual signals can be shorter and do not need to be repeated as often. Assuming that the die can fit into the package, physically, the multi-core CPU designs require much less Printed Circuit Board (PCB) space than multi-chip SMP designs. Also, a dual-core processor uses slightly less power than two coupled single- core processors, principally because of the increased power required to drive signals external to the chip and because the smaller silicon process geometry allows the cores to operate at lower voltages; such reduction reduces latency. Furthermore, the cores share some circuitry, like the L2 cache and the interface to the front side bus (FSB). In terms of competing technologies for the available silicon die area, multi-core design can make use of proven CPU core library designs and produce a product with lower risk of design error than devising a new wider core design. Also, adding more cache suffers from diminishing returns. Let's start by looking at some of the multi-processor technologies which have contributed to AMD and Intel's newest products.

SMP (Symmetric Multi-Processing): SMP is the most common approach to creating a multi- processor system, in which two or more separate processors work together on the same motherboard. The processors co-ordinate and share information through the system bus and the processors arbitrate the workload amongst themselves with the help of the motherboard chipset and the operating system. The OS treats both processors more or less equally,assigning work as needed. Both AMD and Intel's new dual-core chips can be considered SMP capable (internally). AMD's dual-core Opteron server processors can be linked to other dual- core chips externally also, but this capability is not present in either company's desktop dual- core lines. The major limitations of SMP have to do with software and operating system support. Many operating systems (such as Windows XP Home) are not SMP capable and will not make use of the second physical processor. Also, most modern programs are single-threaded, meaning that there is only ever one current set of linked instructions and data for them. This means that only one processor can effectively work on them at a time. Multi- threaded programs do exist, and can take better advantage of the potential power of dual- or multi-CPU configurations, but are not as common as we might like. No other current mainstream desktop processors are SMP capable, as Intel and AMD tend to restrict cutting edge technology to the higher-end server processors such as the Opteron and Xeon. In the past though, mainstream processors have been SMP capable, most notably the later Intel Pentium 3 processors.

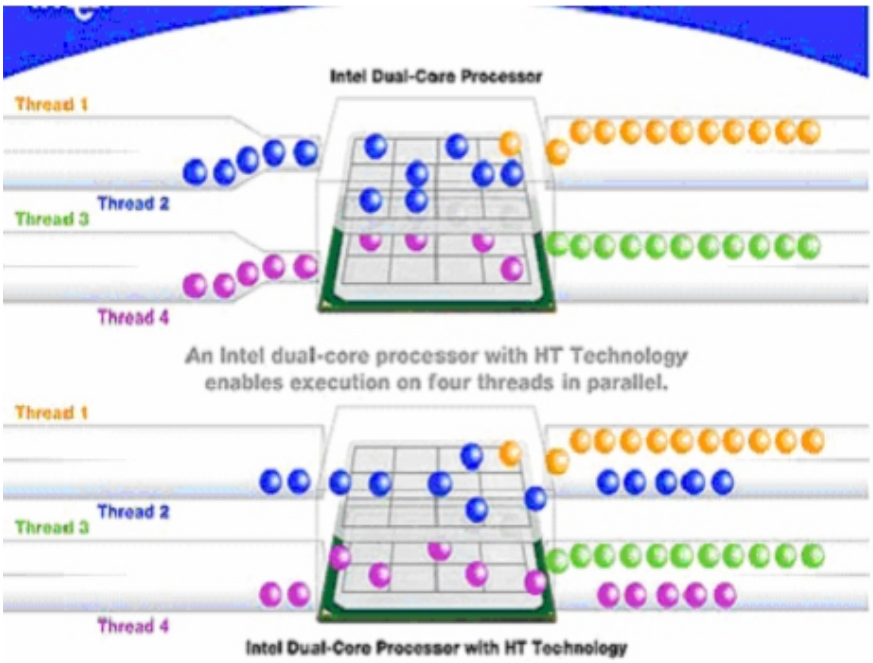
IMPLEMENTING MULTIPROCESSORS

Implementing a multiprocessing system involves more than just buying a second processor. You'll need to outfit your SMP motherboard with a special chip that will manage the various functions of all processors simultaneously. The processors you use will also need to be SMP- friendly. Some of the complex a function that multiprocessing necessitates are worth looking at. In order for two processors to not get ahead of each other and start generating data that renders either of them out-of-date, multiprocessing depends on something called cache coherency. As one processor digs into its cached memory to retrieve a piece of information, the other processor checks that it hasn't updated that particular item without the other processor's knowledge. If it has, then it supplies the latest data. This cache coherency prevents data from becoming corrupted as two or more processors work more or less separately on one task.

Another feature that makes multiprocessing work better is referred to as bus arbitration. In order to access memory that's deeper than the processor's cache -- i.e., the RAM or the computer's hard drive -- the processor must go through something called the system bus. When more than one processor is operating, however, a squabble can arise over which processor gets to use the bus first. Bus arbitration assigns different voltages to pins on the processors, which in turn determine which processor is active and which is inactive. The active processor gets dibs on the bus. And this active/inactive status switches randomlybetween the processors, so nobody gets upset. At the end of the day, multiprocessing is a neat concept, but for at-home computer-users, it hardly pays off. Doubling your processor does not necessarily double your speed, unless you're constantly using applications that benefit from dual processors. For most at-home users, this is not the case. For them, given the cost of all the necessary upgrades, you're better off investing in a typing course to speed up your keyboard skills than shelling out for this kind of glamorous upgrade.

HIGHWAY TO HYPER THREADING:

Hyper threading was Intel's pre-emptive take on multi-core CPUs. The company cloned the front end of its high-end Pentium 4 CPUs, allowing the Pentium 4-HT to begin two operations at once. Once in process, the twin operation 'threads' both share the same set of execution resources, but one thread can take advantage of sections left idle by the other. The idea of Hyper threading is to double the amount of activity in the chip in order to reduce the problem of 'missed' memory cache requests slowing down the operation of the processor. It also theoretically ensures that less of the processor's resources will be left idle at any given time.



While Hyper threaded CPUs appear as two logical processors to most operating systems, they are not comparable with true dual-core CPUs since each parallel pair of threads being worked on share the same execution pipeline and same set of L1 and L2 cache memory. Essentially, Hyper threading is smoke-and-mirrors multitasking, since a single Hyper threaded processor cannot actually perform two identical actions at the same time.

Hyper threading does speed up certain operations which would be multi-processor capable, but never as much as a true multi-processor system, dual core or not. One of Intel's new dual- core chips, the higher-end Pentium Extreme Edition 840 processor, also support Hyper threading within each core,

meaning that to an operating system it would appear as four logical processors on a single die. How this will work out remains to be seen.

DUAL CORE PROCESSORS

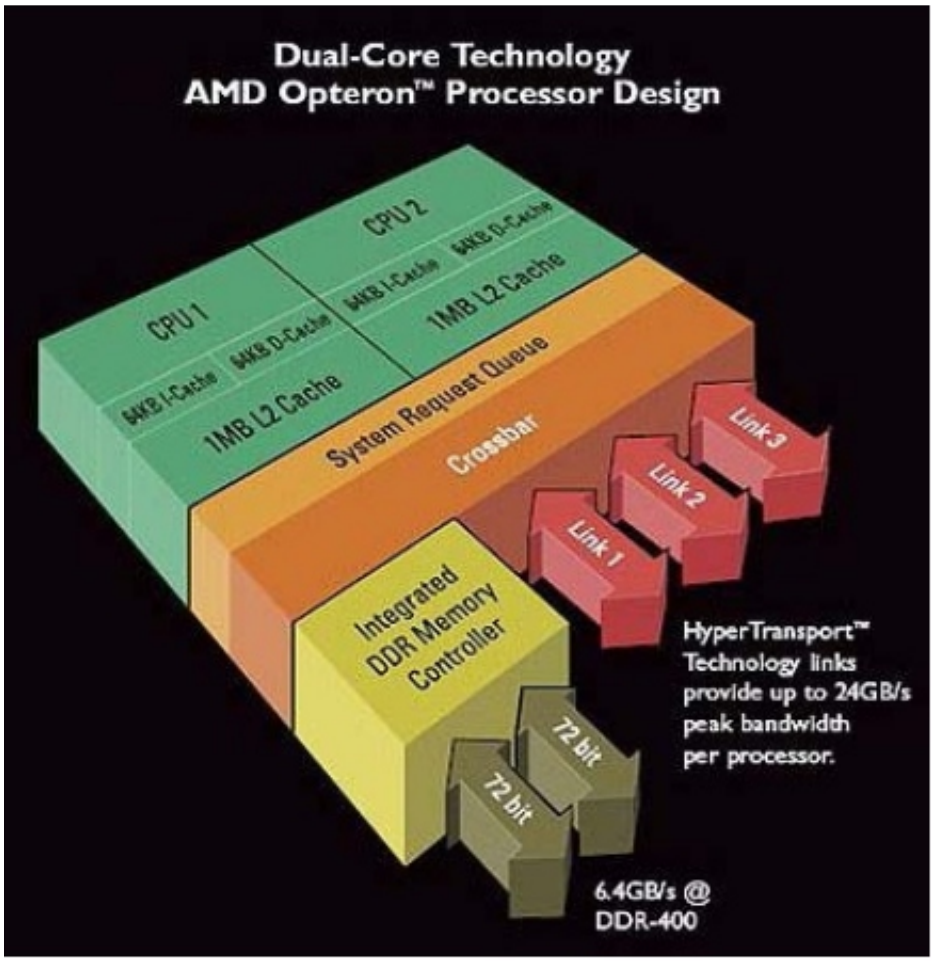
A dual-core processor is a single chip that contains two distinct processors or "execution cores" in the same integrated circuit.

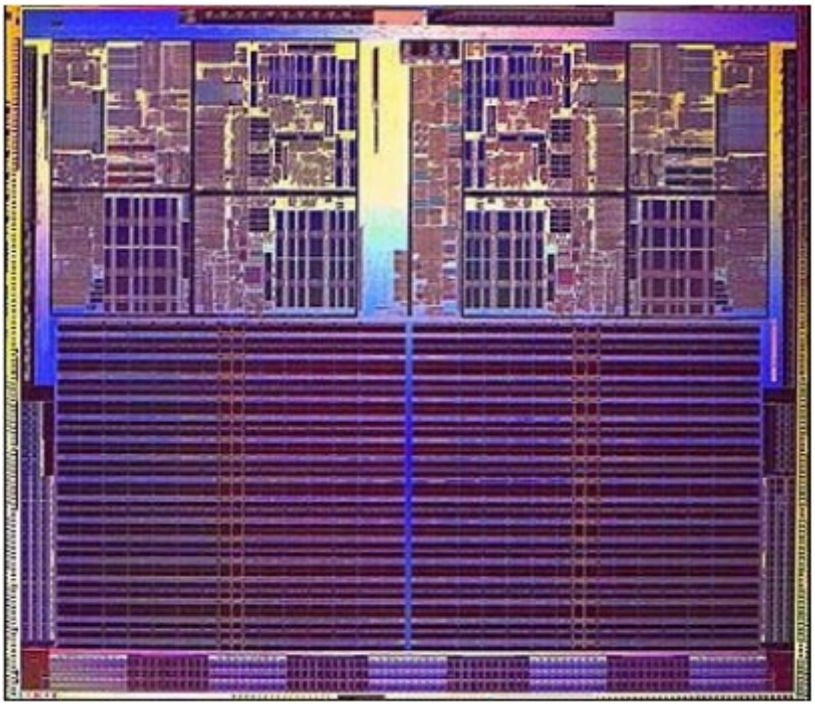
Two Chips on One Die… Why?

Several reasons; first of all; competition, competition, competition!!! The race to superiority between AMD and INTEL (the precursors of Processor technology). AMD built the potential for dual-core capability into its 64-bit processors right from the start. The necessary I/O structure for the second core already exists, even on single core chips. Neither company can afford to let the other get much of an edge, and AMD has already stolen way too much attention for Intel's comfort with its incredibly successful line of 64-bit processors. It is imperative for Intel to launch a 'pre-emptive strike' and get its own dual-core technology to market quickly, lest market share flutter away. As for why dual core processors are being developed in the first place, read on to reason number three. Secondly, performance. Certain 'multi-threaded' applications can already benefit greatly by allowing more than one processor to work on them at once. Dual processor systems also gain from a general decline in latency. Simply put, while there is no current way to share the current operating system load evenly between two processors, the second processor can step in and keep the system running smoothly while the first is maxed out to 100% burning a CD or encoding a file (or from a software error). Obviously, if dual-core systems become main stream, which it looks like they are going to, future operating systems and applications will be designed with the feature in mind, leading to better functiona4lity down the road. Thirdly, and less obviously, AMD and Intel are desperate. Both companies have run into barriers when it comes to increasing the raw speed of processors, or decreasing the die size. Until these roadblocks are cleared or until the general buying public understands that GHz does not directly translate to performance, both companies will be scrambling to discover any new improvements that will improve processor performance... without actually boosting core speed. This is why the idea of dual- core processors is now a reality.

DUAL SINGLE-CORE VS. SINGLE DUAL- CORE

AMD's Opteron chip is capable of SMP due to its multiple hyper transport links, so which is faster; a single dual-core chip or two single-core chips? On paper, dual Opterons should be faster than a single dual-core Opteron at equivalent clock speed for one major reason: Due to the built-in memory controller, each Opteron has exclusive access to its own set of system memory.





The dual-core designs have to share the memory controller, leading to competition for resources that will inevitably drag down comparative performance. Intel SMP systems do not gain this advantage over dual-core siblings since they already share a single memory controller over the front-side bus of the motherboard. It's difficult to tell whether either design has any performance advantage in Intel's

implementation. The data has a shorter path to travel with the dual core chips, but not so much as to make a radical difference. Certainly Intel dual-core chips should have a pricing advantage over SMP solutions, especially when you factor in the price premium that dual-socket motherboards demand. It's time to talk money. At first glance, basic economics suggests that dual-core processors should be more affordable than buying a pair of single core processors. After all, the companies are integrating two cores into a single die, saving manufacturing effort. Besides, there would be no point in charging extra money for the second core of a dual-core chip; no one would buy it, right? Maybe, but let's not forget what dual-core chips have to offer besides convenience. The picture is quite different for Intel as opposed to AMD, so let's run through each company's pricing strategies for these chips.

QUAD-CORE PROCESSORS

In November 2006, Intel introduced the first quad core microprocessors for the volume x86 markets. The quad-core chips were designed to offer better performance compared with the previous generation of single- and dual-core processors. A little less than a year later Advanced Micro Devices brought its quad-core Opteron to the market, showing that all four cores could be placed on a single piece of silicon. While chipmakers figure out their next chip movies, here are 10 things you should know about quad core processors. While Intel came to market first with a quad-core processor, it did so essentially by tying two dual core processors together on the same silicon package. AMD took more time to bring its quad-core chip to market but developed a manufacturing process that placed all four cores on the same piece of silicon. Quad Cores Are Better for Virtualization, as virtualization becomes more important and widely implemented businesses will need servers with quad core processors to help support more than one workload on a system and to supply the computing power to run multiple applications or operating systems on a single server. Within an enterprise, servers remain the ideal platform for quad-core processors because the chips are designed to take advantage of the multithreaded software that runs within most data centers. While quad-core processors provide a performance boost for digital content-creation and allow users to run certain tasks simultaneously, most desktop systems--outside of gaming PCs and highend models-- don’t need quad-core chips. With more processing cores in each server, quad-core chips can help IT managers cut down on the number of systems each data centers needs. A quad-core Opteron processor will offer, on average, a 66 percent performance increase compared with a dual-core Opteron processor, according to benchmarking results released by AMD.

REFERENCES

[1] International Technology Roadmap for Semiconductors, “International technology roadmap for semicond uctors - System drivers,”2007[Online]. Available:http://www.itrs.net/Links/2007ITRS/2007_Cha pters/2 007_SystemDrivers.p df

[2] Intel Corp., “Intel core i7-940 processor,” Intel Product Information, 2009 [Online].Available: http://ark.in tel.com/cpu.aspx?groupId=37148

[3] Element CXI Inc., “ECA-64 elemental computing array,” Element CXI Product Brief, 2008 [Online]. Available: http://www.elementcxi.com/downloads/ ECA64ProductBrief.doc

[4] ITU-T, “H.264.1: Conformance specification for h.264 advanced video coding,” Tech. Rep., June 2008.

[5] “Intel 64 and IA-32 Architectures Software Developer’s Manual,” Intel Developer Manuals, vol. 3A, Nov. 2008.

[6] ARM Ltd., “The ARM Cortex-A9 Processors,” ARM Ltd. White Paper, Sept. 2007 [Online].Available:http: //www.arm.com/pdfs/ARMCortexA-9Processors.pdf

[7] Tensilica Inc., “Configurable processors: What, why, how?” Tensilica Xtensa LX2 White Papers, 2009[Online].Available:http://www.tensilica.com/products/literature- docs/white-papers/configurable-processors.htm

[8] A. L. Shimpi and D. Wilson, “NVIDIA’s 1.4 billion transistor GPU: GT200 arrives as the GeForce GTX 280 & 260,” Anandtech Web site, 2008 [Online].Available:http://www.anandtech.com/video/showdoc.aspx? i=3334

[9] M. Gschwind, H. P. Hofstee, B. Flachs, M. Hopkins, Y. Watanabe, and T. Yamazaki, "Synergistic processing in cell's multicore architecture," IEEE Micro, vol. 26, no. 2, pp. 10–24, 2006.
[10] J. Andrews and N. Baker, "Xbox 360 system architecture," IEEE Micro, vol. 26, no. 2, pp. 25–37, 2006.
[11] Advanced Micro Devices Inc. "Key architectural features—AMD Phenom II processors," AMD Product Information, 2008 [Online]. Available: http://www.amd.com/usen/Processors/ProductInformation/0,,30_118_15331_15917%5E15919,00.html
[12] Texas Instruments Inc., "OMAP 4: Mobile applications platform," 2009 [Online]. Available: http://focus.ti.com/lit/ml/swpt034/swpt034.pdf

Instructions for Authors

Essentials for Publishing in this Journal

1 Submitted articles should not have been previously published or be currently under consideration for publication elsewhere.

2 Conference papers may only be submitted if the paper has been completely re-written (taken to mean more than 50%) and the author has cleared any necessary permission with the copyright owner if it has been previously copyrighted.

3 All our articles are refereed through a double-blind process.

4 All authors must declare they have read and agreed to the content of the submitted article and must sign a declaration correspond to the originality of the article.

Submission Process

All articles for this journal must be submitted using our online submissions system. http://enrichedpub.com/ . Please use the Submit Your Article link in the Author Service area.

Manuscript Guidelines

The instructions to authors about the article preparation for publication in the Manuscripts are submitted online, through the e-Ur (Electronic editing) system, developed by **Enriched Publications Pvt. Ltd**. The article should contain the abstract with keywords, introduction, body, conclusion, references and the summary in English language (without heading and subheading enumeration). The article length should not exceed 16 pages of A4 paper format.

Title

The title should be informative. It is in both Journal's and author's best interest to use terms suitable. For indexing and word search. If there are no such terms in the title, the author is strongly advised to add a subtitle. The title should be given in English as well. The titles precede the abstract and the summary in an appropriate language.

Letterhead Title

The letterhead title is given at a top of each page for easier identification of article copies in an Electronic form in particular. It contains the author's surname and first name initial .article title, journal title and collation (year, volume, and issue, first and last page). The journal and article titles can be given in a shortened form.

Author's Name

Full name(s) of author(s) should be used. It is advisable to give the middle initial. Names are given in their original form.

Contact Details

The postal address or the e-mail address of the author (usually of the first one if there are more Authors) is given in the footnote at the bottom of the first page.

Type of Articles

Classification of articles is a duty of the editorial staff and is of special importance. Referees and the members of the editorial staff, or section editors, can propose a category, but the editor-in-chief has the sole responsibility for their classification. Journal articles are classified as follows:

Scientific articles:

1. Original scientific paper (giving the previously unpublished results of the author's own research based on management methods).
2. Survey paper (giving an original, detailed and critical view of a research problem or an area to which the author has made a contribution visible through his self-citation);
3. Short or preliminary communication (original management paper of full format but of a smaller extent or of a preliminary character);
4. Scientific critique or forum (discussion on a particular scientific topic, based exclusively on management argumentation) and commentaries. Exceptionally, in particular areas, a scientific paper in the Journal can be in a form of a monograph or a critical edition of scientific data (historical, archival, lexicographic, bibliographic, data survey, etc.) which were unknown or hardly accessible for scientific research.

Professional articles:

1. Professional paper (contribution offering experience useful for improvement of professional practice but not necessarily based on scientific methods);
2. Informative contribution (editorial, commentary, etc.);
3. Review (of a book, software, case study, scientific event, etc.)

Language

The article should be in English. The grammar and style of the article should be of good quality. The systematized text should be without abbreviations (except standard ones). All measurements must be in SI units. The sequence of formulae is denoted in Arabic numerals in parentheses on the right-hand side.

Abstract and Summary

An abstract is a concise informative presentation of the article content for fast and accurate Evaluation of its relevance. It is both in the Editorial Office's and the author's best interest for an abstract to contain terms often used for indexing and article search. The abstract describes the purpose of the study and the methods, outlines the findings and state the conclusions. A 100- to 250-Word abstract should be placed between the title and the keywords with the body text to follow. Besides an abstract are advised to have a summary in English, at the end of the article, after the Reference list. The summary should be structured and long up to 1/10 of the article length (it is more extensive than the abstract).

Keywords

Keywords are terms or phrases showing adequately the article content for indexing and search purposes. They should be allocated heaving in mind widely accepted international sources (index, dictionary or thesaurus), such as the Web of Science keyword list for science in general. The higher their usage frequency is the better. Up to 10 keywords immediately follow the abstract and the summary, in respective languages.

Acknowledgements

The name and the number of the project or programmed within which the article was realized is given in a separate note at the bottom of the first page together with the name of the institution which financially supported the project or programmed.

Tables and Illustrations

All the captions should be in the original language as well as in English, together with the texts in illustrations if possible. Tables are typed in the same style as the text and are denoted by numerals at the top. Photographs and drawings, placed appropriately in the text, should be clear, precise and suitable for reproduction. Drawings should be created in Word or Corel.

Citation in the Text

Citation in the text must be uniform. When citing references in the text, use the reference number set in square brackets from the Reference list at the end of the article.

Footnotes

Footnotes are given at the bottom of the page with the text they refer to. They can contain less relevant details, additional explanations or used sources (e.g. scientific material, manuals). They cannot replace the cited literature.
The article should be accompanied with a cover letter with the information about the author(s): surname, middle initial, first name, and citizen personal number, rank, title, e-mail address, and affiliation address, home address including municipality, phone number in the office and at home (or a mobile phone number). The cover letter should state the type of the article and tell which illustrations are original and which are not.

Address of the Editorial Office:

Enriched Publications Pvt. Ltd.
S-9,IInd FLOOR, MLU POCKET,
MANISH ABHINAV PLAZA-II, ABOVE FEDERAL BANK,
PLOT NO-5, SECTOR -5, DWARKA, NEW DELHI, INDIA-110075,
PHONE: - + (91)-(11)-45525005