Post Boost Track Processing Using Conventional DBMS Software

 Aniruddha Basu

*Defense Research & Development Laboratory, Hyderabad*

 Email: basu\_chinu@yahoo.com

*Abstract*

***The design of Air Defence (AD), Command Control system is very challenging and has been treated with basic methodologies. The present design is associated with unstructured and uncorrelated data with huge lines of code*. *Hence an attempt was made for a better simplified (Database Management System) DBMS software data access methodology, on which futuristic command control system can rely. A typical Command Control Computer communication & Intelligence (C4I) subsystem algorithm was undertaken for prototype development on DBMS data access methodology. Development involved existing algorithm in C++ implemented code of post boost tracking was designed in Oracle database using Solid State drive. The purpose of this development was to find the advantage using online fast processing with a new upcoming methodology using basic Oracle database software. Used kinematic parameters generated from dynamic processing in Oracle database for determination of target Flight vehicle (FV) and decoy during separation from post boost tracks (PBTs). Earlier DBMS data access was never thought of for track processing application due to its slow disk access involving more time for transaction processing which has now being improved with the use Solid state drive apart from dynamic usage of new DBMS methodology.* *This paper presents a brief experimentation on the database (DB) methodology.***

**Keywords:** Cluster, Flight Vehicle**,** Decoy, BMD, RDBMS, DBMS, SQL, real-time, embedded

1. INTRODUCTION

The primary objective of this paper is to explore track processing capability with RDBMS Oracle Data Base software. The goal is to reduce the complexity of application development and make it more simplified resulting from the handling of data by DBMS in real time. Further objective was to process it in a solid state drive to minimise the time

Constraints with this approach.

Once this goal is achieved it will further help to realise futuristic heterogeneous command control architecture with better interoperability solution. To solve major challenge in data standardization is to make standards general enough to apply to all services and still meet the requirements of the individual specific services. Proposed architecture being industry standard professional skills and manpower can be easily available than present one.

Present experiment was made on host Linux environment on which Oracle 10.2 DB was installed with both HDD and SSD. The design used on line dynamic data of track messages at the host, processed the message and decoded to engineering Earth centric value and kept in DB repository. The database objects (Tables, views, triggers, procedures) that executed at RDBMS kernel, providing robust performance and functionality.

Features of real time DBMS often referred fast DBMS transaction within deterministic criterion are still unexplored in mission critical network centric application. Recently various Command Control application of Air Defense system have started using these commercial / open system databases having conventional DBMS software for supporting data processing modules at the application host to enable better improvement in data integrity compared to flat, unrelated and unmanaged data structures. Presently US Navy’s multi input tracking and control system 1 runs in a multitasking operating system but not a real-time operating system by the deterministic criterion. Hence optimizing conventional DBMS with Solid State Drive hardware will ensure better design and improved dynamic track kinematics within deterministic criterion. Development have started in Air Defense system using Oracle database / other open system databases having conventional DBMS software for supporting host data processing modules to enable better improvement in data integrity compared to flat, unrelated and unmanaged data structures. Hence optimizing conventional DBMS with SSD will comply better design and improve dynamic track kinematics.

During decoy target separation Flight vehicles (FVs) and decoys, debris from PBTs, are created and air situation processing becomes complicated. The threat initially consists of closely spaced objects. Hence it is required to identify the actual FV in the closely space scenario. The association process can only estimate the state vector of tracks and maintain it but cannot identify FV and decoy tracks. There is high volume of data access and computational requirement of kinematic parameters while dealing with the tracks processing for identification of these tracks of decoys which travels in parallel along with FV track.

A design and experiment was undertaken with mission specific deterministic criteria the description and detail report of which is submitted in the following Paragraphs given below.

1. LITERATURE REVIEW

Some studies on application of database architecture and experimentations for mission critical applications, tactical Defense application around the Globe have been carried out.

 Ceruti and Gessay(1998)**2**, publishedpaper which has given a broader outline and comparison in order to evaluate the state of new data-access technology, methods and architectures, to explore the systems and software developed within the C4I sector that can use this technology and to recommend an architecture on which to base the framework of data-access methodologies into which Naval C4I systems could evolve.

 Layman and Daly (2002)**3** described in their Public report organized by Naval Research Laboratory, US Washington, DC. This paper reports the progress on the US Navy’s Embedded Simulation Infrastructure (ESI) Program implementation and management of embedded models and simulations for C4I. The Defense Information Infrastructure Common Operating Environment (DIICOE) is the framework upon which many modern U.S C4I systems are developed.

 Ceruti (2003)**5**, reported in his paper explores challenges facing information system professionals in the management of data and knowledge in the Department of Defense (DOD),USA, particularly in the information systems utilized to support C4I. Para 8.2 of this report specifies architecture of dynamic database of data-driven collection management and situational estimation.

 McDaniel and Schaefer (2003)**1** published report used algorithms, data characteristics, and scenarios from deployed and R&D systems. The results do show that even under intense scenarios and with massive fusion on a general purpose medium performance off-the shelf computer using a non-real time operating system, the embedded DBMS can perform adequately.

 Shane and Hollenbeck (2004)**4** described in their paper a working prototype system which can help solve the problem for US department of Defence and potentially for multinational forces by incorporating a national repository and intelligent automation system using database in ACSIS system. The ACSIS is using Oracle database because Oracle is the only DBMS which incorporates EU Common Criteria certified multi-level security features, reinforcing the team’s original choice of a DBMS in designing the data model.

 Above studies explores views on data access architecture on various mission critical applications on C4I and network centric areas to enable the DOD to meet the present and future information management challenges. It is to infer that while most apply in organizational contexts, the relationships derived from the view by McDaniel D. and Schaefer G. (2003)**1** is probably more appropriate for a specific application of mission critical command and control. The present paper submitted is a specific mission critical application on a non real time operating system and database where the methodology adopted, the main analysis technique employed are the main findings of the study.

1. METHODOLOGY AND DESIGN

In client server of Oracle Data base model, transactions are accompanied between server and client DB where separate client installati3on is done at client host. In the present experiment instead of separate client installation, transaction was made Oracle instant client where operating (Linux or other OS) can communicate with the server through OCCI library and host linker. Data base instant client can be connected to multiple hosts having OS specific OCCI libraries. Thus the query, transactions to the data base server from host OS were accompanied by embedded direct pass through method **2**. This architecture is shown in figure 1.

Further Oracle database is more convenient and system built in. One / more processes for each connected user or one or more processes of other user from hosts can be handled and shared by client requests with automatic shared server process in database server. Augmentation of multi processes / multi threading automated features in Oracle database **6** has made it more convenient for real time processing.

Based on this concept and methodologies application design was made in C++ software in Linux system. Incoming track raw message received from sensor during trial run was captured and decoded to Earth Centric coordinate of engineering values at the host system and simultaneously inserted into Oracle database base table. Database processing is made for conversion to East North Vertical coordinates and processing for Flight vehicle and decoy booster. This database process diagram is shown in Fig. 2 .

**Figure 2**

Database design and flow of database operation for determination of

FV and decoy

 **Dynamic storage, Update at T0, T\_1 …..**

**Data insertion to DBT1**

**----------- (ECEF) --------**

**Data insertion to DB T2 (ENV)**

**View of DB T 2 (All latest updates) Tnow, Tnow-1…**

**Data insertion to DB T 3**

**FV, Decoy info. To T 4**

**(Max. and next to max. Energy)**

**Procedure (New Rep.)**

**Procedure**

**T1 ends**

**T1 starts**

**Trigger**

Decoded Msg. from host system

Sensor level cluster checks are done during

**Figure 3, Target tracking configuration**

reception of information. The track clusters are associated from same type of radar sources and also from different local sensor sources. Associated and fused data from these two types of sensor sources are again processed by central level of processing. Sometimes fusion of data, received from numerous radar sources, (Ref. Fig. 3) based on selection of priority (main) source, performs integration of such parameters like classification of cluster tracks, state identity on target because these data are considered to be the most accurate in terms of tracking and conducting of combat operations. This approach provides tracking of fused track which is good enough. Especially good track tracking is provided, if the priority source performs a complete cycle of trajectory data processing, including data fusion and smoothing.

Present experiments on trial data was smoothed fused trajectory data, received from radars of the same type and used the method of integrated processing. The integrated processing of trajectory data includes estimation of target position and motion parameters on the basis of all received data, weighted with account for the covariance matrix of errors in measurement of target coordinates and motion parameters. Hence the present cluster design in database has different concept than the sensor level cluster group. Present cluster grouping is done in database from the fusion output from typical command control system where set of targets and measurements can be divided into sets of independent clusters in post phase, then a great deal of simplification may result. Instead of one large tracking problem, a number of smaller tracking problems can be solved independently. Since the amount of computer storage and computation time grows exponentially with the number of targets, this can have an important effect in reducing computational requirements. If every target group cluster could be separated into its own individual cluster group, the design architecture can be simplified in database and computational time can be reduced with solid state drive even though processing of tracks can grow linearly with the number of targets (Fig 4 below).

240

230

220

210

200

190

180

170

160

150

140

130

120

110

100

90

80

70

60

50

40

30

 20

10

 0

In the Present design, the FV and decoy has been determined where cluster group is completely defined by specifying the set of parameters within group.



**New Track (raw) (ECEF) Y/N ?**

**Convert to ENV** ECEF

**1st clust** ggrp. Grp.

**Last clust.** ggGGGrp.(n) + 1

 **Check Kinematic** conditions

**Failed track** Failed clusters

**Passed Trk** Passed clusters

 **Alt, Av.energy, Highest (FV), next** Highest En. En.(decoy)

**Figure 5**

Flowchart of cluster group & Fv identification

 Flowchart of cluster group & FV determination.

**End**

**Yes**

**No**

**R**

**e**

**t**

**u**

**r**

**n**

**N**

***e***

**x**

***t***

**N**

**e**

**w**

**C**

**l**

**u**

**s**

**t**

**e**

**r**

**Old Trk ?** **Yes /No**

 Growth Of Tracks during Post

 Boost Phase

 N

**Figure 4,** **Growth of Track Ids with cluster group**

**V. Cluster Group and Flight / Target Vehicle Identification.**

**Basis of FV Decoy Booster Determination from Cluster Group**

**Track Id Rep. time Clst. Kinematic Energy**

 **Grp status**

**245 643174336 2 P 116800**

 **247 643189520 2 F 117023**

 **245 643234304 3 P 116900**

 **247 643709420 3 P 117210**

 **248 643709529 3 F 119250**

 **245 -- Update 643235103 3 121010**

 **247 -- Update 643812050 3 118020**

 **245 643237304 4 P 121349**

 **247 643949520 4 P 118129**

 **248 643709520 4 P 121900**

 **249 643934901 4 F 113660**

 **245 -- Updat 643239634 4 121349**

 **247 -- Update 643953205 4 118240**

 **248 - -Update 643735100 4 122200**

**Figure 6,** **Determination of FV and booster decoy**

**Max. and next to max. energy of Track ids in last clus Gp. (P) and track update parameters from clus. start time till next cluster group formation are determined in DB dynamically.e.g. FV and booster decoy in cluster Gp. 3 and next group 4 has been determined till next Group creates in trajectory propagation.**

 **Final FV and Decoy (B) determination**

**Fv B Fv\_en B\_en Fv\_Time B\_Time Grp.**

1. **247 121010 118012 643235103 643812050 3**

**248 245 122200 121349 643735100 643239634 4**

Even though there are number of tracks reported during post boost separation and form various parallel trajectories but identification process of Fv and booster decoy are started while new track is reported. The track kinematic conditions within cluster group are checked at the instant of new track report with all previous tracks. Tracks passed in kinematic checks in last cluster group and its update are considered for Fv, decoy determination as shown in flow diagram of Fig. 5 which are processed in Oracle database.

Dense Id. Clst.grp. pair

**Figure 7, Track Cluster pair distribution in Energy Time Graph**

Figure 6 shows the basis of cluster group creation and current track updating parameters within the group. Kinematic checks are done to ensure whether tracks are falling in same ballistic group or not. emanating from same source and if is passed are processed for Fv on the basis of energy as per equation derived. Hence outputs of a current cluster group provide Fv / Target vehicle obtained with next nearest proximity decoy booster within all parallel tracks during post boost mid course track identification.

The Figure 7 shows below the distribution of track id and cluster group pair of sample distributed in a Energy time plot of one R&D trial mission.

Figure 7 shows track id and cluster (group) pair distribution at various energy level within a time zone or cluster group. This parameter can be identified at any point of a trajectory profile of interest using dynamic knowledge repository and information retrieval from database to host system resulting from an improved methodology.

**Figure 5**A Proj. of vector A on vector B.

Let’sLet’s consider a non collinear vectors =(xa,ya,za) and =(xb,yb,zb); let α be the angle between them. The

Fig. 8 Proj. of vector a

On b.



projection pra\_b of the vector on the vector is calculated as 

**Figure 9** Vector projections from

Reference plane is used for determining kinetic parameters.

This formula is obtained from correlation

XL

*Local*

*Meridian*

ZE

XE

P1(x1 ,y1 ,z1)

**(0;0;0)**

N-pole

ZL

YL

YE



 **Figure-10 ECEF frame with ENV**

 **coordinate**

**VI. Track Energy Determination**

Algorithm used in simulation study for calculation of highest energy calculation **7** (P 18) has been used for calculation of track energy at the latest instant of track comparison for FV and number of decoys in Oracle database and the two samples amongst cluster group has been evaluated for FV and next decoy.

The average energy of a Flight vehicle which is equal to the sum of its kinetic energy (**Ek**) and potential energy**(Ep)** and is constant during flight**.** VT is the target velocity, **g** the gravitational

 constant, R the radius of the earth earth and RT the distance between the target and the center of the earth the normalized (point mass), VT is the target velocity, **g** the gravitational constant.

Semi major axis ,

Semi minor axis 



-Latitude

at point P1.

 Where  – Range of the object in equatorial plane,

Hence the Energy can be written as

 

 The Energy derived as per equation (2) is processed in Oracle database at every update of parameters. Due to error associated with the track parameters, the FV and decoy can be identified distinctly can if FV energy>=1.3 booster decoy energy. The period of a cluster group is the time for forming a group during new track measurement and its update period. Hence it will be helpful for BMD to obtain the FV information for engagement at the required moment of the mid course trajectory.

If e1 and e2 are eccentricities of the Earth ellipsoid,

then.=0.00669437999, =0.00673949674222

  Target velocity vector in a earth P1 is the position vector  of Centered Cartesian coordinate system.



**Figure 11 A section of Range Time plot of simulation showing track Ids in trajectory**

|  |  |  |  |
| --- | --- | --- | --- |
| **FLIGHT VEHICLE** | **DECOY** | **FLIGHT VEHICLE** | **DECOY** |
| **TIME(ms)** | **He MAX** | **ID** | **He MAX** | **ID** | **TIME** | **He MAX** | **ID** | **He MAX** | **ID** |
| **113.87** | 115850 | 245 | ------ | ------- | **116.57** | 115810 | 246 | 114780 | 249 |
| **114.37** | 116500 | 247 | 115840 | 246 | **116.77** | 115810 | 246 | 114660 | 249 |
| **114.07** | 114.07 | 247 | 115850 | 245 | **116.97** | 115810 | 246 | 114540 | 249 |
| **114.97** | 115950 | 247 | 115830 | 246 | **117.17** | 115810 | 246 | 114420 | 247 |
| **115.17** | 115830 | 246 | 115790 | 247 | **117.27** | 115810 | 246 | 114370 | 249 |
| **115.27** | 115830 | 246 | 115830 | 246 | **117.47** | 115810 | 246 | 114260 | 248 |
| **115.47** | 115550 | 247 | 115820 | 246 | **117.57** | 115810 | 246 | 114210 | 249 |
| **115.67** | 115400 | 249 | 115820 | 246 | **118.57** | 115810 | 246 | 113740 | 249 |
| **116.07** | 115110 | 249 | 115820 | 246 | **118.77** | 115810 | 246 | 113650 | 249 |
| **116.27****116.37** | 114970115820 | 247246 | 114970114910 | 247249 | **118.97****119.27** | 115810115810 | 246246 | 113570113450 | 249249 |

**Table 1**, **Flight Vehicle and Decoy of simulation**

From simulation (Ref. fig 11 above), the result is tabulated in Table 1 for a flight vehicle and its decoy/booster using Oracle database design.

~~Table 1~~

VII. **Results**

 The results of experiments of R&D trials and simulation carried out in the new methodology and design referred in figure 1,2,3,5 above are found satisfactory.

 **Ps Time**

 **in**

 **25**

 **Micro**

**Micr Sec**

 **Scale**

**Track Id and Processing Time plot of R&D Trial**

**Figure 12 Comparisons SSD, HDD**

**Track Id**



|  |
| --- |
| **Table 2: Database Processing time**  |
| id | id\_cnt |  proc\_t Tr/row(mics)  |
| 245 | 361 | 80275 |  222  |
| 247 | 188 | 66600 |  354  |
| 248 | 191 | 86425 |  452  |
| 249 | 57 | 23300 |  409  |
| 250 | 169 | 75800 |  449  |
| 251 | 35 | 14950 |  427  |
| 252 | 358 | 184700 |  516  |
| 253 | 57 | 26600 |  467  |
| ---- |  ----- | ------ |  ------ |
| 305 | 7 | 7175 |  1025  |
| 307 | 139 | 130850 |  941  |
| 308 | 59 | 54100 |  917  |
| 309 | 344 | 361700 |  1051  |
| 313 | 145 | 145450 |  1003  |
| 315 | 176 | 184300 |  1047  |
| 316 | 204 | 209375 |  1026  |
| 317 | 15240 | 48160925  |  3160 |
| T=43 | 21657 | 53082700 |  2452 |

  ~~Table2~~

This is observed that process time in database shown in table 2 called through SQL procedure (Ref. figure 1) above, the total time for all those new 43 track processing and further update track maintenance were counted for 21657 track update occurrences in the database with a total processing time was 53082700 micro seconds thus average transaction time / row processing is 2452 microseconds in the Linux system with SSD Drive for particular R&D trial.

The processing time in Solid State Drive rather than disk access has greatly reduced. This paper describes the benefit to use a DBMS system and shows results to reflect how optimized non real time data base can even better support the command control application with fast processing.

VII.CONCLUSION AND FUTURE SCOPE

The commercial Database (Oracle DB) has achieved the near real-time database (RTDB) property whose operations execute with predictable response, and with application-acceptable levels in addition to timely execution of transactions. Powerful information access by advanced SQL pass-through method to database **2, 5** and techniques with C++ program & tools can provide a foundation for better air situation processing. Because the design is transparent, many types of air situation algorithm can be simplified with this methodologies and operate in an integrated manner over a broad spectrum of information processing. The data base resources and objects can be shared that they will interoperate with other integrated system in a collaborative manner. Present platform centric war fighting can be realized and transformed to Network centric war fighting based on information fusion and the methodology presently used is based on dynamic data access against track situation awareness. As the complexity of real-time systems and application is going up, the amount of information to be handled by real-time systems increases, motivating the need for database and data service functionality (as opposed to ad hoc techniques and internal data structures), the results of the experiments provide evidence and argues that; apart from high-performance embedded DBMS’s which have much higher data access and quick analysis capability than application dependent data structures, a non real time database can also be optimized and designed which can perform under intense scenarios of the air situation process on a general purpose medium performance computer using a non-real-time operating system, the embedded DBMS can perform adequately better with advanced SQL capabilities with proprietary implementation of stored objects (variables, procedures, functions, Trigger etc.) that execute at the RDBMS kernel, providing robust performance and functionality. This type of RDBMS databases (e.g. Oracle) can be optimized with Solid State Drive with non real time based Operating system to go in a long way. With more powerful processors, a real-time operating system, and a distributed computing environment, the embedded DBMS could be expected to perform even better.

ACKNOWLEDGEMENT

The author would like to express his sincere appreciation for embedded programming support by collogues of DCCT/C4I, of DRDL. This application would not have completed

without their support, discussion. This division has not only intelligent

programmer, but domain expert in the field of BMD. The knowledge of that has contributed much to the successful development of cluster processing algorithms.

REFERENCES

[1] McDaniel D. and Schaefer G. (2003), Real-time DBMS for Data Fusion, proceedings of the sixth international conference ,IEEE conference publication, Vol. 2, pp 1334-1341.

[2] Ceruti M. G and Gessay S. A (1998), White paper on Next-Generation Data-Access Architecture for Naval C4I Systems, pp 1-24.

[3] Layman G. and Daly J. (2002), C4I Tactical Applications Utilizing Embedded Simulations, pp 1-15, organized by Naval Research Laboratory, Washington, DC

[4] Shane R. and Hallenbeck P. (2004), Proceedings of 9th International Command and Control Research and Technology

[5] Ceruti Marion G (2003), (pp1-12), Report on “Data Management Challenges and Development for Military Information

Systems, DOD , Govt. of US”, Published in *IEEE Transactions on Knowledge and Data Engineering (TKDE).* Vol. 15, No. 5, pp. 1059-1068, September/October 2003.

 [6] Oracle documentation for multi process and multi threading,

<http://www.csee.umbc.edu/portal/help/oracle8/server.815/a67781/c07procs.htm>

<https://docs.oracle.com/database/121/CNCPT/process.htm#CNCPT124>

[7] Chang C B, (1979-1), Technical note Optimal State Estimation of Ballistic trajectories with Angle-Only Measurements, Ballistic Missile Defense Office Department of the Army,. MA **22133.**