Defence Science Journal, Vol. 66, No. 2, March 2016, pp.130-137, DOI : 10.14429/dsj.66.9244

Post Boost Track Processing Using Conventional DBMS Software

Aniruddha Basu

Defence Research and Development Laboratory, Hyderabad - 500 058, India
E-mail: basu_chinu@yahoo.com

ABSTRACT

The design of air defence, traditional command control system is very challenging which has been used with basic methodologies. Traditional design is associated with unstructured and uncorrelated data and requires huge lines of code using hard disk drive (HDD) in the system. Hence an attempt was made for a better simplified database management system (DBMS) software data access methodology, which processed the incoming airborne data, message in RDBMS database to achieve full automation on real-time. The transaction is accomplished through SQL pass through method from the host decision making system into database. An algorithm of track identification during midcourse track separation was undertaken for prototype development on DBMS data access methodology. In this methodology Oracle C++ calls interface embedded query call was used from the host interface system. The purpose of this development was to find a comparison of online process timing between HDD and SSD using commercial database, and to evaluate performance of dynamic processing of RDBMS Database for identification of target vehicle and booster after separation. Produced experimentation results from improved performance of the proposed methodology on which futuristic command control system can rely.

Keywords: Oracle C++ connect interface, relational database management system, object oriented programming, target warhead, booster, computer communication and command control intelligence

1. INTRODUCTION

Generation of air situation information in a C4I is very critical and complicated task. Although there has been a tremendous improvement of information processing in C4I technology and its related architecture in modelling and simulation, interoperations across various platforms, and in many other areas in defence but core level design of air situation within a C4I architecture is unexplained and unexposed. An air situation is the monitoring of airspace by information processing from various sensor sources which is formed by correlation function of flying objects (track) corresponds to an object on display.

During air target vehicle separation in a mid course trajectory, the original target warhead is separated into various fragmented parts: Target warhead vehicle and boosters and debris. All these separated targets and fragment debris are initially closely moving in a parallel trajectory. Hence it is required to identify the actual target warhead in a closely spaced trajectory. The track association process can only estimate the state vector of tracks and maintain tracks state but cannot identify target warhead and booster tracks. There is high volume of data access and computational requirement to determine and compare of dynamic strobe value while dealing with the track identification process. While original target is separated into parts, the target warhead always maintains a highest trajectory than others although all fragmented parts are in close proximity.

The database designs involved receiving incoming track messages, to process and decode into engineering earth centric value and corresponding east north value and populated in DB repository. Cluster is created in database which is a logical number allotted at the instant of new track report. As soon as the cluster is created, the latest track updates are automatically included within the last group. Thus the entire previous and present cluster groups created are containing old track history and current track state. Cluster group may be presumed moving along with all separated fragmented trajectories. The identification process starts automatically considering tracks which are qualified after dynamic checks and stored in new/latest cluster group.

The primary objective is to explore track processing capability using RDBMS 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 (SSD) to minimise time constraints and compare the process time between SSD and hard disk drive (HDD).

The methodology proposed is based on Oracle RDBMS database (v10.2) and encapsulating API OCCI features of Oracle DB by OOP programming technology in Linux platform. Oracle Database is preferred for its wide ranging technical features, industry standard and proven real time architecture in many defence and military applications.

The proposed DBMS design architecture has not migrated

Received 04 November 2015, revised 30 January 2016, online published 22 March 2016

identical code design or schema matching between similar elements or data sources of traditional system except concept of dynamic strobe checks within the algorithm.

The experiments used trial data from real R&D mission and simulated data. Figure 1 shows the simulated data trajectory generated by using mathematical model. This data was processed in the Database program to evaluate the performance for the DBMS data access processing capability along with experiments of real time trial data.

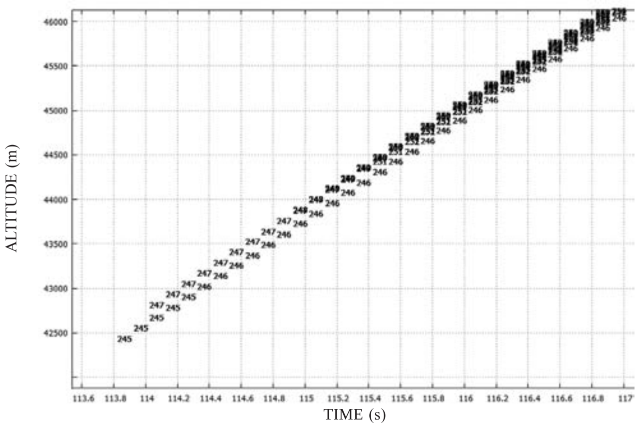

Figure 1. Simulated data trajectory with altitude *vs* time.

2. TRADITIONAL C[4]I PROCESSING ARCHITECTURE AND DBMS BENEFITS

Sensor level checks are done during reception of information. At the beginning track clusters are associated from same type of radar sensor sources and also from different local sensor sources. Associated and fused data from multi sensor sources are again processed by central level of processing. As shown in Fig. 2 sometimes fusion of data, received from numerous radar sources, 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.

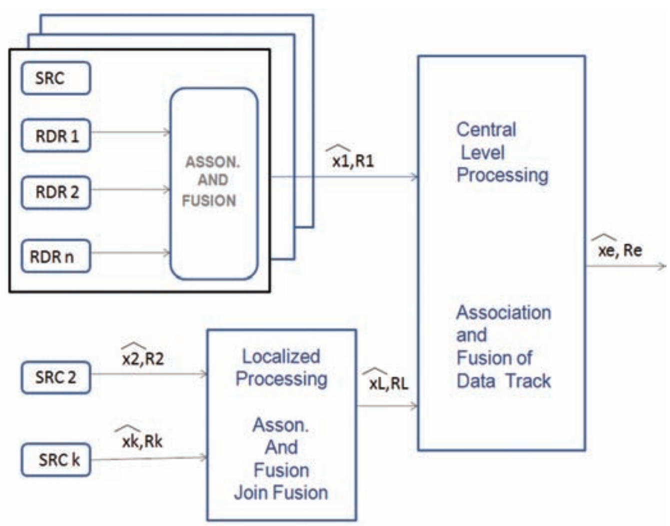

Figure 2. Target tracking architecture in a typical command control system.

Proposed experiment on trial data was smoothed fused trajectory data from command control, received from Radar sensors 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 reports.

Traditional and basic architecture process static and derived data from Flat files with various inter connected messages from heterogeneous subsystems. Data in a traditional architecture is not structured and interrelated. Members of various message structures are redundant and data are redundantly stored involving more complex, lengthy coding for internal data handling and more time for processing[1] than proposed architecture. The application requires dynamic memory estimation for workload at different points in time at each occurrence of new track report. The dynamic memory allocation cannot be handled automatically in the traditional software architecture. There is no automated storage space optimisation or dynamic memory control in the traditional system. Initialisation and rigorous testing is required in integrated mode, hence not quick deployable. It becomes very tedious and time consuming to locate and debug an overall system/subsystem level fault during integrated mode of operation. Whereas DBMS data access architecture has an improved feature in the proposed design by using Oracle database feature (v 10.2g). Apart from better data organisation and management, database can automatically optimise memory allocation by an automatic shared memory management which adjusts the size of the system global area components on the fly as the workload changes[2]. The database also controls the storage through automatic storage management features[3].

3. RELATED STUDY AND LITERATURE REVIEW

To evaluate the issues related to track processing in DBMS software and to review current solutions, following literature reviews are among many that was done:

Mark[4], *et al.* described the development of battle management language is a promising approach to interoperation of C2 and simulation systems. Here application used in database for joint consultation, command and control information exchange data model, developed and maintained by the multilateral interoperability program.

Siddiqui[5], *et al*. covered and discussed a service oriented architecture based common view interoperability approach by integrating different dispersed applications using Oracle enterprise service bus. They have introduced a common platform middleware that communicate on the bases of messaging, routing, invocation and mediation services.

Kilimci[6], *et al.* described their work in moving object temporal data using Oracle DBMS in space defence application. These developments in location management, uncertainty and indexing research will increase the command-control performance.

Xu[7], *et al.* tried to resolve and researched some bottleneck of certain missile weapon system simulation, simulating architecture, distributed simulation engine based, simulation database management based on Oracle Berkeley database. The simulating result indicates that above research has great promoted the usage and deployment of certain missile weapon system.

McDaniel[8], *et al.* 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.

Ceruti[9] explores challenges facing information system professionals in the management of data and knowledge in the Department of Defence, USA, particularly in the information systems utilised to support C^4I. Para 8.2 of the report specifies architecture of dynamic database of data-driven collection management and situational estimation.

Chang[10] derived the total energy content in a ballistic track. Proposed experiment has used this method to identify the tracks after separation.

Singh[11] discussed in his report the importance and requirement for a quick reaction time efficient data organisation. He pointed out the need for battle field information exchange for modern warfare system.

Studies explores views and implementations on data access architecture on various mission critical applications on C^4I and network centric areas. It is to infer that while most apply in contexts of modelling and simulation, interpretability platforms in command control application, the relationships derived from the view by McDaniel[8] is probably more appropriate for a specific air situation application within mission critical command and control. Proposed architecture, i.e. 'post boost track processing using conventional DBMS software' is a specific C^4I subsystem level application development using non real time operating system and database where simplification, data access methodology, present trend of DBMS access technique employed across the globe are the main findings of the study.

4. METHODOLOGY AND DESIGN

Transactions are accompanied between server and client where separate client database installation is done at client host. In the present experiment instead of separate client installation, transaction was made through API call of OCCI feature of Oracle database and embedded OOP program. The API SQL calls from single / multi users' from single or from heterogeneous host will create various client processes to the server. Thus the query call to the server from host OS were accompanied by embedded direct pass through method.

Some RDBMS database is more convenient for its system built in features like auto memory, space management, handling of multi processing multi threading, locking, triggers, procedures, etc. These client processes 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[12] has made it more convenient for real time processing.

Recently various command control application of air defence 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[8] runs in a multitasking operating system but not a real-time operating system by the deterministic criterion.

Based on this concept and methodologies, proposed application was designed in OOP software in Linux system where incoming raw track message received from sensor during trial run was captured and decoded to earth centric coordinate of engineering values and simultaneously into corresponding east north up coordinates in the database. Values obtained processed further for evaluation of target warhead vehicle and booster identification. This database process flow diagram is as shown in Fig. 3.

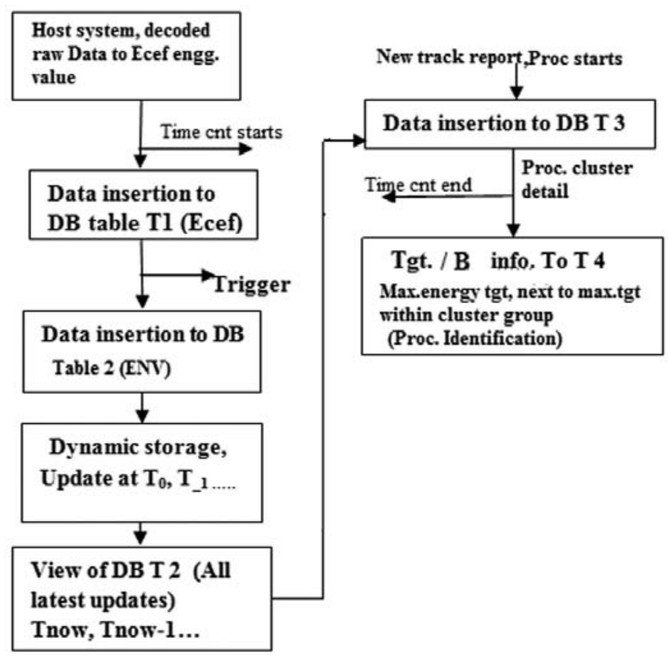

Figure 3. Database design and internal flow of track identification.

Present cluster grouping is done in database from the fusion output of a command control system where set of targets and measurements can be divided into sets of independent cluster group in post phase resulted a great deal of simplification using database. 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. Here track state parameters are bounded into its last 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. Figure 4 shows the track ids growing linearly with cluster group requiring more processing time.

Figure 5 shows the process sequence within the flow chart to check various conditions from track input at host to target determination at the database end.

Figure 6 shows the basis of cluster group creation and current track updating within a group. Dynamic strobe checks are done to ensure whether tracks are emanating from same source and if falling in same source of ballistic group, it is processed for target identification on the basis of energy as per Eqn. (10) derived. Hence outputs of a current cluster group provide target warhead vehicle along with nearest booster amongst fragments.

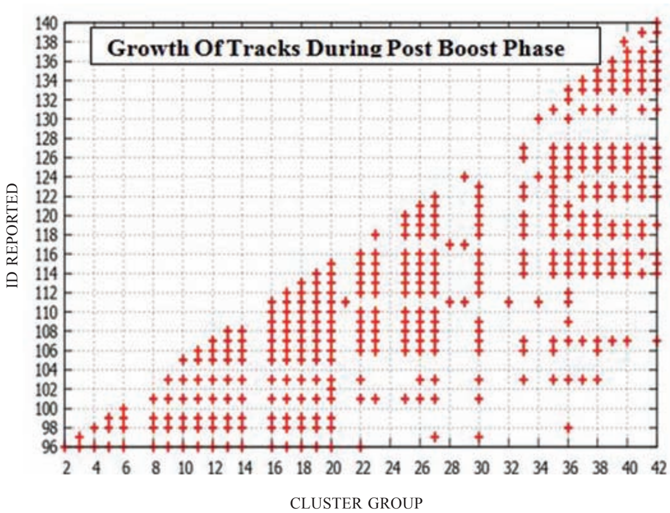

Figure 4. Growth of more tracks with increment of cluster Ids.

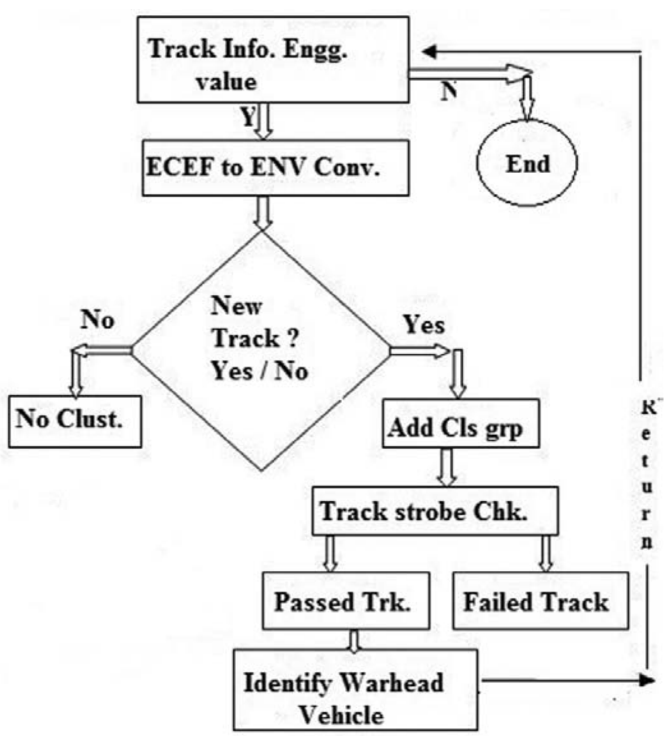

Figure 5. Cluster group formation and processing of target identification.

Track Id	Rep. time	Clst. Grp.	Status	Energy
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

Remarks

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 dynamicallv.e.g. tgt. and booster frag. in cluster Gp. 3 and next group 4 has been determined till next Group creates in trajectory propagation.

Tgt. warhead and Booster Determination

Tgt	B	T en	B_en	T_ Tgt	T_Boost	Grp.
247	245	121010	118012	643235103	643812050	3
248	245	122200	121349	643735100	643239634	4

Figure 6. Basis of highest flying target determination in a cluster group.

5. CLUSTER GROUP AND TRACK DYNAMIC CHECKS

Figure 7 shows the distribution of the pair of track id and cluster group in an Energy time plot of a trial R&D mission. This track can be identified at any point of a trajectory profile of interest using dynamic knowledge repository and information retrieval from database to host system.

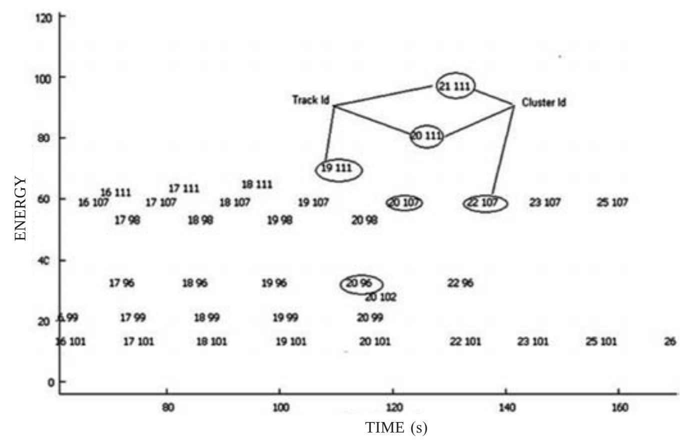

Figure 7. Energy *vs* time distribution of tracks in cluster group. The plot sample has been shown in track id and cluster group id pair.

The identification process of target warhead vehicle and booster starts with dynamic checks of separated track state whenever a new track is reported. The new track is compared with all previous tracks within the cluster group after separation to find out if the new tracks state is originating from same group source. The checks include comparing angle difference between new track with other existing track state, perpendicular distance of new reported track with others, distance between two tracks stat based on velocity vector components of new and other tracks. Tracks passed in dynamic checks in existing/ current cluster group are processed for target warhead, booster determination in the database as shown in flow diagram of Fig. 5.

Figure 8 shows the track state vectors on various planes. At any instant of time the tracks position and velocity state vector can be obtained. The difference of angles, perpendicular distance and velocity vector difference between the new reported track and existing tracks can be evaluated. The dynamic checks for the above three conditions can be determined by

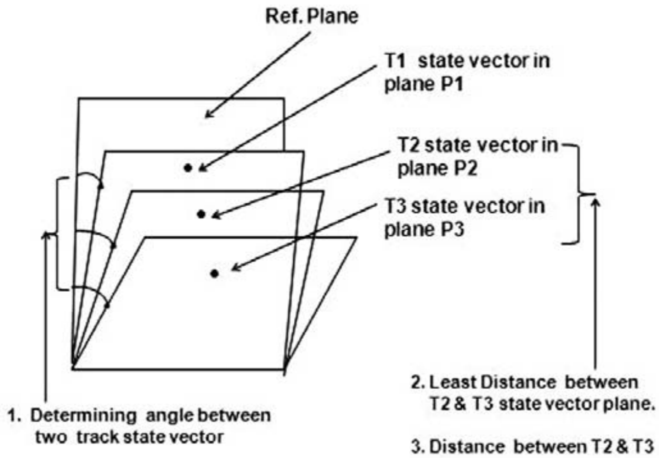

Figure 8. Dynamic checks of reported and existing track state vector.

the algorithm if the all the above conditions are satisfied to be within predetermined limit. Existing passed tracks within the cluster group can be included for target identification. The basic equation shows the projection vectors in terms of coordinate parameters of track.

$$pr_{a\_b} = |\vec{a}| \cdot \cos(\alpha) = |\vec{a}| \cdot \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} = \frac{\vec{a} \cdot \vec{b}}{|\vec{b}|} \quad (1)$$

$$or,\ pr_{a\_b} = \frac{x_a \cdot x_b + y_a \cdot y_b + z_a \cdot z_b}{\sqrt{x_b^2 + y_b^2 + z_b^2}} \quad (2)$$

$$pr_{a\_b} = \frac{x_a \cdot x_b + y_a \cdot y_b + z_a \cdot z_b}{\sqrt{x_b^2 + y_b^2 + z_b^2}} \quad (3)$$

where x_a, x_b, y_a, y_b, are the x and y position of vector a and b.

From the above equations, the projection in Eqn. (3) between vector states is derived in terms of position vector coordinates. Hence angle difference can be determined between two tracks. Similarly the perpendicular distance between two track vectors and velocity difference between two velocity vectors of tracks can be determined dynamically.

6. TRACK ENERGY DETERMINATION

Algorithm used in simulation study for calculation of highest energy calculation[10] has been used for calculation of track energy of target states in Oracle database and the two samples amongst cluster group has been evaluated for the average energy of a target vehicle which is equal to the sum of its kinetic energy (*Ek*) and potential energy **(***Ep***)** and is constant during flight.

$$HE = Ek + Ep = \frac{1}{2}V_T^2 + gR(\frac{RT - R}{RT}) \quad (4)$$

where V_T is the target velocity, g the gravitational constant, R the radius of the earth and R_T the distance between the target and the center of the earth the normalised (point mass), V_T is the target velocity.

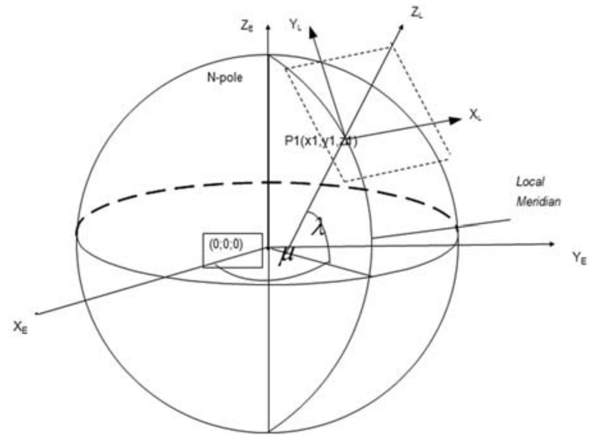

Figure 9. Ballistic track energy at point P1, ECEF and ENV frame.

Semi major axis (equatorial Earth radius) a = 6378137

Semi minor axis (polar Earth radius)

b = 6356752.314245179

$$e1^2 = \frac{a^2 - b^2}{a^2} = 0.00669437999$$

$$e2^2 = \frac{a^2 - b^2}{b^2} = 0.00673949674222$$

$e1$ and $e2$ – eccentricities of the Earth ellipsoid

$$p = \sqrt{X1^2 + Y1^2} \quad (5)$$

Range of the object in equatorial plane,

$$\theta = acrtg\left(\frac{Z \cdot a}{p \cdot b}\right) \ ..Theta \quad (6)$$

Latitude

$$\lambda = arctg\left(\frac{Z + b \cdot (e2)^2 \cdot \sin(\theta)^3}{p - a \cdot e1^2 \cdot \cos(\theta)^3}\right) \ ..-\ at\ point\ P1 \quad (7)$$

$$R = a\ /\ (1 - e^2 sin^2)^{1/2} \quad (8)$$

effective radius of earth at point $P1$

$$Alt = [p / cos\ ()] - R\ with\ reference\ to\ sea\ level \quad (9)$$

Target velocity vector in an earth

... in earth centric coordinate system, $V_T = \begin{bmatrix} \overrightarrow{Vx1} \\ \overrightarrow{Vy1} \\ \overrightarrow{Vz1} \end{bmatrix}$

P1 is the position vector of Earth centered

coordinate system $\begin{bmatrix} \overrightarrow{x1} \\ \overrightarrow{y1} \\ \overrightarrow{z1} \end{bmatrix}$

Hence total ballistic track energy derived from Eqn. (4)

$$HE \ \alpha \ (ALT + \frac{Vx1^2 + Vy1^2 + Vz1^2}{2g}) \text{ Meter} \qquad (10)$$

The energy derived as per Eqn. (10) is processed in Oracle database. Due to error associated with the track parameters and closed proximity, the target and booster can be identified distinctly if target energy >= 1.3 booster energy. The period of a cluster group is the time covered when new track is reported and till next new track reports. Hence it will be helpful for target information at every segment of cluster group during the mid course trajectory.

7. RESULTS

The results of experiment of R&D trials is as shown in Table 1. The table depicts process time in database for individual track, its number of occurrences, total process time and transaction time per row. A total of 43 tracks were reported. It is seen that transaction per row of each track and average transaction time is not more than 3-4 ms. For total 43 tracks and its occurrences of 21657 time's total processing time was 53082700 micro seconds. The trial runs used Linux system with SSD and HDD. The result shown was conducted using SSD.

Table 1. Database transaction performance in real time mission data

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

Result is as shown in Table 2 which was derived from Figure 10 (energy *vs* time) for the experiment carried out on simulation mode in same database with SSD. Here the tracks have shown the total energy content at a particular time instant. Hence at every time instant or within a time interval, principal track vehicle and next below booster track can be determined and identified. However there was some limitation for the experiment as it was not a migration of a subsystem design in Database, various others associated boundary conditions could not be considered.

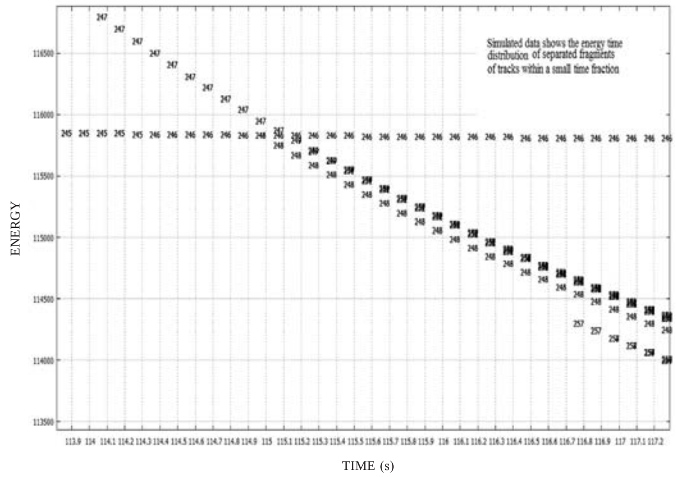

Figure 10. Separation of original tracks in to fragments. In energy time plot the samples are shown in track ids. Energy *vs* time plot derived from simulation run of Fig. 1.

Figure 11 shows that the processing time in solid state drive rather than disk access has greatly reduced. This paper describes the benefit of using a DBMS system and shows results to reflect how optimised non real time data can even better support the command control application with fast processing.

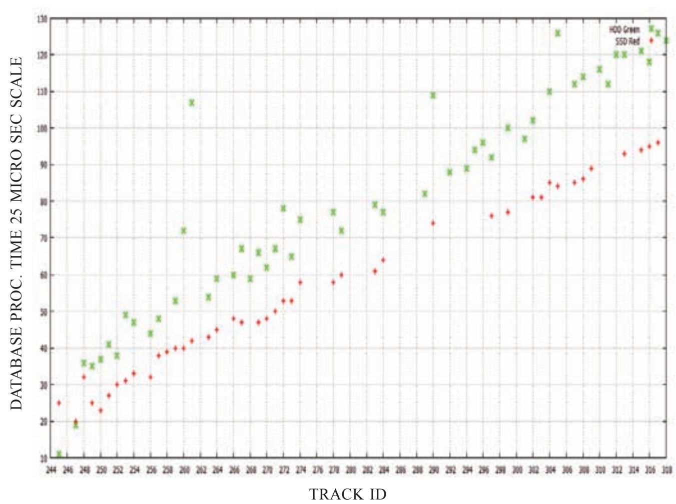

Figure 11. A comparison of process time in solid state and hard disk drives. The samples have been plotted in terms of process time.

Hence the experiment for identification in an air situation real time scenario in commercial Oracle database with improved timing performance was achieved.

8. CONCLUSION AND FUTURE SCOPE

The commercial database (Oracle DB) has achieved the near real-time processing property whose operations execute

Table 2. Time wise warhead vehicle and booster fragments identification in simulation run as shown in Fig. 10

WH Vehicle			Booster frag.		WH vehicle			Booster frag.	
Time (s)	**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	114970	247	114970	247	118.97	115810	246	113570	249
116.37	115820	246	114910	249	119.27	115810	246	113450	249

with predictable response, and with application-acceptable levels in addition to timely execution of transactions. Powerful information access by advanced SQL OCCI calls to database from encapsulated OOP (e.g. C++ program) & tools can provide a foundation for better and simplified air situation design as the design is transparent where air situation algorithm can be simplified with this methodology and it can operate in an integrated manner over a broad spectrum of information processing. The database resources and objects can be shared so that they will interoperate with other integrated system in a collaborative manner for futuristic design. Traditional platform centric war can be realised and transformed to a collaborative network centric war fighting based on information fusion and the methodology proposed. 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 traditional basic techniques and internal data structures). The results of the experiments provide evidence that a non real time database can also be optimised and designed to work under intense scenarios of the air situation process. A high-performance embedded DBMS's (real-time in memory databases) which have much higher data access and quick analysis capability with more powerful processors, a real-time operating system can perform much better with advanced SQL capabilities although its implementation and benefit is yet to be understood. However the embedded non real time DBMS could be expected to perform better and support overall integrated system till its deadline agrees up to deterministic criterion.

References

1. Shabbir. Database vs files. *In* Web development. 29 April 2005. http://www.go4expert.com/articles/database-vs-files-t283 (Accessed on 13th March 2016.)
2. White paper on Oracle database online documentation (Memory architecture) 10g Release 2 (10.2), 2005, pp. 1-36. https://docs.oracle.com/cd/B19306_01/server.102/b14220/memory.htm (Accessed on 13th March 2016)
3. White paper Oracle database online documentation on automatic storage management. Oracle 10.2, 2005, pp. 1-14. http://www.oracle.com/technetwork/database/asm-1.pdf (Accessed on 13th March 2016)
4. Mark, Pullen J. & Nicklas, Lisa. Maturing supporting software for C2-simulation interoperation. *In* the 15th IEEE/ACM International Symposium, C4I Center George Mason University Fairfax, VA, USA, 2011. pp 148-154. doi : 10.1109/DS-RT.2011-20
5. Siddiqui, Zeeshan & Alghamdi, Abdullah S. SOA based C4I common-view interoperability model. C4I Centre of Advance Systems (C4ICAS), King Saud University, Riyadh, Kingdom of Saudi Arabia. *Sci. Int.* (Lahore), 2014, **26**(1), 175-180.
6. Kilimci, P. & Kalipsiz, O. Moving objects databases in space applications. *In* the 3rd International IEEE Conference on Recent Advances in Space Technology, 2007. pp. 106-108 doi: 10.1109/rast.2007.4283959
7. Xu, Junming; Hou, Xuelong; Lu, Cuihua; Luo, Musheng & Zhou, Ranhui. The key technology research on certain missile weapon system simulation based on HLA/RTI. *In* the 5th IEEE International Conference on Software Engineering and Service Science, 2014, pp. 617-620. doi: 10.1109/icsess.2014.6933644.
8. McDaniel, D. & Schaefer, G. Real-time DBMS for Data Fusion. *In* Proceedings of the 6th IEEE International Conference. 2003, **2**, pp.1334-1341. doi: 10.1109/icif.2003.177392
9. Ceruti, Marion G. Report on data management challenges and development for military information systems DOD. Govt. of US. *IEEE Trans. Knowledge Data Eng.*, 2003, **15**(5), 1059-1068. doi: 10.1109/TKDE.2003.1232263
10. Chang, C.B. Optimal state estimation of ballistic trajectories with angle-only measurements. *In* the 18th IEEE Conference on Ballistic Missile. Defense Office Department of the Army. 1979. Technical note 1979-1, pp. 632 - 637.

doi: 10.1109/CDC.1979.270261

11. Singh, Sarbjit. Database model for military operational information system. Army Hqr, New Delhi, *Def. Sci. J.*, 1988, **38**(2), 141-159.
doi: 10.14429/dsj.38.4832
12. Oracle Database Online Documentation. 10g Release 2 (10.2), Process architecture, page 14 of 36. https://docs.oracle.com/cd/B19306_01/server.102/b14220/process.htm. (Accessed on 13 March 2016)

ACKNOWLEDGEMENT

The author would like to express his sincere appreciation for embedded programming support by colleagues of DCCT/C^{4}I, of DRDL. This application would not have completed without their support, discussion. The author also sincerely thanks to Asst. Professor Mr Pandu Ranga Rao, Department of Computer Science of IIT, Hyderabad for the advice and support rendered for preparing this paper at later stage.

CONTRIBUTOR

Mr Aniruddha Basu had completed his BE (Electrical Engineering) from Institution of Engineers, India. He is working as Technical Officer in Directorate of Computer Communication Technology, Defence Research and Development Laboratory, Hyderabad in C^{4}I group. His current area of interest includes : Database application in C^{4}I real time application development in RDBMS Oracle database and real time databases.