

FLIGHT OPTIMISATION USING IN-MEMORY HASHING

Pathirage Samya Dimithrie

(118072T)



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

December 2013

FLIGHT OPTIMISATION USING IN-MEMORY HASHING

Pathirage Samya Dimithrie

(118072T)



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

*This dissertation submitted in partial fulfilment of the requirements for the Degree of
Master of Science*

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

December 2013

Declaration

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date:

The above candidate has carried out research for the Masters thesis under my supervision.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

.....
Signature of the supervisor:

.....
Date

Abstract

Current travel booking systems take an appreciable amount of time (often over 30 seconds) to process even a relatively straightforward query. Producing results faster will give a competitive advantage in the travel market. Traditionally, data is placed in storage and then accessed and acted upon in the computer's memory which is a reason for the reduction in speed. The emergence of multi-core processors and availability of large amounts of main memory at low cost allows even large data sets such as airline flights and fares to be stored in-memory and processed using many-core processors.

In this work we use in-memory technology to develop a fast hash-based algorithm to search for flights and fares in air travel planning.

The objective of this project is to select a reasonable-sized set of priced itinerary solutions from the huge number of all possible solutions. Generated priced solutions must satisfy a set of constraints (such as maximum number of connecting points should be equal or less than three, departure station, arrival station, departure date, one way or round trip, maximum price, maximum duration, etc.). A graphical user interface can be created to manipulate this set. Availability checking and flight booking were considered beyond the scope of this project.

We show that air travel planning can be implemented using in-memory hashed lookups, eliminating the need for database lookups, searches or graph operations.

With the use of in-memory technology, hashing and lookups, priced solutions for a query are generated on average within 40-50 milliseconds. Memory is used efficiently and the total memory consumption for the full data set is approximately 1GB which is easily available on a standard PC.

This algorithm can also be extended to take advantage of many-core processors such as GPUs.

Keywords: flight search, hashing, in-memory

Acknowledgements

Foremost, I would like to express my sincere gratitude to my academic supervisor Professor Gihan Dias for providing guidance, ideas, support, and encouragement as well as his great effort to explain things clearly and simply throughout the duration of my research.

My sincere thanks also go to CODEGEN (Pvt) Ltd for providing me financial support, guidance and required data, during the period of research. And also especially Dr.U.Kohomban, my industrial supervisor, for his valuable ideas and support, and to Lasantha Wimalasena for all the support, and for all the other members who helped me in numerous ways.

I take this opportunity to thank the Department of Computer Science and Engineering of University of Moratuwa for giving me the opportunity to carry out this research and providing resources.

I would like to thank all the members of 'Multicore and Manycore' research group of University of Moratuwa, for providing me valuable ideas and guidance.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

I wish to thank K.H.M.K.T. Jayawardene for continuous help, Anushke Guneratne for reviewing my thesis and all the other friends for their support and encouragement throughout.

Last but not least I would like to thank my parents who helped me to stay focused when I most needed it.

TABLE OF CONTENTS

Declaration	ii
Abstract	iii
Acknowledgements	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF TABLES	viii
1 INTRODUCTION	1
1.1 Problem Definition	1
1.2 Objectives	3
1.3 Methodology	3
2 BACKGROUND	4
2.1 Conventional Travel Planning Systems (TPS)	4
2.2 Airline Reservation System	5
2.3 Hubs and Spokes	6
2.4 Itinerary Types[1][20][26].....	7
2.5 Complexity of Air Travel Planning	8
2.6 Applicability of Graph Traversal	9
2.7 In-Memory Data Management	9
2.8 Hashing.....	12
2.9 Hash Functions	13
2.10 Sparse Data	16
2.11 Google Sparse Hash.....	17
2.12 sparse_hash_map	17
2.13 MurmurHash.....	17
2.14 XML.....	18
2.15 XML Query.....	19
2.16 Xpath.....	19
2.17 Xpath with Large XML Files.....	19
3 METHODOLOGY AND ALGORITHM DESIGN	20
3.1 Data Preparation	20
3.2 Store Data In Memory	21
3.2.1 Data structure selection	21
3.3 Restricting and Filtering	30

3.4	Lookups	31
3.4.1	Route and flight schedule search.....	31
3.4.2	Apply the rules and identify fares	32
3.5	User Interface of the System	35
4	RESULTS AND DISCUSSION	36
4.1	Criteria for Performance Evaluation	36
4.2	Hash Function Selection.....	38
4.3	Response Time	38
4.4	Resource Utilization	40
4.5	Comparison with A* Algorithm.....	41
4.5.1	Implementations of the programmes.....	41
4.5.2	Response time comparison for random queries	42
4.5.3	Response time comparison per flight.....	43
4.5.4	Overall performance comparison with A* algorithm	43
4.6	Implemented using background worker thread	47
4.7	Connecting points expansion as a pre-processing step.....	47
5	CONCLUSION AND FUTURE IMPROVEMENTS	49
5.1	Conclusion.....	49
5.2	Future Improvements	50
References	53



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF FIGURES

	Page
Figure 2-1: Airline booking process [29].....	5
Figure 2-2: Atlanta hub	6
Figure 2-3: Storage Vs Price over the years [21].....	10
Figure 2-4: In-memory creates value through speed [21].....	11
Figure 2-5: Hashing	14
Figure 2-6: Sparse data.....	16
Figure 3-1: Overview	20
Figure 3-2: Routes and rules	22
Figure 3-3: Arrange routeid s	23
Figure 3-4: Store fares.....	23
Figure 3-5: Fare contract.....	27
Figure 3-6: Route data.....	27
Figure 3-7: Fare break-down.....	29
Figure 3-8: Data loaded in to memory.....	30
Figure 3-9: Connecting airport expansion	31
Figure 3-10: System overview.....	34
Figure 3-11: User Interface	35
Figure 4-2: Response time of random queries	38
Figure 4-1: Comparison between MurMurHash3 and std hash	38
Figure 4-3: Data loaded in to memory	40
Figure 4-4: Memory and resource overview of data loading.....	40
Figure 4-5: Response time comparison between A* and hash-based implementations	42
Figure 4-6: Response time per flight -comparison A* and hash-based implementations	43
Figure 4-7: Hashed connecting points Vs runtime connecting points generation	48
Figure 5-1: Parallel implementation fare rules lookup and flight routes, schedules lookup.....	50
5-2: Parallelizing according to connecting points.....	51

LIST OF TABLES

	Page
Table 1-1: Access and Read Times for Disk and Main Memory.....	2
Table 2-1: Itinerary types	7
Table 2-2: Comparison of hash functions [17]	15
Table 3-2: Arrange and store flight leg schedules data in a sparse hash map.....	25
Table 3-1: Store flight leg schedules in an array of flight leg structures	25
Table 3-3: Store connecting airports between two airports.	26
Table 3-4: Store route data in a sparse hash map.....	28
Table 3-5: Store rules under the applicable routes, in a sparse hash map.....	28
Table 3-6: For every route for every applicable rule store adult fare	29
Table 4-1: Performance measurements, hash-based implementation	44
Table 4-2: Performance measurements, A*-based implementation	45
Table 4-3: Comparison with A* Algorithm.....	46
Table 4-4: Store all the extended connecting points in memory.....	47



LIST OF ABBREVIATIONS

Abbreviation	Description
ARS	Airline Reservations System
CRS	Computer Reservations System
SSIM	Standard Schedules Information Manual
TPS	Travel Planning Systems



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk