УДК 004.624:004.087.4

ЗАПРОС ДАННЫХ С ПЛАТЫ FPGA (ПЛИС)
QUERYDATAWITH ANFPGA BOARD

PhD Candidate: Haider I Mohsin, Prof. J.I.Batyrkanov
Kyrgyz state Technical University
liveoffice12@gmail.com

In thispaperacommercialFPGA coprocessorboard isused toacceleratethe dataofqueries onarelational databasethatcontainstextsandimages. FPGAdesigns fortextsearchingandimage matchingaredescribed and their performancessummarized. Apotential design foradatabaseJOINoperatoristhen studied. Aqueryoptimizationpreprocessor isthen proposed.

Keywords:Field ProgrammableGateArray (FPGA), ConfigurableComputing,Textand ImageDatabase, QueryOptimization

1. Introduction

Computerswith FPGAcoprocessorboards havebeen usedto acceleratemanydifferent applicationsin thepast,especiallythose that do notinvolvemassiveamountoffloating pointcomputations [1,2, 3, 4]. Query dataon textand imagedatabases providesampleopportunities foracceleration withsuchcoprocessorboards. In thispaperacommercialF-PGAcoprocessorboard isused toacceleratethe dataofqueries onarelationaldatabasethatcontainstextsandimages. In [5], ASICchipsweredevelopedforimage and stringmatchingdataand ahybridand reconfigurableboard architecturewas proposedthatcombinedthoseASICchipsand FPGAs/EPLDsformultimedia applications. Thediffi-

cultyof that approachisthat the functionalityofthe ASICchipsneedsto beversatileenough to cover theapplicationdomain.

In Section 2, FPGAdesignsfortext searchingand imagematchingaredescribed and theirperformancesusingaXilinx XC4085on acommercialFPGAboard are summarized. Thedesignsareusedto acceleratequeriesonan Oracle8idatabase. Apotentialdesign foradatabaseJOIN operatoris then studied. In Section 3,a queryoptimization preprocessoris proposed. Section 4concludesthepaper.

2. TextSearching andImage Matching

In order to evaluatetheproposed approach, a testplatformwasused which wasa600

MHzPentiumIIIpersonalcomputerwith an AnnapolisMicro Systems'sWildForceTM FPGAboard. Onthatcomputera small text/imagedatabasewassetup andaGUI interface-wasdevelopedin Microsoft Foundation Classto displayimage/textdata records. Queriestoaccessthedatabasewere implemented based on ODBC(Open Data BaseConnectivity)APIfunctions.Three differentoperationswerestudied,text searching, imagematching, andJOIN operator.Theyaredescribedinthe followingsubsections.

2.1 Text Searching

The databasecontainsrecordsthatinclude some photosandresumes.Akeywordstring canbe input throughtheGUI andusedto matchagainsteachtext file.All the records whoseemployeeresumefiles include the keyword weredisplayed.

An efficientkeyword-matching algorithm calledtheKMPalgorithm (inventedby Knuth,Morris,andPratt)wasimplemented on-thePC.Thealgorithm doesnotusemore than M+Ncharactercomparisonstomatcha keywordofMcharactersagainstastringof Ncharacters.

Two FPGAdesignsfor thecomparison ofa keyword against a text file were implemented, one a straightforward hardware implementationoftheKMP algorithmandtheotherabruteforceusage of parallel-comparators to implement the naïvetextsearchalgorithm (of O(M*N) complexity sequentially). It-wasfoundthat whiletheKMPalgorithmismoreefficient on sequentialmachines, itenjoys no advantage in FPGAimplementation. In fact, itsuffersfromacouple-ofdrawbacks:(1)Its control circuit ismorecomplicatedand(2) it ismoredifficult fortheFPGAtograb externaldata every clock cyclebecause whetherit-needsnew dataornotdependson thecomputationon-thecurrentdata.

Asa result,withouthaving acomplicateddata buffer, the KMPFPGA implementationcan onlygrab one newdata once every three clockcycles. Bycon-trast,thebruteforce

parallel comparatorimplementationsimply grabsnewcharactereveryclockcycle. In

addition because thememory portcan providefourconsecutivecharactersinone

clock cycle, four copies of the same hardwarecanbe used totakeadvantageof that data par-allelism with the brute force implementation. Be-causeofthesereasons the parallel comparator imple-mentation is considered superior totheKMP one.
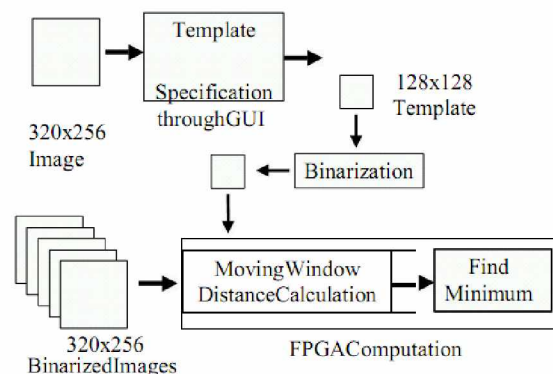
Inorderto test "caseinsensitive"keyword search,all thetexts in eachtextfilewere convertedto lowercases rightbeforethe keyword comparison. It-wasfoundthaton thePC foraspecificcase-with22Kbytesof text, it took 1.5 seconds for "case conversion"and0.01 seconds for"keyword matching." AnFPGAdesignthat performs both thecaseconver-sionandthekeyword matching wasimplemented.

Thedesign, utilizing 1,035 configurable logicblocks (CLBs), runs at10MHz andprocessesfour characters perclock cycle.Since theFPGA design spends less than 0.01 seconds for bothcomputa-tions,thespeedupismorethan

150. Note that themajorityofthe computa-tion is on the case conversion insteadofthekey-word matching.

2.2 ImageMatching

With the GUI, a userspecifies a128x 128 ar-eaoutofa320x 240imageandusesthat area as atemplate tocomparetoallthe images inthedatabase. The record-whose image *bestmatchs* with the template (within an"acceptable" degreeofmatching)is displayed. To simplifyandspeed up the matching process, thegray-scaledimages were preprocessed and their pixels converted into binary values. The binary images are stored in the database together with thegray scaled ones. The templateis also binarized using a thresh-old chosen basedonthehistogram ofthetemplate. The resulting system is as shown in the following figure.
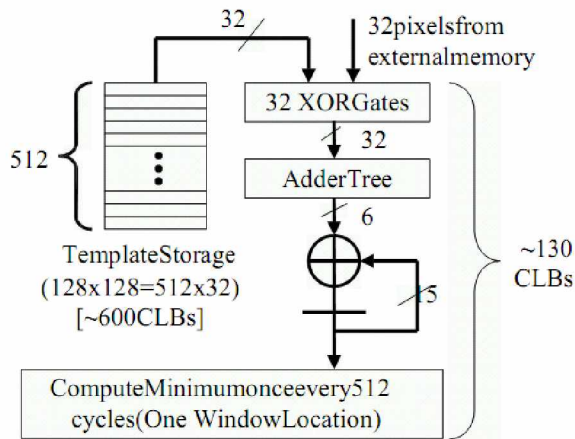
320x256 Image → Template Specification throughGUI → 128x128 Template → Binarization → MovingWindow DistanceCalculation → Find Minimum

320x256 BinarizedImages

FPGAComputation

In termsof thematchingcomputation, itisa movingwindowbased distancecalculation. When-thetemplateisapplied to apixel location ofanimage, thedistanceis computed asthenumberofmismatches between thetemplateand theimage

window. Becausethe templateand the im-agearebothbinarized, atemplatepixel and animagepix-elareconsidered mismatchedifanXOR operation pro-duces a resultofone.Thecomputation therefore lendsitselftoaveryefficientPentium implementation becausean integerXOR instruction handles32pixelsin parallel. A

256-entrylookup tableisused onthe Penti-umcode tocount thenumberofones in a32-bit inte-gerresultingfromtheXOR operation.With thiseffi-cientPentiumcode, thesearch onthePCtookaround 8.48 seconds fornineimages. Thesame search us-ingasinglecomputationalunitona XC4085 FPGAchip(with 3,136 CLBs)with aclockrateof40 MHzon aWildForce

boardtookaround 3.91 seconds.Therefore thespeedup isaround 2.17.TheFPGA design is-asshownin thefollowingdiagram.

TemplateStorage
(128x128=512x32)
[~600CLBs]

ComputeMinimumonceevery512
cycles(One WindowLocation)

Byusing multiplecomputation units, the potentialimprovementsofafuturedesign on thecurrentWildForceBoardareestimated

as follows. On asingleFPGAchip,around

4 computationalunitscanbebuiltand the speedup willbecloseto 5orabove. With

fourFPGAchipsperboard,thespeedup is expectedto bearound 10 orabove. Here the

speedup isnotassumed to belinearbecause otheroverheadsmaydegrade the

performancewhenhigherdegreeof parallelismisexplored. Anotherpoint worth mentioningis that Xilinx Virtex

FPGAchipswillbemore suitableforthis design thanthe4085 chipsbeingused

becausethetemplatecould then bestored moreefficiently(both inspaceand inspeed) in on-chip blockRAMs.

2.3JOIN Operator

Typicaldatabaseoperationsinvolve"JOIN" operations. AJOINoperation ontwo tables producesatablewhosekey valuesarein bothtables.Theoperation isbasicallyfor each elementin asequenceA,tofinda correspondingmatchingelement inanother sequenceB. In order todevelopan efficient FPGAdesign for theJOINoperation,the followingsequentialalgorithmwas developed.

Supposetherearetwo sequences of integers, A andB. ThereareNelements in A, and M elements in B. AssumeN

<<M. Thealgorithm to join thetwo sequences includes two steps: (1)Sort thesequenceA : complexityO(N logN) and (2)ForeachBelement, find the matching element in A by using abinary search :complexityO(M log N). Thereforethetotal complexityis O(M logN)that comes from theM binary searches. Thealgorithm was implemented and tested on an Oracle database. WhileapureOracleJOIN operation includingdatafetchingtook27 secondsonaparticulartestcase, the proposed sequentialJOINoperation with datafetched through the sameOracle8i databasetookonly22 seconds.Thisshows

thatitispossibleto build ourownJOIN operation thatisasefficientasoreven better thanapureOracleJOINoperation (atleast forthelimited testsweran).

An FPGAdesign toacceleratetheJOIN operation, ormore speciallytoacceleratethe binarysearch hasbeen proposed.Thedesign usesapipelineofcomparatorswhereeach comparisonresultdetermines theoperand to be sentto thenextcomparatorstage. With

a40 MHzdesign, itisestimatedthat

roughly40 millionintegerscan beprocessed eachsecond (assumingeachintegeris

compared to1,024sorted values). Sincea programrunningon a600 MHzPentiumIII

PCspendsaround 30secondson similar operations,theFPGAdesign can potentially

leadto an orderofmagnitudeimprovement fortheJOINoperation.

3. Query OptimizingPredata

Onceadatabaseiscreatedandinuse,the main performance concern is about the data,optimization, and execution of queries. A querycan typically be representedasa"querytree,"whereeach leafnoderepresentsarelation(i.e.,table) andeachinternalnoderepresents adatabase operation. Forexample,thequerytreeas shown inthefollowingfigurecorrespondsto an optimizedrepresentationofthequery **Q** (Find the lastnamesofemployeesbornafter

1970 who workon atasknamed "Champ").
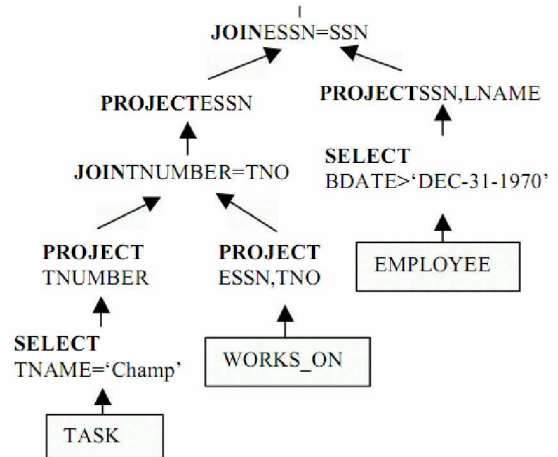
With optimizationtheinitialquerytree

would use theCartesian productofthree tables, EMPLOYEE, WORKS_ON, and

TASK, which maylead toahuge table. Thereforeitisbetterto applya setof rules

to transformthe initialquerytree toan optimized treethatismoreefficientto execute.Theresultisshown in theprevious

figure, wherebasicdatabaseoperations such as**SELECT**and**PROJECT**areappliedto

individualtablessoto reducetablesizes beforetheJOINoperation isappliedto joiningmultipletables.Theprocessof

convertingan initialquerytreeto an optimized querytreeiscalled query

optimization. Sinceeachbasicdatabase operation maybe implemented in many different ways, theoptimization also

involveschoosingthemostefficient implementation foreach operation. For

example, thereareat least six differentways to implementa**SELECT**operation,

includingabruteforcelinearsearch, a binarysearch, usingaprimary keyorhash key, and so on, fourdifferentways to implementa**JOIN**operation,andtwo different ways to implementa**PROJECT** operation. Queryoptimization ina commercialdatabaseisnormallydoneby thedatabaseengineusingbuilt-in index structuresonthehostcomputer.

InordertouseFPGAcomputingresources todrastically speed upqueryexecution,we propose todevelop a"query preprocessor" thatoptimizesqueriesby taking into account theavailability ofFPGAresources. The inputtothequerypreprocessorisaquery application in"embedded SQL," which is basically-aprogram intheClanguagewith SQL statementsembedded init.(ThoseSQL statementsare enclosedbypre-compilation directives.) The"querypreprocessor"will separateembeddedSQLstatementsfrom the restoftheapplicationandconvertthem into twoparts, one for the FPGAresourcesand theotherfor theback-enddatabaseengine. The rationale is, because of the finite amountofFPGA resources, itmaynotbe possibleto squeezeallthequeryoperations into the FPGA coprocessor board. In addition, in the case when there is not enough FPGA resources, the query preprocessorwillneedtoevaluatewhether toreconfiguretheFPGAformoredatabase

operations or simply allocate those operations tothehostmachine. Thisalso implies that optimization opportunities

available through index are not wasted. More specifically, the steps involved in

using thequery preprocessorwillbeas follows.

1. The preprocessor isolates SQL statementsfrom anembeddedSQL application.

2. ThepreprocessorconvertsaSQLquery toaquerytreeandtransformsittoan

optimized tree.

3. Thepreprocessoridentifies"sub-trees"

thatcan beacceleratedwithFPGA computing. Each sub-treewillbe replacedwithCcodes thatcontrolsthe FPGA coprocessorboard-whilethe remaining partofthetreewillstillbe codedin SQL.

4. After being compiled, the final application codecan beexecuted. A commercial database backendwillbe invoked fortheSQLpart andanFPGA board forthesub-treeC code. There will beparalleloperationsforthesub-treesallocatedtotheFPGAboard.

Tree Partitioning

The query treeispartitionedintotwoparts, one-fortheFPGAand the other foraback- enddatabaseengineon thehostmachine,as shown in the following figure.
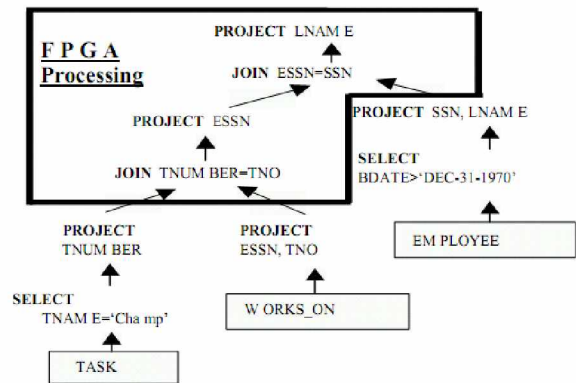
Inthisexample, instead ofsendingthethree tables to theFPGAboarddirectly,wefirst apply **SELECT**and**PROJECT**operations onthose three-tablesandthensendtothe FPGAboard the resulting tableswhichare supposed to be much smaller. Forexample, after the table TASKgoesthroughthe **SELECT** TNAME='Champ'and **PROJECT** TNUMBER operations, the resulting tablehasonly onesinglecolumn, i.e., TNUMBER,andcontainsonlythose rowswhose TNAMEis'Champ'.Asshown inthefiguretheFPGAboardisresponsible for two **JOIN** operations and two **PROJECT**operations.

FPGA Computing

Dependingonthesizeofthethreetables sent to theFPGAboard,differentFPGA designscould beused.Computation ofthose **JOIN**and **PROJECT**operations canbe pipelined to increasespeedup.

Forthe**JOIN** TNUMBER=TNO operation,supposethere are only ten TNUMBER rows. In thatcase,ten comparatorscanbe usedforeveryincoming TNO to seeifthereisamatch.However, since the number of TNUMBER rows cannotbepre-determined, amore complicateddesignthat includes comparators, FIFOs, and acontrollerwillbe moreappropriate. Thespeedupwilldepend onhowmany comparatorscanbesqueezed intothehardware.

For the **PROJECT** ESSN operation, it is necessary to remove duplicateESSNrowsandkeeponlydistinct ones. This can be done either with brute forcecomparisonsor-

withhashing. Ineither case, parallelcomparisoncanagainbe applied. (Computationofthehashfunction canbepipelinedwiththecomparison,inthe case of using hashing.) The query preprocessorwillneedtomakethedecision
and choose the most efficient way of implementation. Notethatthethreetables inthedatabasearelockedduringthewhole transactionprocesssoas toavoid any modificationto thetablesduringthe transaction byotherapplications.
Example 2

Thisexamplecorrespondsto anotherquery **Q2**(i.e., Findthelastnamesofemployees wearing glasseswhowork onatasknamed "Champ" andwhose resumecontains the keyword "FPGA"). In thiscasetext/image databasesareinvolved.
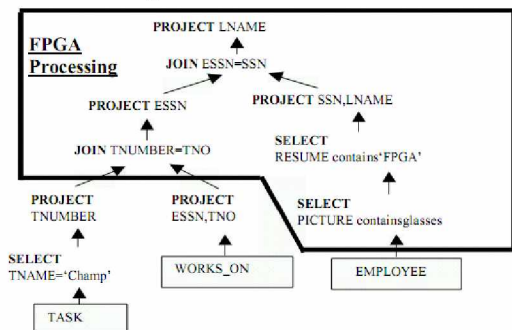
Asshownintheprevious figure,theimage data ofthepicturesandthesearching of the text documents should be handled with theFPGAresources Asamatterof fact, if there are not enough FPGA resources, the query preprocessor may decidethat itisbettertoimplementONLY the text/imagedatapartontheFPGA since those parts provide the maximal amountofparallelism.Sucha partitionis shownin the followingfigure.Apparently therearemany designoptionsother thanthe two previousdesigns. Itis the responsibility ofthequery preprocessortorankthose different design options and choose the "best"one for implementation.
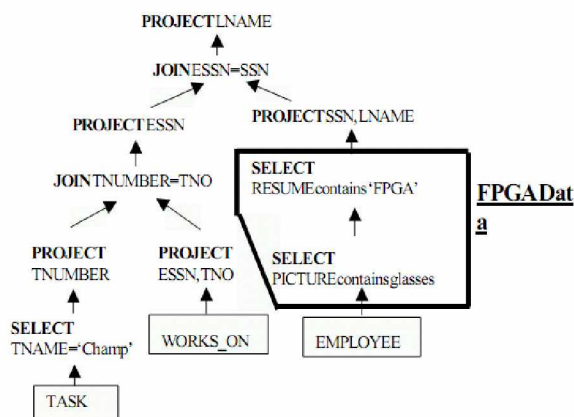
QueryPreprocessorInternal

The figureas shown at the end ofthe paperschematicallyillustrates thetwo majortools that need to bedeveloped: "QueryPreprocessor"and the "FPGA Design Tool". Given aC/C++ application program (i.e., C/C++ file) that contains ODBCstatements as an input, ausermayidentify someparts that havenothingto do with database operations but can be accelerated with FPGA boards. Theusercan then use somedomain-specificFPGA design tools, whethertheyare forimage data, video data, ortext stringdata, to producesome FPGA designs and insert the correspondinghost codeinterfacein the C/C++ file. In addition, theusermay add somepreprocessordirectives to help specifytheparts in theC/C++ filethat can beput into aquerytree. The
resultingC/C++ file,FPGA design files, and FPGA libraryofbasicdatabase operators act as inputs to the "Query Preprocessor" and the"FPGA Design Tool".

4. Conclusions

Multimediadatabases havemany applications that need to beaccelerated. In this papersomesuch potential applications areaccelerated byusinga commercialFPGAcoprocessorboard based on XC4085. FPGAdesigns fortext searchingand imagematchingaredescribed and theirperformancessummarized. A potentialdesign foradatabaseJOIN

REFERENCES

[1]J.Villasenor,B.Schoner, K.Chia,C. Zapata, H.Kim, C. Jones, S.Lansing, and B. Mangione-Smith, "Configurable Computing Solutions for Automatic Target Recognition," in IEEE Symposium on FPGA Custom Computing Machines, pp.
70--79, 1996.

[2] J.S.N.Jean,X.Liang,B.Drozd,K. Tomko, and Y. Wang,"AutomaticTarget Recognition with Dynamic Reconfiguration,"to appear inJournalof VLSISignalData.

[3]M. Alderight, E.L. Gummati, V. Piuri, and G.R. Sechi, "A FPGA-based Implementation ofaFault-Tolerant

Neural Architecture forPhoton Identification,"in Proc. ofACM/SIGDA International Symposium on FPGAs, pp.
166-172, 1997.

[4] M.ShandandL.Moll, "Hardware/Software Integration inSolar Polarimetry,"inIEEESymposiumonFPGA Custom Computing Machines, pp. 18-26,

2011.

[5]H. Blume, H.M. Bluthgen, C. Henning, and P. Osterloh,"IntegrationofHigh- PerformanceASICs into Reconfigurable SystemsProvidingAdditional Multi-media Functionality,"IEEEInternational Conference-on Application-Specific Systems, Architectures, and Processors (ASSP),July2010.