FPGA Acceleration of Information Management Services

29 Sep 2004

Richard Linderman
Mark Linderman
AFRL Information Directorate
C. S. Lin
Univ. of Missouri-Columbia
Information Management supporting Horizontal Fusion within the Battlespace

- INTEROPERABILITY of C4ISR systems
- Establish a "Joint Battlespace Infosphere"
- Achieve Persistent Battlespace Awareness
- Support Dynamic Planning and Execution

Step towards web-based distributed C4ISR "intelligence"
What do people want?

Right Information to the Right People (and machines) at the Right Time

Very Popular Information Objects (=> many subscribers):
1. Moving objects (airborne, ground, etc) with a “region of interest”
2. Imagery (EO, SAR radar, Hyperspectral)
3. Other “detections”—cyber, chem-bio, signals
Key Information Drivers

• Vision: A globally interoperable information “space” that integrates, aggregates, filters and disseminates tailored battlespace information.

• Open standards-based information management core services of Publish, Subscribe, Query & Control to improve extensibility & affordability of future AF C4ISR systems.

HPC can help the infosphere scale to 100x current proportions and beyond
Pub-Sub Brokering Problem

• Information regarding a publication is described using an XML metadata document.

• What the subscribers want are defined using XPath predicates.

• The pub-sub brokering system evaluates predicates against the XML document to find matches.
<metadata>
  <baseObject>
    <InfoObjectType>
      <Name>mil.af.rl.mti.report</Name>
      <MajorVersion>1</MajorVersion>
      <MinorVersion>0</MinorVersion>
    </InfoObjectType>
    <PayloadFormat>text/plain</PayloadFormat>
    <TemporalExtent>
      <Instantaneous>2003-08-10T14:20:00</Instantaneous>
    </TemporalExtent>
    <PublicationTime/>
    <InfoObjectID/>
    <PublisherID/>
    <PlatformID/>
  </baseObject>
  <IntelReportObject>
    <OriginatorID>VMAQ1</OriginatorID>
    <DetectionDateTime>20030728T163105Z</DetectionDateTime>
    <Latitude>42.538888888888884</Latitude>
    <Longitude>19.0</Longitude>
    <MTIObject>
      <TrackID>000001</TrackID>
    </MTIObject>
  </IntelReportObject>
</metadata>
Examples of Predicates

- \(((/\text{metadata/IntelReportObject/Latitude}>60) \\
  \text{or} (/\text{metadata/IntelReportObject/Longitude}<60)) \\
  \text{and} (/\text{metadata/IntelReportObject/OriginatorID} = '\text{bravo}')\) \\

- \(((/\text{metadata/IntelReportObject/MTIObject/TrackID}>17) \\
  \text{and} (/\text{metadata/IntelReportObject/OriginatorID} !='\text{alpha}') \\
  \text{and} (/\text{metadata/IntelReportObject/Latitude}>45) \\
  \text{and} (/\text{metadata/IntelReportObject/Longitude}>45))\) \\

- \(((/\text{metadata/IntelReportObject/Latitude}<45) \\
  \text{and} (/\text{metadata/IntelReportObject/Longitude}>=45) \\
  \text{and} (/\text{metadata/IntelReportObject/OriginatorID} !='\text{delta}') \\
  \text{or} (/\text{metadata/IntelReportObject/Latitude}>=30) \\
  \text{and} (/\text{metadata/IntelReportObject/Longitude}<=90) \\
  \text{and} (/\text{metadata/IntelReportObject/OriginatorID} = '\text{alpha}')\))\)
1. The metadata of a publication is parsed into an organized data structure using software.

2. Retrieve the data needed for evaluating predicates.
• Use FPGA to implement a finite state machine to parse the metadata document. The XML document is read into the block RAM of the FPGA from a microprocessor through DMA.

• Predicates are evaluated in parallel using the data generated by the parser. (Combinational logic).
Heterogeneous HPC Hardware

- 48 Nodes in 2 cabinets
- Server product leverage
- Each node with: Dual 2.2 GHz+ Processors
  - 4 Gbyte SDRAM
  - Myrinet 320 MB/sec Interconnect
  - 80 GB disk
  - 12 M gate Adaptive Computing Board
- 34 TOPs demonstrated
- Online FEB 2003 supporting HIE, TTCP and SBR projects
The System

- **Microprocessor**
- **Input FIFO & Output FIFO**
- **64-bit bus**
- **XML Parser**
- **Predicate Evaluator**
- **FPGA board**
An Example for Illustration

**XML Document**

```
<A>
    <B> Great </B>
    <C>
        <D> Rome </D>
        <E/>
        <F> 106 </F>
    </C>
</A>
```

---

**Possible data query (XPATH)**

- `/A/B`
- `/A/C/D`
- `/A/C/E`
- `/A/C/F`

```
A
  B
    Great
  C
    D
    Rome
    E
    NULL
    F
    106
```
Comparing Numerical Fields

• A number in ASCII codes is converted to a binary integer

• To keep the precision up to one thousandth, a number is multiplied by 1000 with the integer part kept. (choice driven by precision used in NITF for longitude and latitude specification).

• Examples: 19.4 → 19400
  4.7729 → 4772
  -11 → -11000

• The 32-bit 2’s complement representation is used in the current design.
Table Generated by FPGA Parser

<table>
<thead>
<tr>
<th></th>
<th>pointer</th>
<th>length</th>
<th>hash_value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/A/B</td>
<td>7</td>
<td>5</td>
<td>xxx</td>
</tr>
<tr>
<td>/A/C/D</td>
<td>22</td>
<td>4</td>
<td>yyy</td>
</tr>
<tr>
<td>/A/C/E</td>
<td>34</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>/A/C/F</td>
<td>41</td>
<td>0</td>
<td>106000</td>
</tr>
</tbody>
</table>

Data will be sent back to the microprocessor and broadcast to the predicate evaluator backend logic.
Comparison of Hash Value/Number

Parser

Hash value or number from parser

Predicates Evaluator

Hash value or number from predicate

Comparators

( # leaves)  

( # clauses before optimization )
Current Results

• A design has been tested on a **single node (with an FPGA board)** on the AFRL/IF Heterogeneous HPC

• The parser, a finite state machine, processes nearly one character per clock cycle

• Predicate evaluator is a massively parallel pure combinational logic evaluated in one clock cycle

• For the first XML example (**700 ASCII characters including Tab, Line Feed, Space, etc.**) used in this presentation, with a clock rate of **50 MHz**, it took **45 microseconds** to complete. The time includes setting up the transfer, transferring the document and result, and parsing the document (about **14 microseconds** for processing on the FPGA).
Timing Data

For 700 character XML document with 14 leaves

<table>
<thead>
<tr>
<th># Predicates</th>
<th># of 64-bit words transferred</th>
<th>Processing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>104 (90 down plus 14 up)</td>
<td>45µs</td>
</tr>
<tr>
<td>11 (to 256)</td>
<td>104+4</td>
<td>48µs</td>
</tr>
<tr>
<td>1000 (to 1024)</td>
<td>104+16</td>
<td>52µs</td>
</tr>
</tbody>
</table>

• Theoretical processing time for this document on an FPGA board of the HHPC is 14µs at the clock rate of 50MHZ.

• Processing time is dominated by data transfer.

• Estimated processing time for 10,000 predicates is 114µs.

• Parse time alone is around 2 ms when implemented solely by software on a microprocessor.
Timing Impact Within JBI

Time for current version of JBI to broker a 700 character XML document with 14 leaves against 1024 predicates with 3% hit rate:

Xeon alone with compiled predicates: 8.5 seconds
Xeon with FPGA: 0.5 sec (17X of 33X max achieved so far)

Sample Predicate:

/metadata/baseObjectData/InfoObjectType/Name='alpha' or
/metadata/IntelReportObject/Latitude='VMAQ3' and
/metadata/IntelReportObject/OriginatorID='ab324e-f42a-4e23-324deac32' and
/metadata/baseObjectData/TemporalExtent/Instantaneous='0' or
/metadata/IntelReportObject/Longitude='VMAQ1' and
/metadata/baseObjectData/PlatformID='afrl'
# Hardware Usage
– with different numbers of predicates

<table>
<thead>
<tr>
<th># Predicates</th>
<th># Slices for Parser and Predicates Evaluator (out of 33792)</th>
<th># Slices for the complete system (including FIFO, etc.)</th>
<th>Synthesis time on 1GHz PC with 512MB RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>494 (1.46%)</td>
<td>1427 (4.22%)</td>
<td>65 sec</td>
</tr>
<tr>
<td>101</td>
<td>681 (2.01%)</td>
<td>1611 (4.77%)</td>
<td>82 sec</td>
</tr>
<tr>
<td>1000</td>
<td>983 (2.91%)</td>
<td>1875 (5.55%)</td>
<td>520 sec</td>
</tr>
<tr>
<td>2000</td>
<td>975 (2.89%)</td>
<td>1859 (5.50%)</td>
<td>1690 sec</td>
</tr>
</tbody>
</table>

* The average number of clauses per predicate is kept the same.

1. The hardware usage for the case with 1000 or 2000 predicates is only 2.9% plus a constant number of slices (about 930) for the FIFO block.

2. When the size of predicate set reaches a level, similarity among predicates becomes high and the optimization technique is capable of implementing them into the almost same size of hardware.

3. Layout generation takes about 8 to 10 minutes for the above cases (determined by the hardware size).

4. Synthesis time increases for larger sets of predicates (longer function simplification and optimization time is needed).
## Hardware Usage
- affected by the number of clauses

<table>
<thead>
<tr>
<th># Predicates</th>
<th>Clauses per Predicate (Average)</th>
<th># Slices for Parser &amp; Predicates Evaluator (out of 33792)</th>
<th>Synthesis Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>1.333</td>
<td>943</td>
<td>233 sec</td>
</tr>
<tr>
<td>600</td>
<td>1.969</td>
<td>970</td>
<td>243 sec</td>
</tr>
<tr>
<td>600</td>
<td>3.288</td>
<td>1021</td>
<td>376 sec</td>
</tr>
</tbody>
</table>

* A string in a clause is selected randomly from a set of 100 words, i.e., a leaf has one of the 100 different values.

- The number of clauses, not the number of predicates, affects the hardware size.
## Hardware Usage
- affected by the number of different leaf values

<table>
<thead>
<tr>
<th># Predicates</th>
<th># of Different Leaf-Values Used</th>
<th># Slices for Parser &amp; Predicate Evaluators (out of 33792)</th>
<th>Synthesis Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>10</td>
<td>670</td>
<td>210 sec</td>
</tr>
<tr>
<td>600</td>
<td>100</td>
<td>1021</td>
<td>376 sec</td>
</tr>
<tr>
<td>600</td>
<td>1000</td>
<td>1081</td>
<td>1005 sec</td>
</tr>
</tbody>
</table>

* A string in a clause is selected randomly from a set of 10, 100 or 1000 words.

- Using a larger set of words reduces the similarity among clauses and possible hardware sharing, and thus increases the hardware size.
Extensions: Two bit predicate representation

If Two-bit representation is used
11: Predicate is true
01: Predicate is false
X0: Result is unsure

For 1024 predicates FPGA usage is 9% vs 5.5% for single bit predicates.
Conclusions

• FPGAs working in concert with programmable processors within “heterogeneous” cluster nodes can improve XML parsing speed more than 40X

• Massively parallel evaluation of predicate logic is even more significant with thousands of predicates partially evaluated in a single clock cycle leading to overall brokering speedups
  – E.g. 17X for a 1024 predicate example with 3% hit ratio

• In general, the acceleration of “association” using FPGAs is a promising development as we explore architectures for cognitive information processing.