FPGA Acceleration of Information Management Services

29 Sep 2004

Richard Linderman

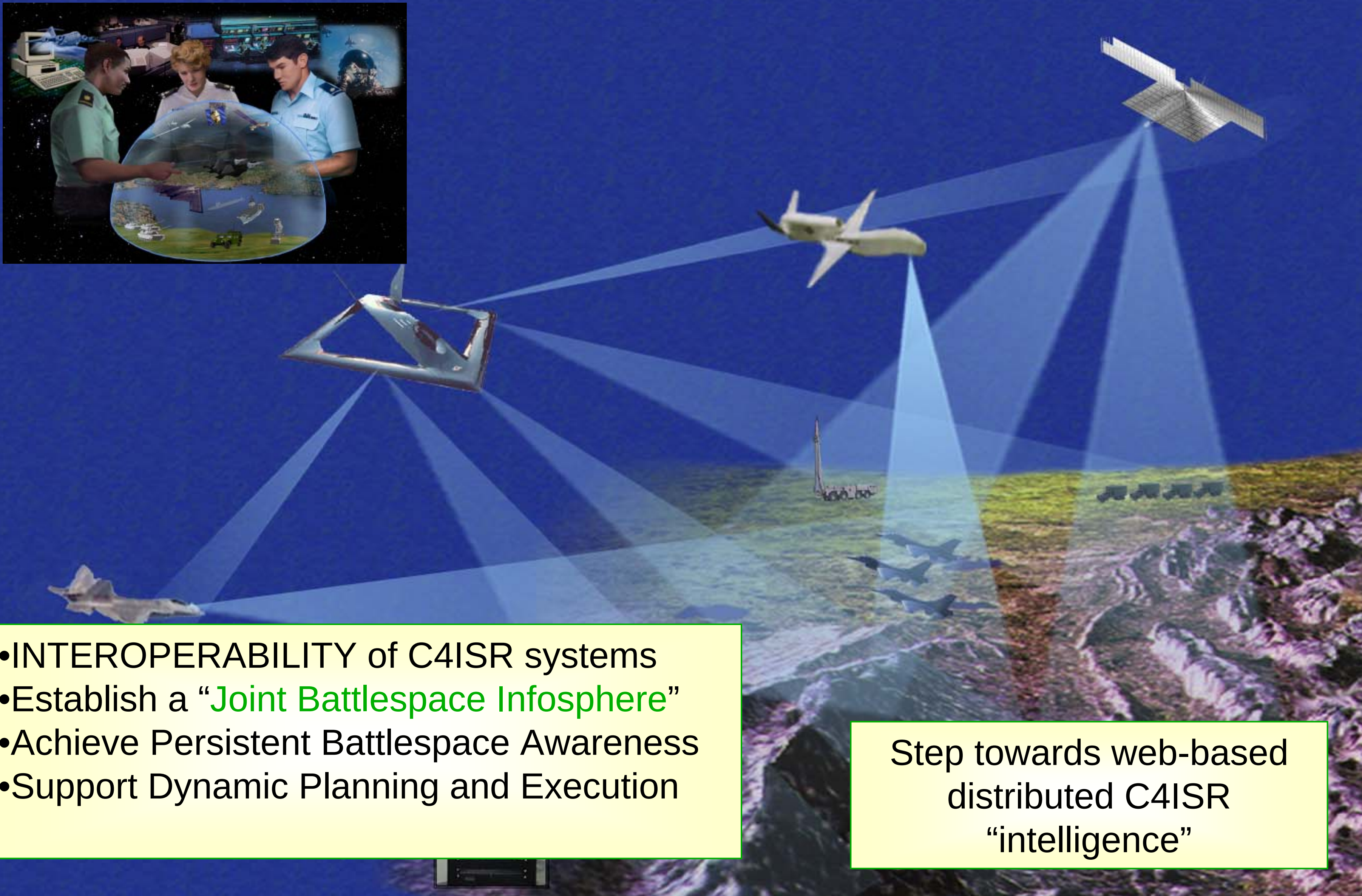
Mark Linderman

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

Joint Battlespace Infosphere

Delivering Decision-Quality Information

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 in XML: an example

```
•   <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

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

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

- (((/metadata/IntelReportObject/Latitude<45)
  and (/metadata/IntelReportObject/Longitude >=45)
  and (/metadata/IntelReportObject/OriginatorID!='delta'))
  or ((/metadata/IntelReportObject/Latitude >=30)
  and (/metadata/IntelReportObject/Longitude<=90)
  and (/metadata/IntelReportObject/OriginatorID ='alpha')))

Current Technique

1. The metadata of a publication is parsed into an organized data structure using software.

2. Retrieve the data needed for evaluating predicates.

FPGA for Acceleration

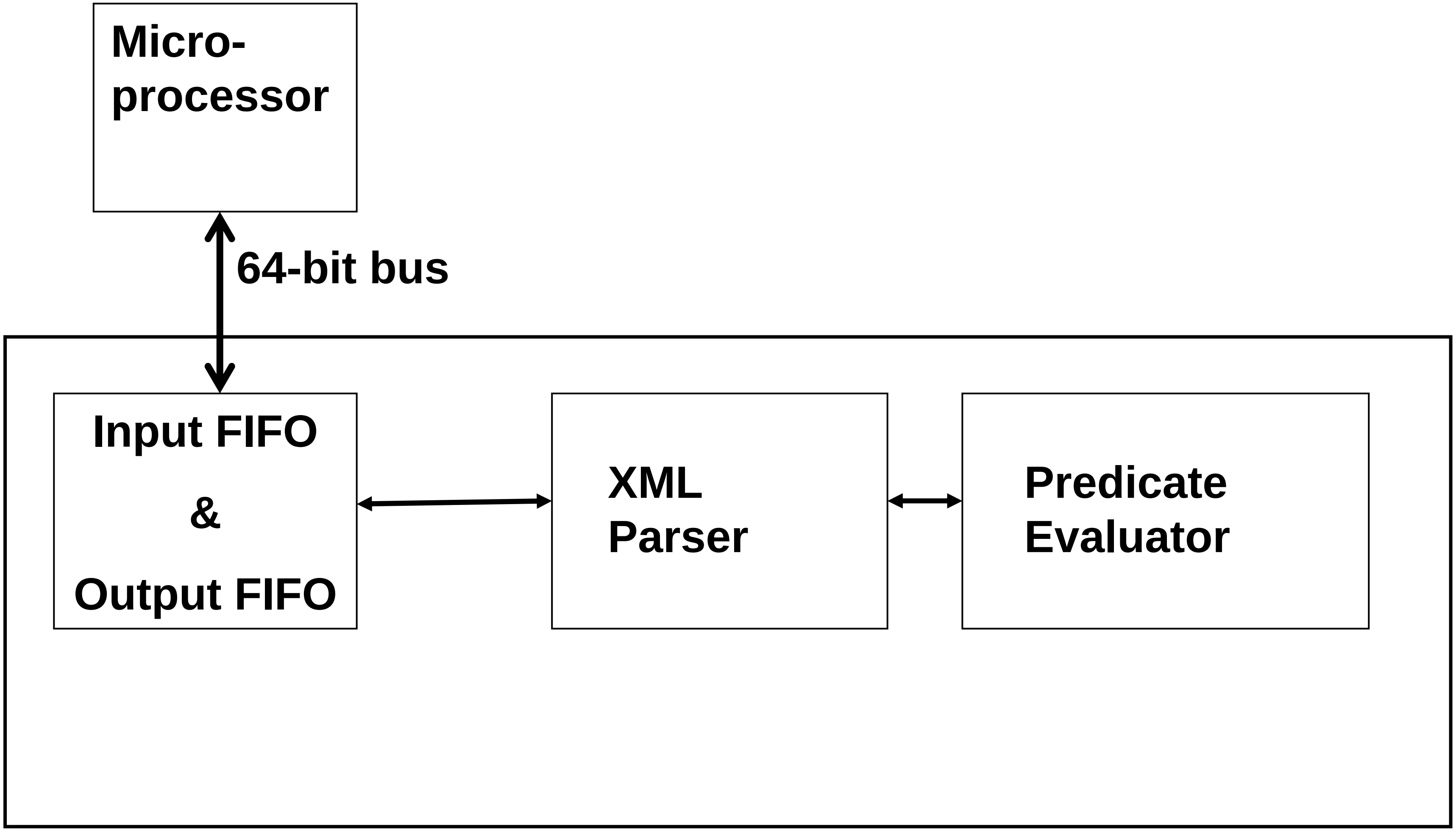
- **Use FPGA to implement a finite state machine to parse the metadata document. The XML document is read into the block RAM of the FPGA from a microprocessor through DMA .**
- **Predicates are evaluated in parallel using the data generated by the parser. (Combinational logic).**

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

Heterogeneous High Performance Computer

The System


Micro-processor

64-bit bus

Input FIFO

&

Output FIFO

XML Parser

Predicate Evaluator

FPGA board

An Example for Illustration

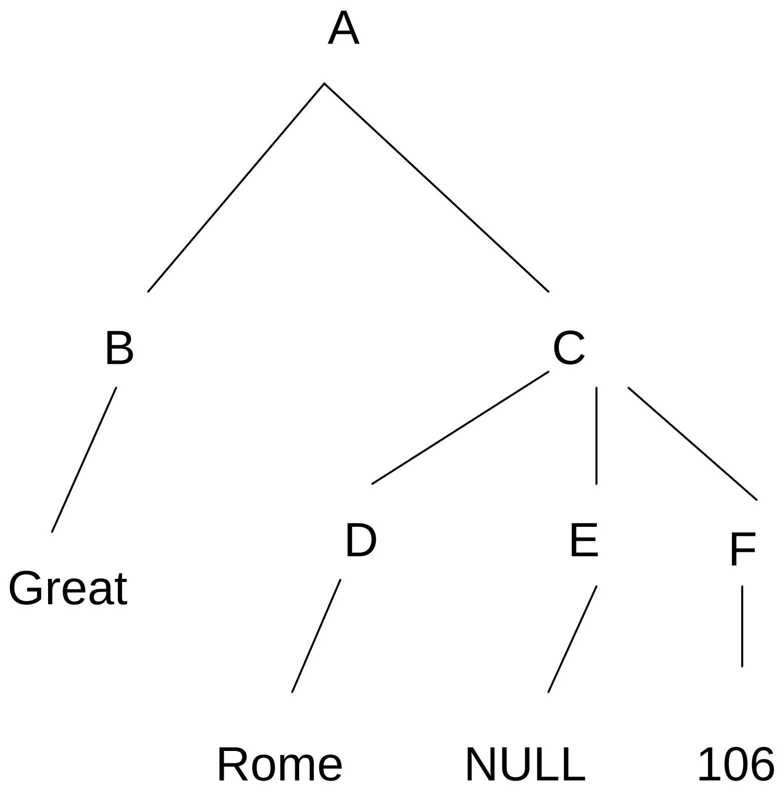
XML Document

```
<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



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.

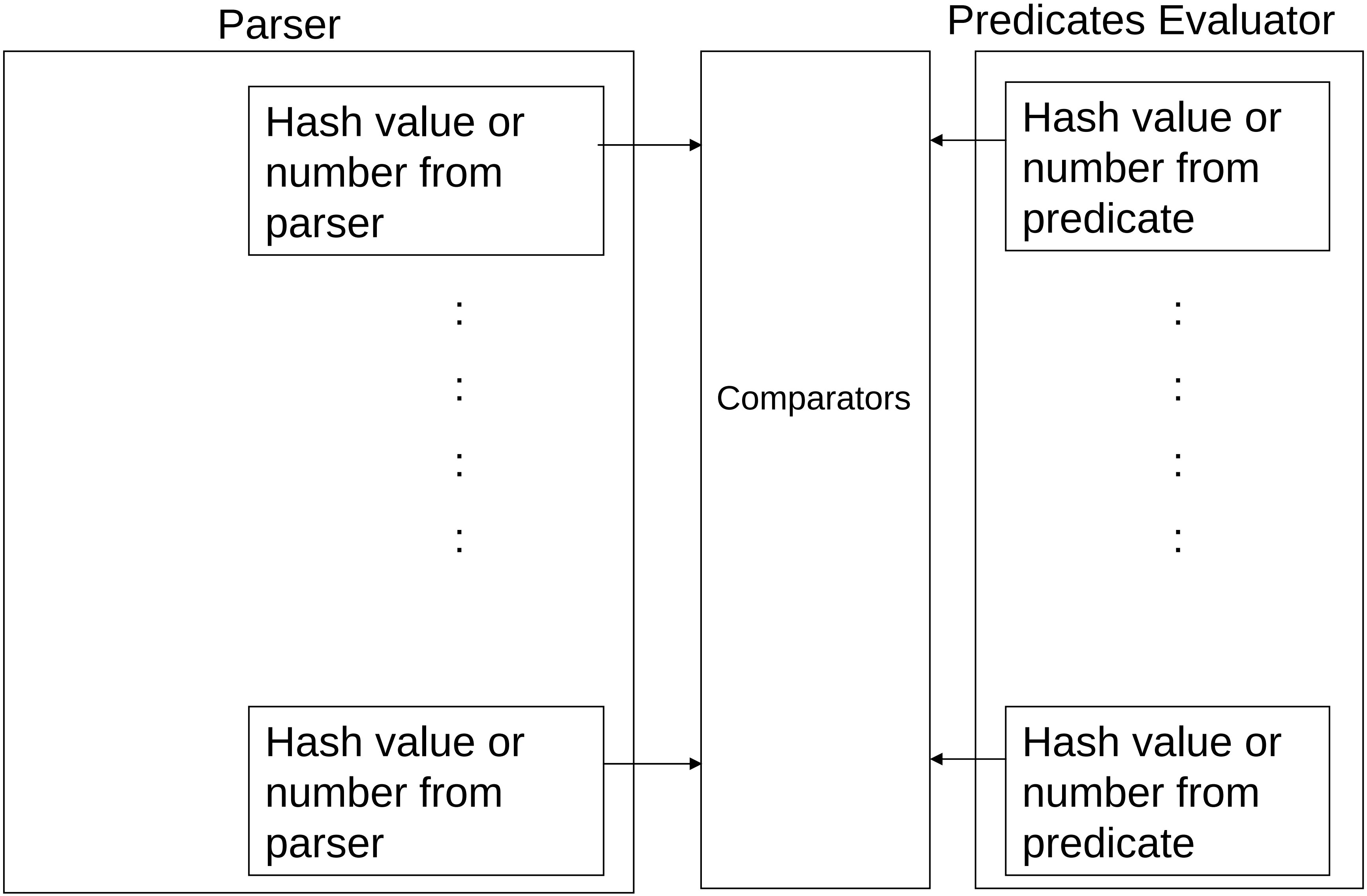
Table Generated by FPGA Parser

pointer to “Great”	length of “Great”	hash_value for “Great”
pointer to “Rome”	length of “Rome”	hash_value for “Rome”
pointer to “NULL”	length 0	hash_value 0
pointer to 106	integer part of 106*1000	

	pointer	length	hash value
/A/B	7	5	*xxx*
/A/C/D	22	4	*yyy*
/A/C/E	34	0	0
/A/C/F	41	106000	

Data will be sent back to the microprocessor and broadcast to the predicate evaluator backend logic

Comparison of Hash Value/Number



Parser

Predicates Evaluator

Hash value or number from parser

:

:

:

:

Hash value or number from parser

Comparators

Hash value or number from predicate

:

:

:

:

Hash value or number from predicate

(# 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

# Predicates	# of 64-bit words transferred	Processing time
0	104 (90 down plus 14 up)	45µs
11 (to 256)	104+4	48µs
1000 (to 1024)	104+16	52µs

- 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

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

* 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

# Predicates	Clauses per Predicate (Average)	# Slices for Parser & Predicates Evaluator (out of 33792)	Synthesis Time
600	1.333	943	233 sec
600	1.969	970	243 sec
600	3.288	1021	376 sec

* 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

# Predicates	# of Different Leaf-Values Used	# Slices for Parser & Predicate Evaluators (out of 33792)	Synthesis Time
600	10	670	210 sec
600	100	1021	376 sec
600	1000	1081	1005 sec

* 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.