Case Studies in Hardware Xpath Acceleration

SYSTOR'11, May 30–June 1, 2011, Haifa, Israel.

Dorit Nuzman, Victor Kaplansky, Sergei Dyshel, Alon Dayan
David Maze, Matthias Nicola, Glenn Marcy

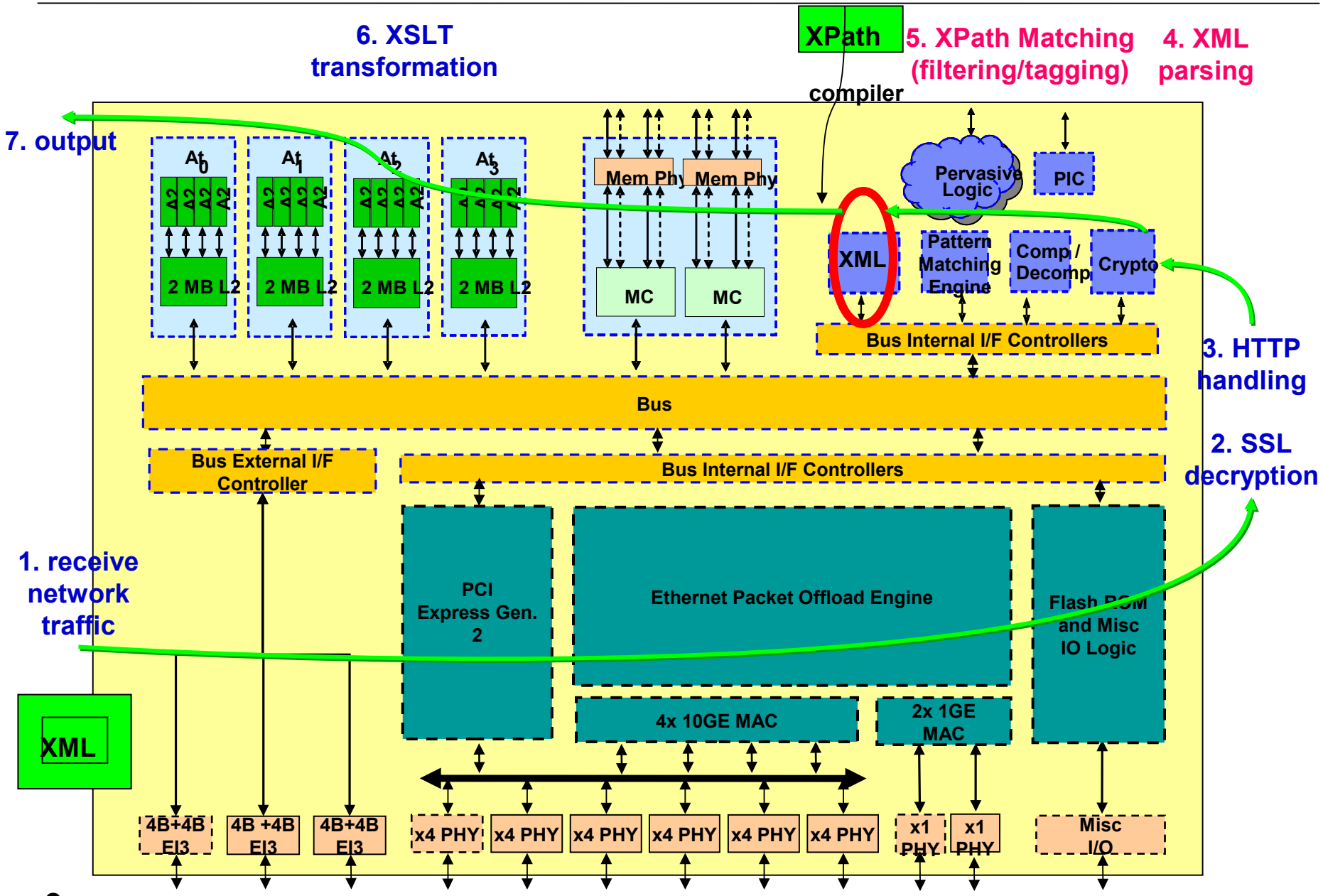
Main Goal and Results.

Acceleration of Xpath processing by Hardware in two real world applications – **WBM** and **DB2-pureXML**.

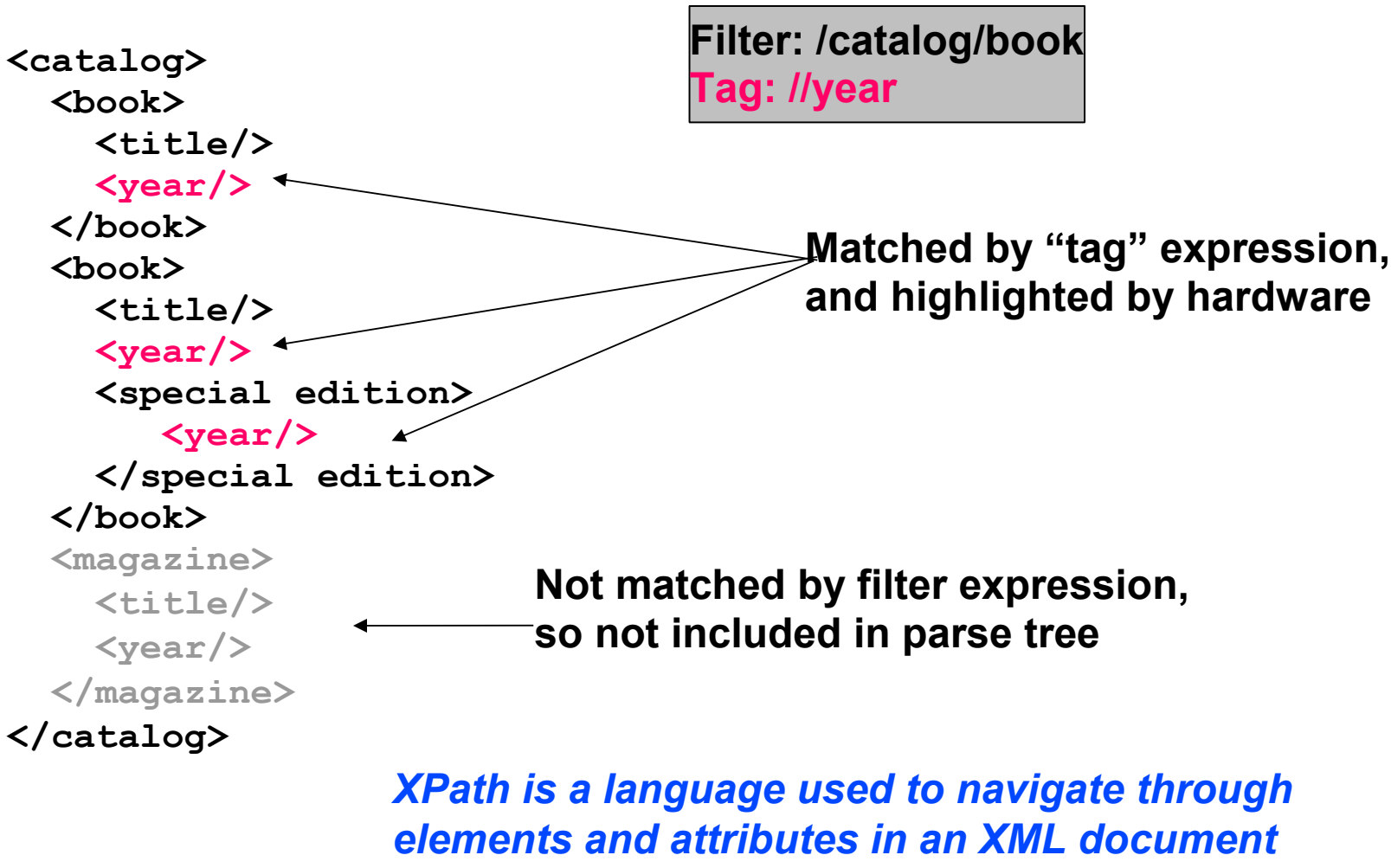
Websphere Business Monitor – 27% improvement in total running time.

DB2-pureXML – up to x6.2 improvement in total query processing time.

IBM's Power Edge of Network (PowerEN)



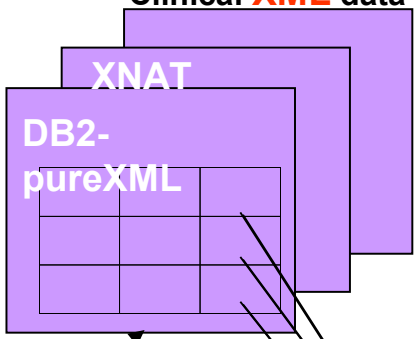
XPath Filtering and Tagging



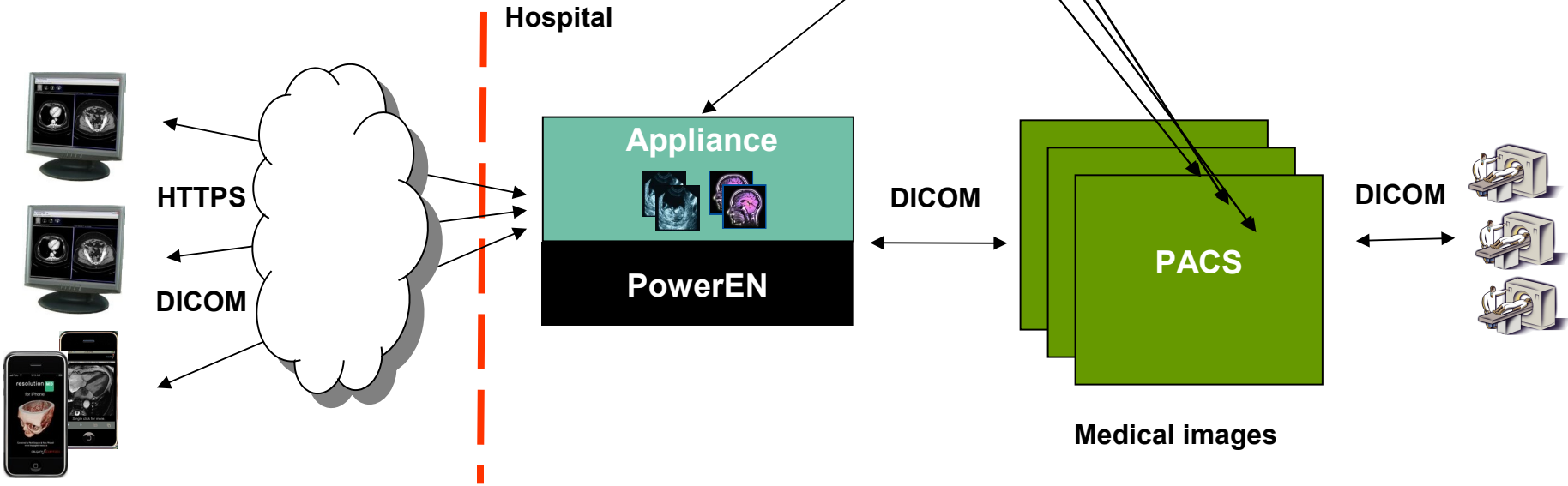
XPath is a language used to navigate through elements and attributes in an XML document

XPath acceleration opportunities: 1) XML in Healthcare / XML Databases

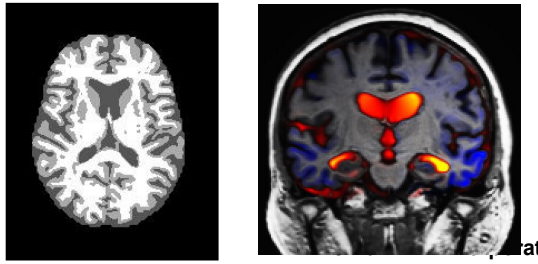
```
<SubjectData ID="ab123">
  <gender>male</gender>
  <handedness>left</handedness>
  <dob>1967-08-13</dob>
  <education>30</education>
  <ses>2</ses>
</SubjectData>
```



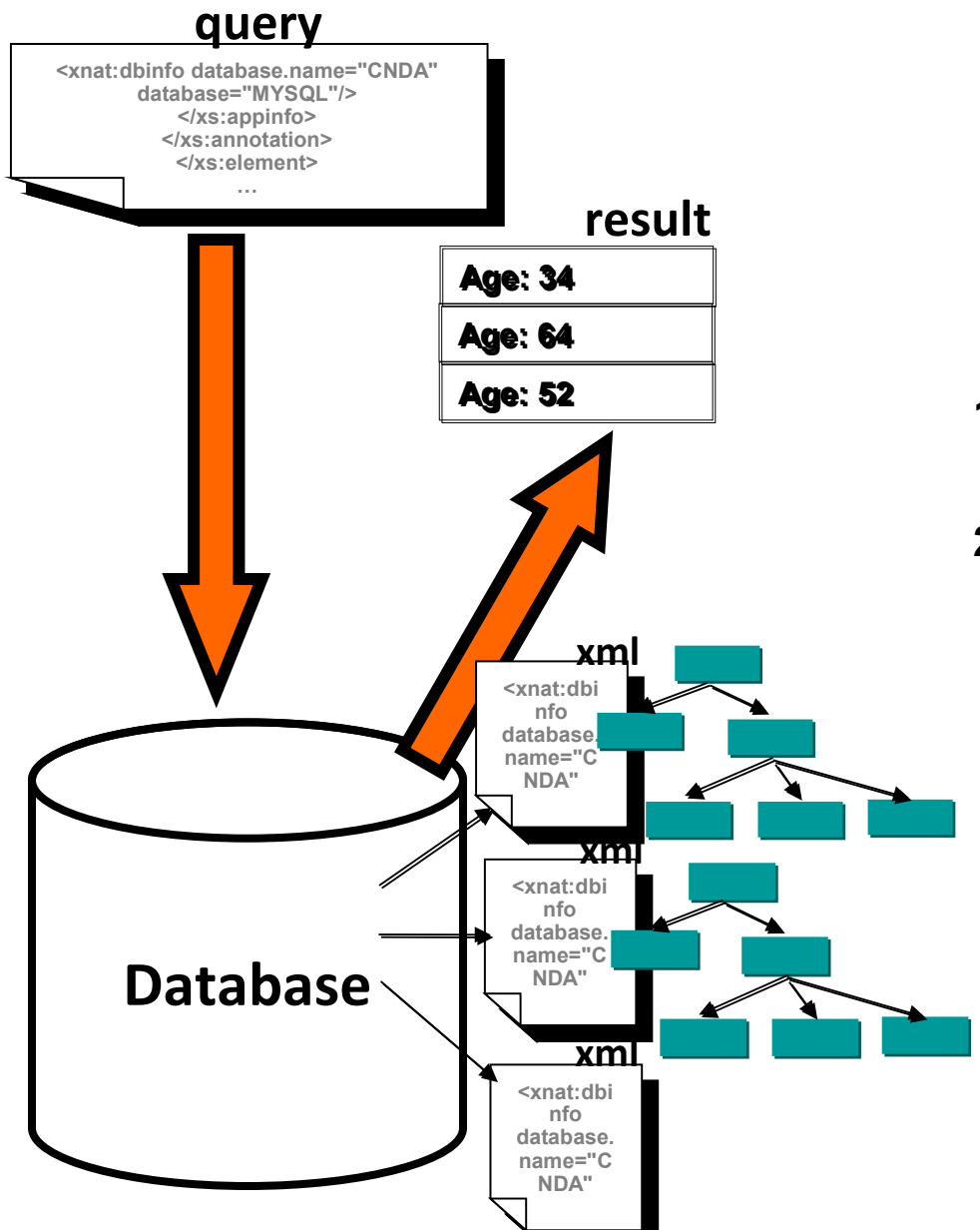
XML: de-facto standard for electronic medical health record interoperability



Medical images



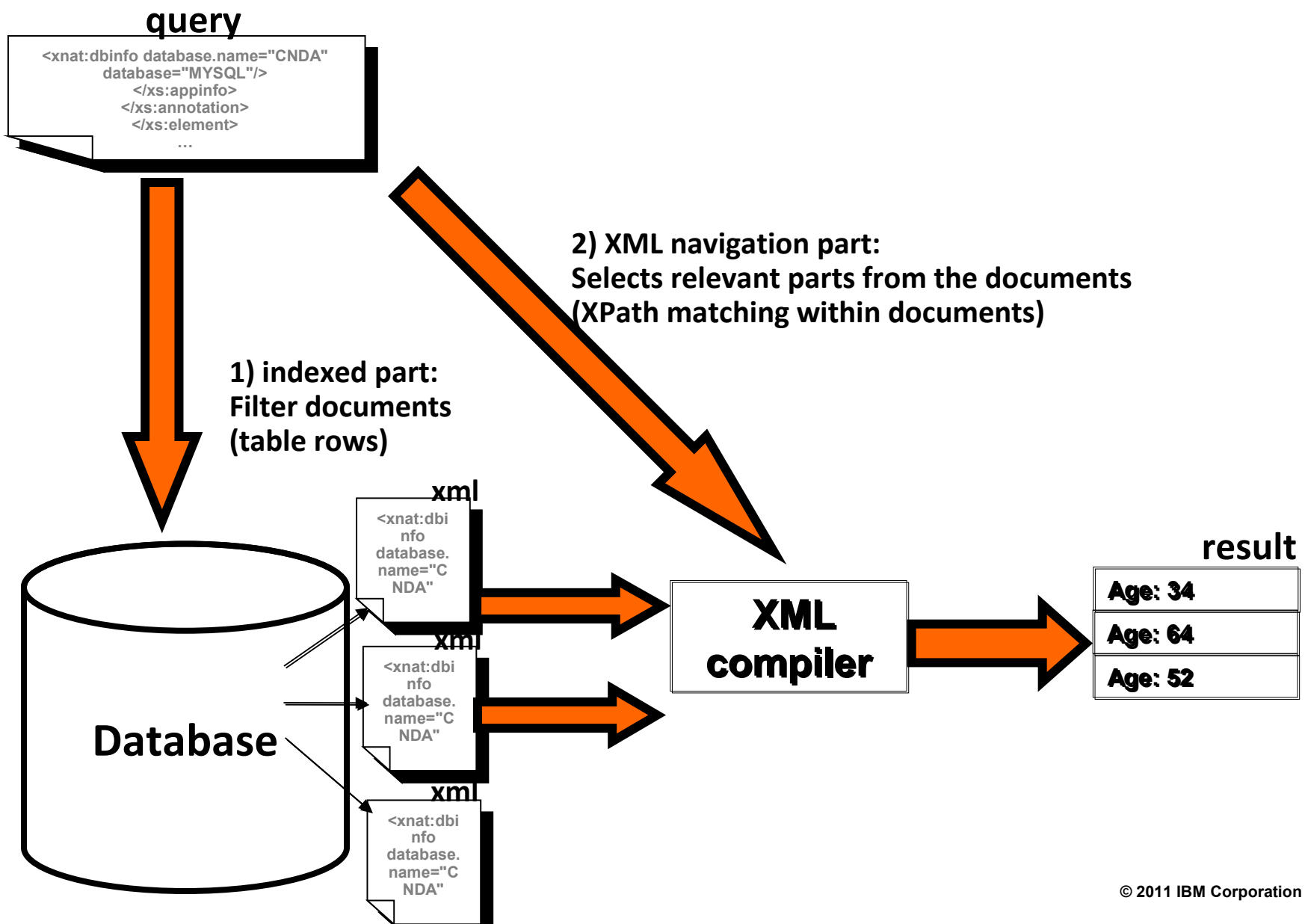
Current DB2-pureXML flow



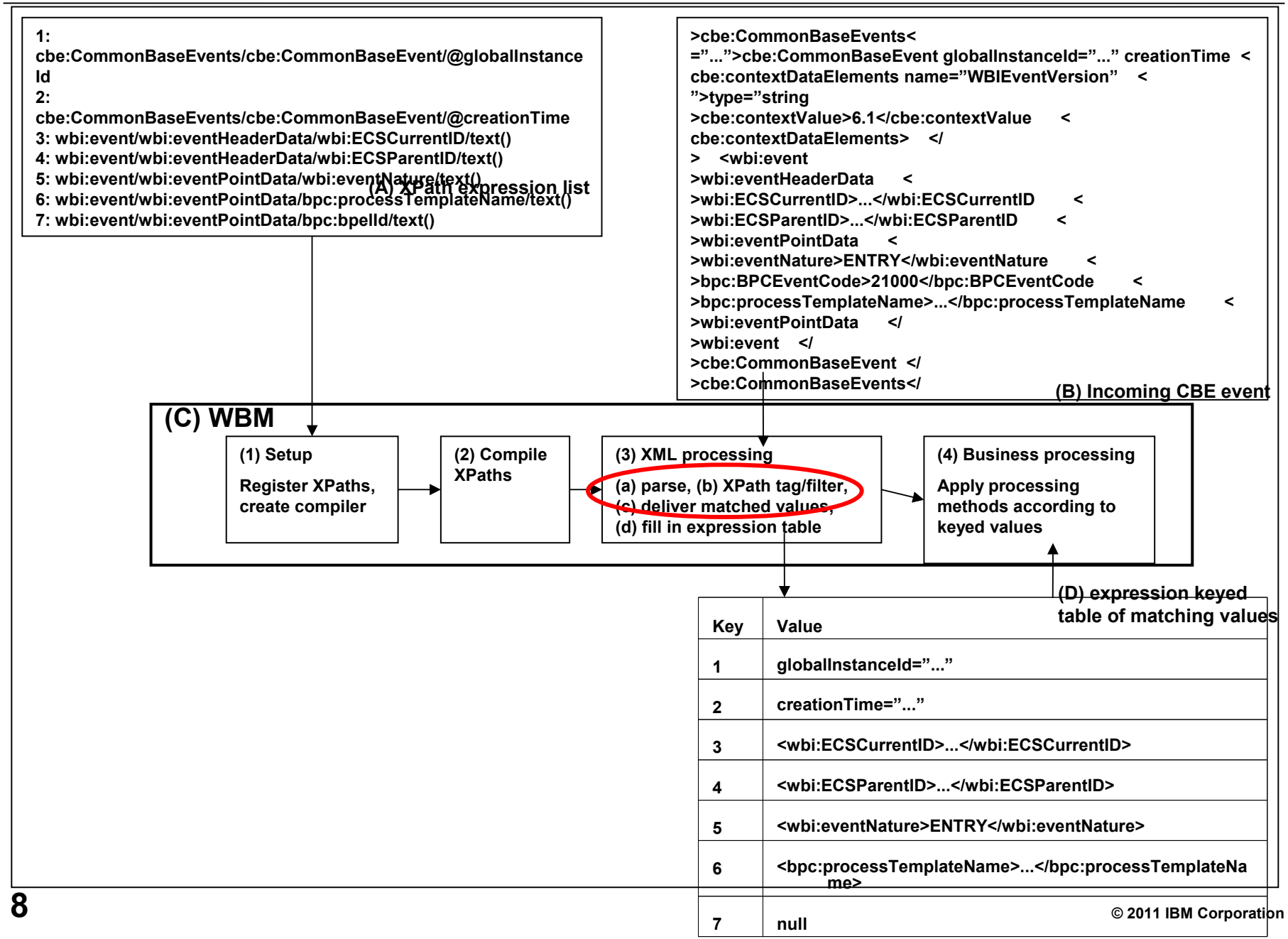
Processing may consist of:

1. **Table operations on indexed elements**
2. **Navigation of the XML documents**

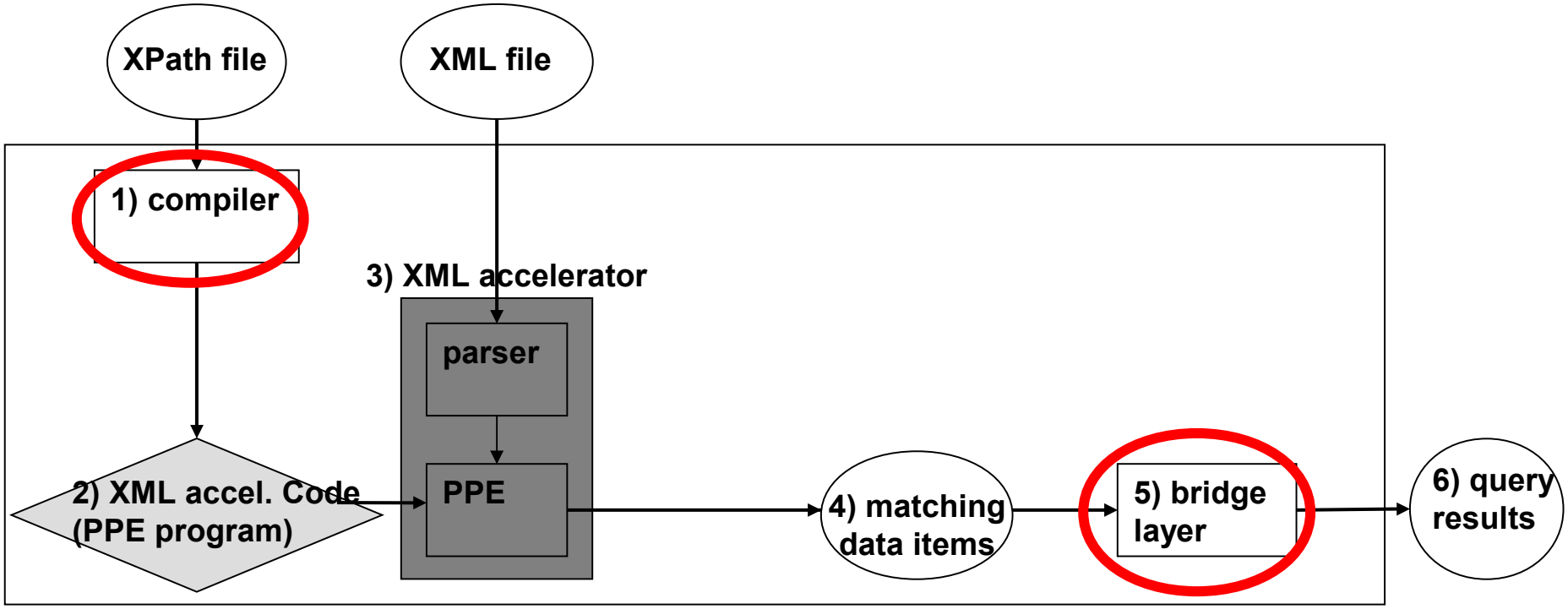
Proposed DB2-pureXML flow with PowerEN



XPath acceleration opportunities: 2) WebSphere Business Monitor

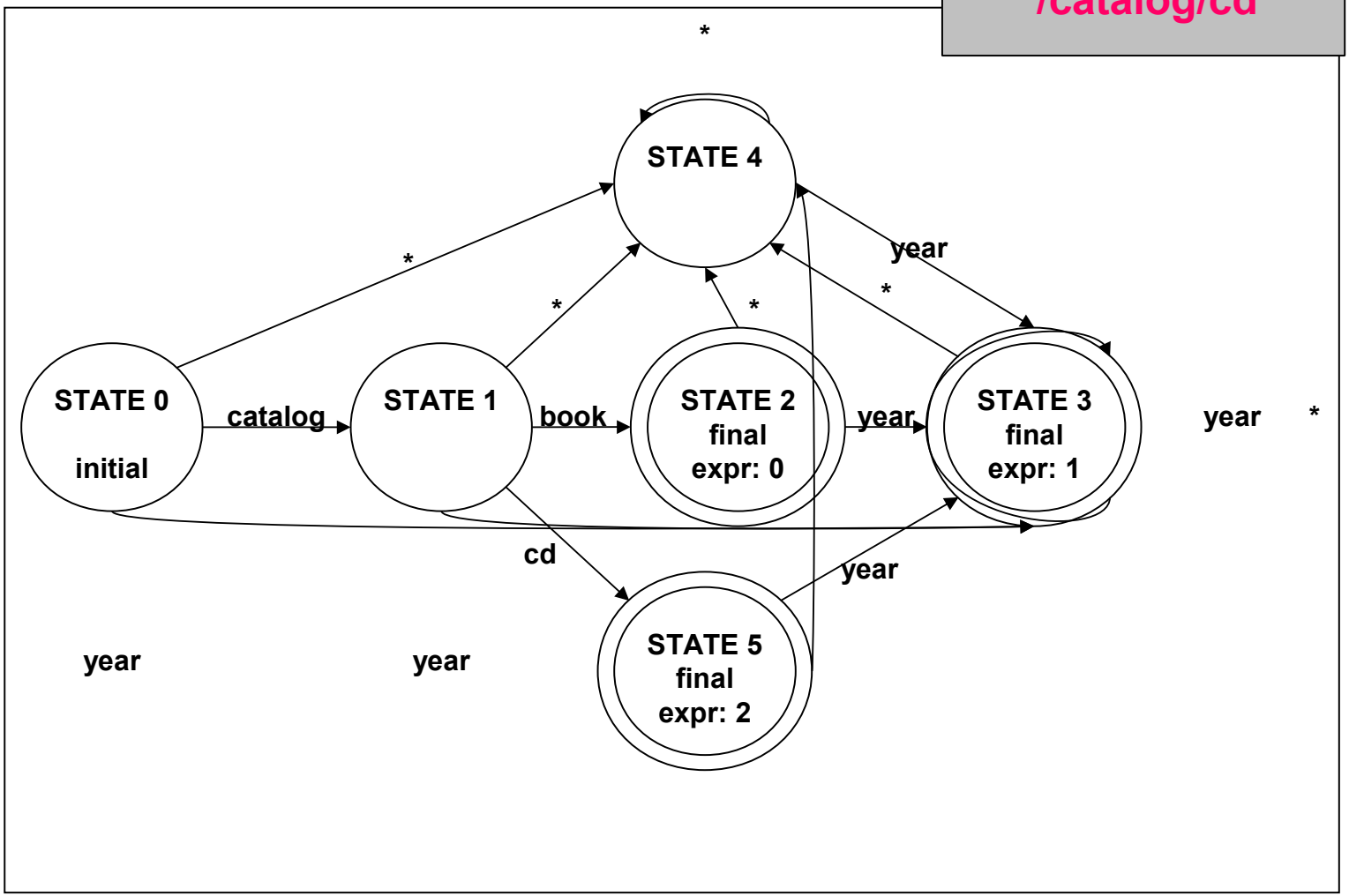


Technical details



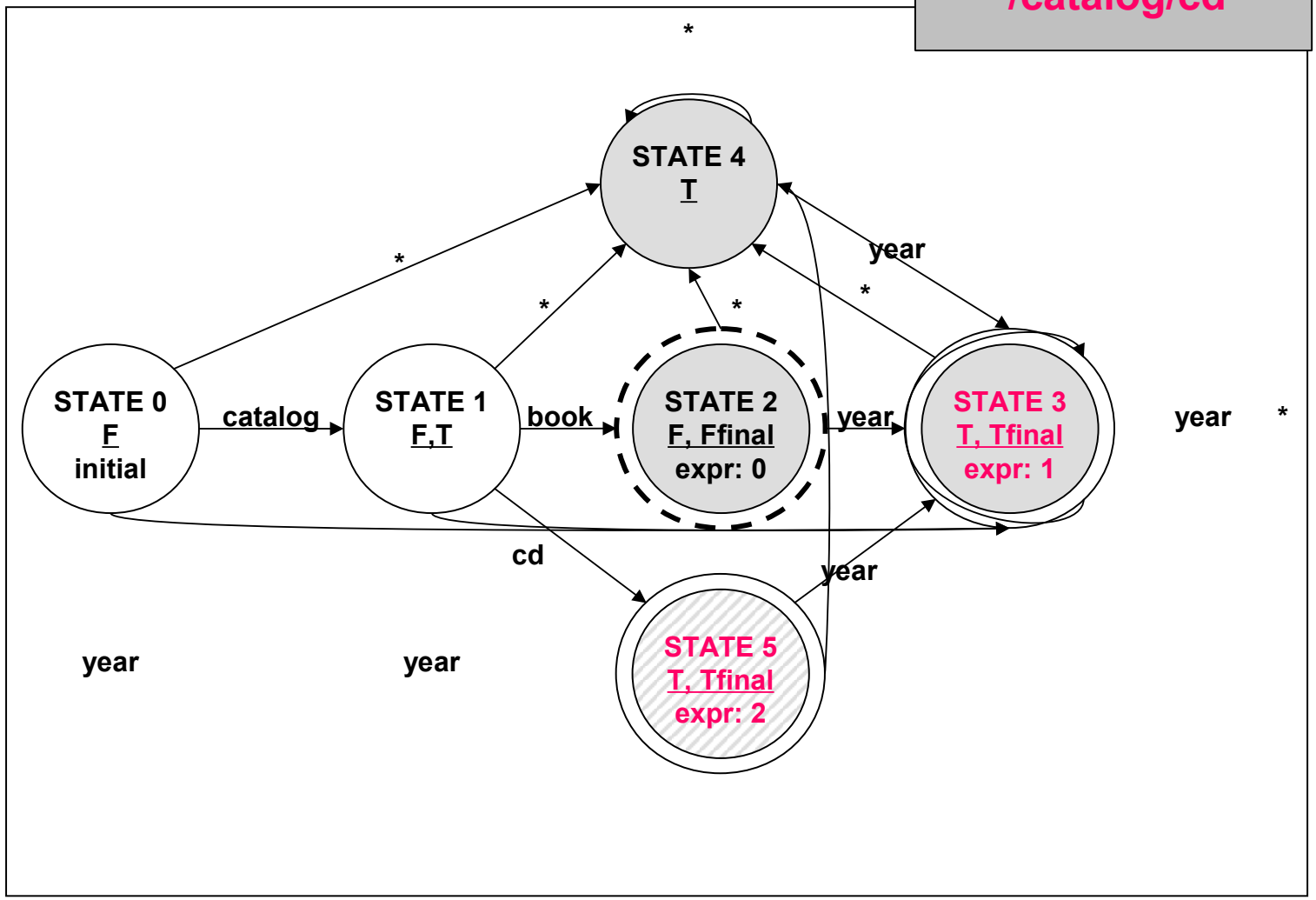
A few technical details: 1) the XPath compiler

Filter: /catalog/book
Tag: //year
/catalog/cd



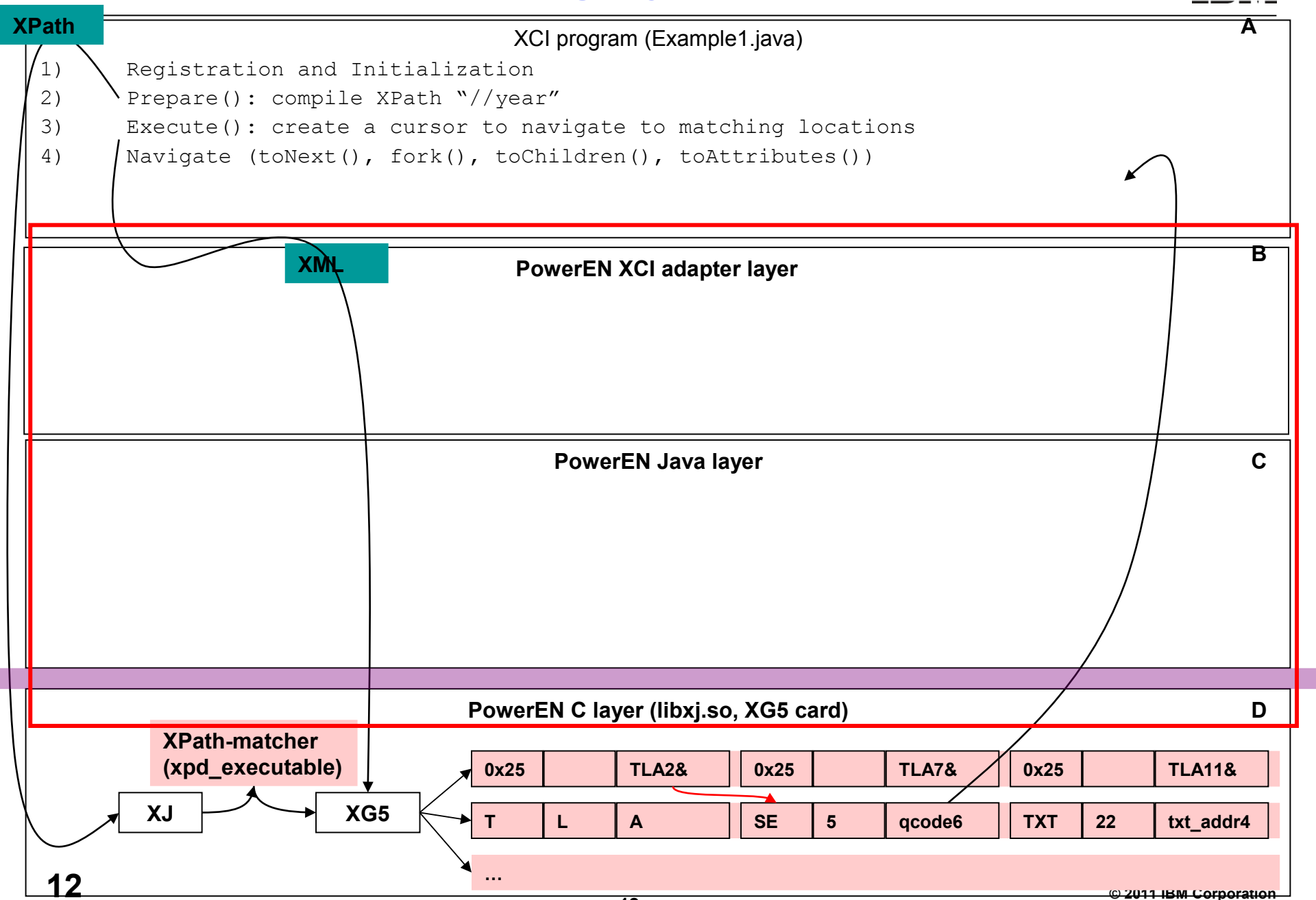
A few technical details: 1) the XPath compiler - con

Filter: /catalog/book
Tag: //year
/catalog/cd



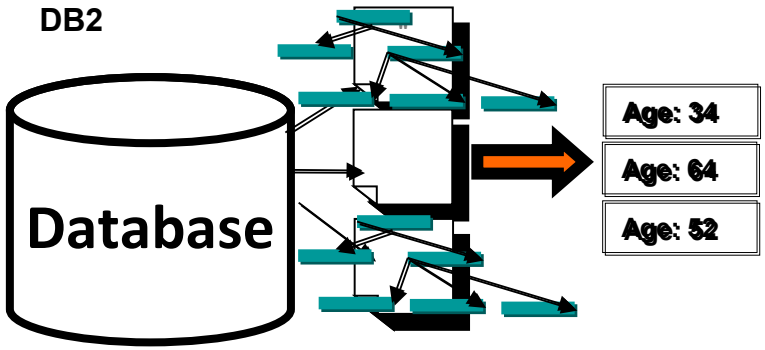
Note: streamable XPaths only (don't support **/catalog/book[special-edition]/year**)

A few technical details: 2) the bridge layer



The integrated experiment, Using JDBC

DB2



(1) Filter documents (rows)

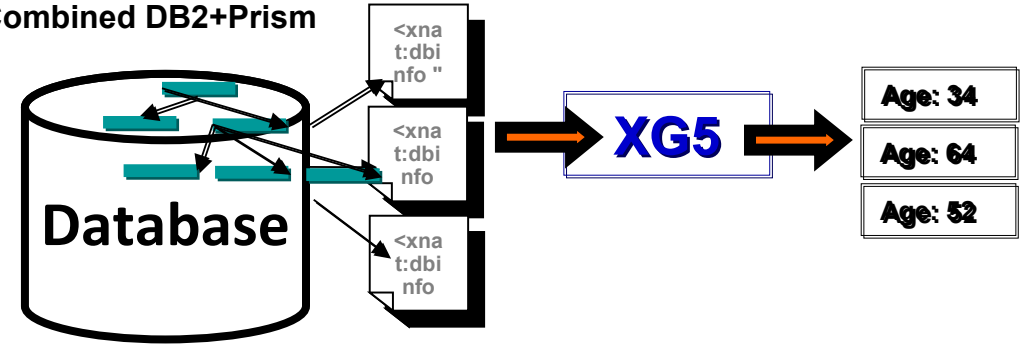
(2) Navigate the parsed document to find matches

(3) Serialize the results

(4) Transmit the results to the client

Processor:
1) dual x86 Harpertown Processors @2.83GHz

Combined DB2+Prism



(1) DB2 filter and serialize documents

(2) Send the XML document from host to Prism

(3) Parse the document to find matches
(+ compile the XPath query into a program that would run on the XML accelerator)

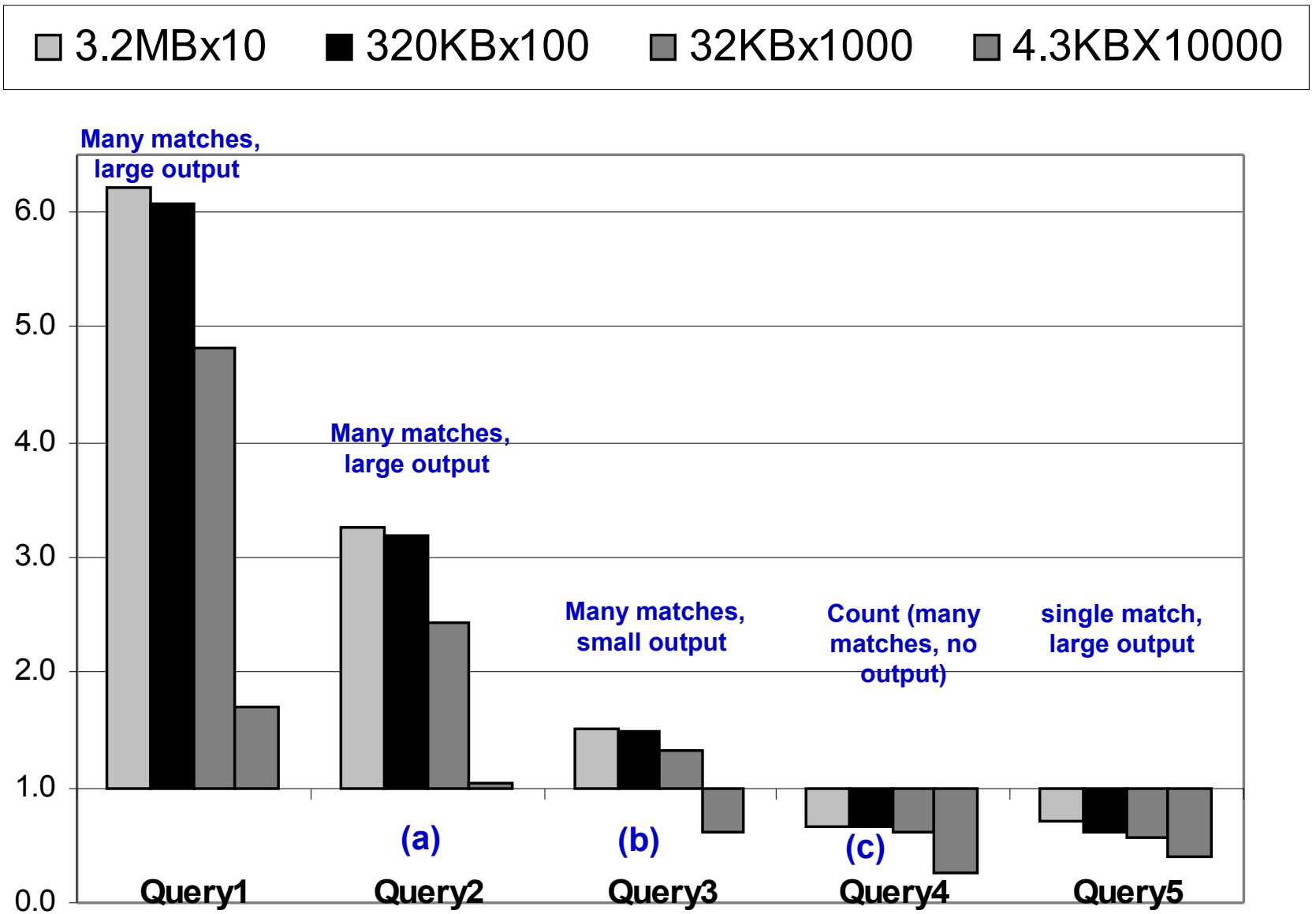
(4) Send the results back to the host

(5) Serialize the results

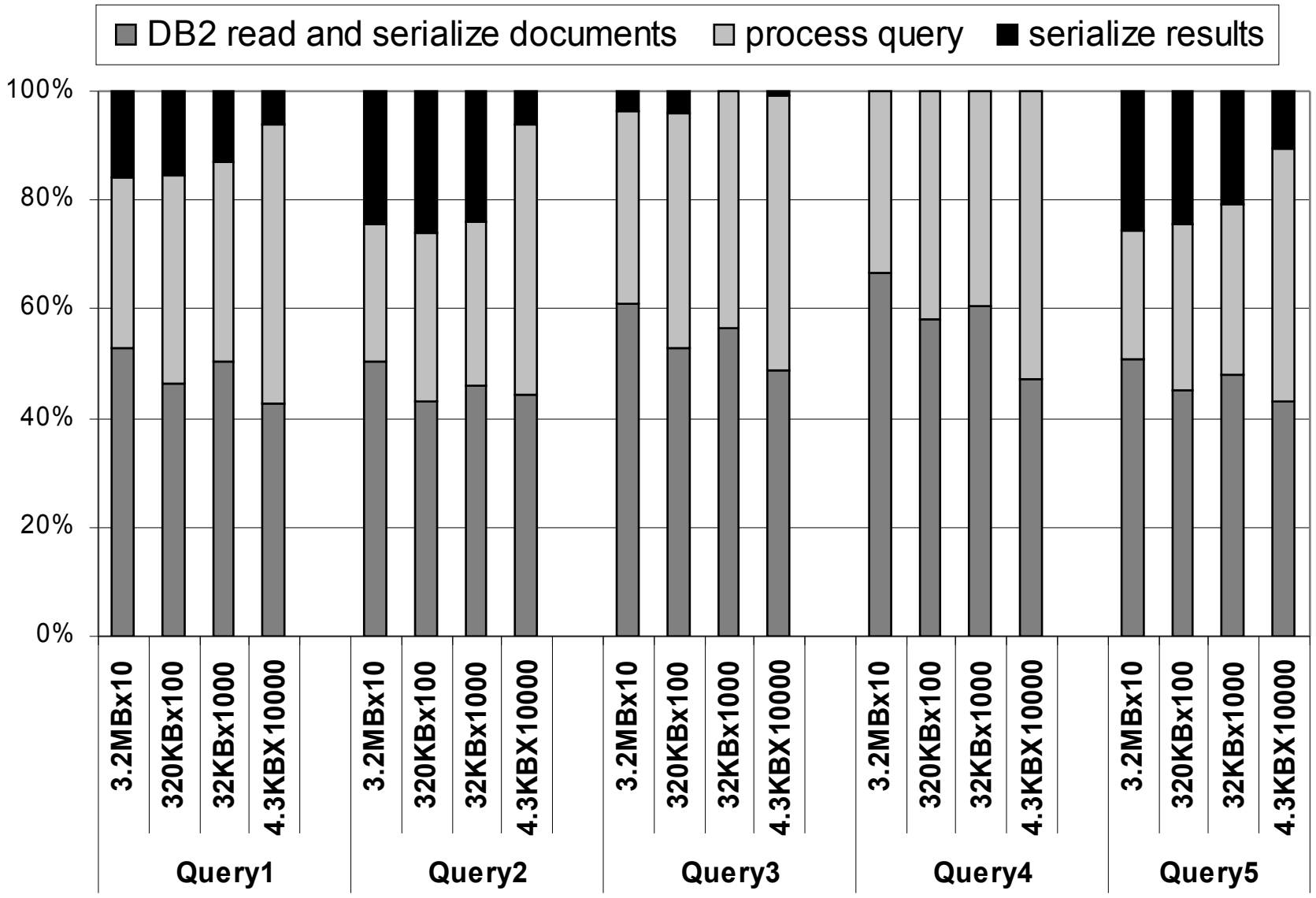
(6) Transmit the results to the client

Processor:
1) dual x86 Harpertown Processors @2.83GHz
2) PRISM offloading the XML processing

XPath query acceleration speedups, HL7



Breakdown of accelerated path, HL7 query



(a) Many matches, large output

(b) Many matches, small output

(c) Count (many matches, no output)

single match, large output

Websphere Business Monitor acceleration speedups

- **XML processing part improved by 27% →**

WBM Overall application improved by 11%

An efficient bridging layer is critical for overall accelerated performance

- buffering of requests to the accelerator

- reduced JNI calls/Java–C conversions

Applications have to use the “right” API

Conclusions:

- High potential for acceleration can be found in applications using large documents and XPath queries matching large numbers of XML nodes and producing large outputs, such as in the healthcare and life sciences domains.

- Limited potential for acceleration can be found in applications using small documents and XPath requests matching small numbers of XML nodes or producing small outputs, such as in the event processing and financial domains.

- An efficient bridging layer is critical for overall accelerated performance. Optimizations to the software bridging layers, such as buffering of requests to the accelerator, reduced JNI calls and Java–C conversion overheads, yielding a 33% improvement to the WBM accelerated path, and up to 2.7x improvement to the HL7 accelerated query processing path.

Future Work

- extend the applicability of XPath acceleration coprocessors

- increase speedups:

Devise a cost model that can automatically identify scenarios that can profit from XPath acceleration.

Extend XML APIs to express more involved XPath scenarios (such as simultaneous filtering and tagging, and multi-step XML processing).

Specifically in the native XML Database domain, data serialization costs are relatively high, and support for compact data formats by the hardware XPath accelerator is critical.

The End

Questions?