Balisage Symposium 2009 Memory Management in Streaming: Buffering, Lookahead or none Montréal

Présentation Innovimax

De nombreuses technologies émergent chaque jour et toute société a besoin de s'approprier et d'intégrer ces atouts pour leurs développements. A travers la jungle des sigles, XML, Java, .Net, SOA, XSLT, AJAX, XUL, vous cherchez à comprendre et à utiliser la bonne technologie. La société *Innovimax* a été créée dans cette optique. *Innovimax* vous accompagne dans toutes les phases de votre projet en vous fournissant le conseil, le suivi, les prestations et la formation nécessaire à sa bonne réalisation.

Basée à Paris (France), *Innovimax* est une société privée spécialisée en technologies émergentes et en innovations. *Innovimax* propose donc ses services regroupés autour de quatre pôles : *Média*, *Software*, *Consulting* et *Learning.*

Contactez-nous / Contact us

Innovimax
9, impasse des Orteaux - 75020 Paris

Tél: +33 9 52 47 57 87
Fax: +33 1 43 56 17 46

contactus@innovimax.fr
http://www.innovimax.fr

SARL au capital de 10.000 €
RCS Paris 488.018.631

Innovimax Learning

Le pôle **Innovimax Learning** est le second pôle important de la société. Point clefs de la réussite de toutes évolutions technologiques, la formation se doit d'être clair, accessible et adaptée. Les technologies émergentes sont légions et il vous semble difficile de faire le tri parmi les sigles : HTML, XML, XSLT, CSS, AJAX. Pour ce faire, le département Learning d'Innovimax vous propose des formations pour vous y retrouver dans ce dialecte et savoir quels sont les technologies dont vous avez besoin.

A destination des décideurs, les formations *Manager* vous propose des formations concrètes expliquant les tenants et les aboutissant de chaque technologie, les gains attendus et les success stories.

A destination des utilisateurs/collaborateurs, les formations *Client* vous propose des formations essentiellement axées sur les technologies en place dans leur environnement de travail et rétablisse les réflexes à prendre avec les nouvelles technologies (sauvegarde, sécurité, spam, etc.)

A destination des acteurs technologiques, les formations *Designer* vous propose des formations ciblées sur votre domaine (Web, graphique, applicatif) afin de vous enseigner les bases approfondies de chacune des technologies et d'être en capacité de mettre rapidement en application ces technologies.

Innovimax & standards

W3C (World Wide Web Consortium), ISO, AFNOR :

Innovimax is a W3C member at XSLT, XML Processing, XML Core and MathML WG and endeavour all XML ecosystem.

Innovimax is also an actif member at ISO and AFNOR.

Innovimax
R & D

This work has been partially funded by the French Agence Nationale de la Recherche under reference ANR-08-DEFIS-004

Greetings – 1/2

<Hello>

</Bonjour>

innovimax learning

Mohamed ZERGAOUI

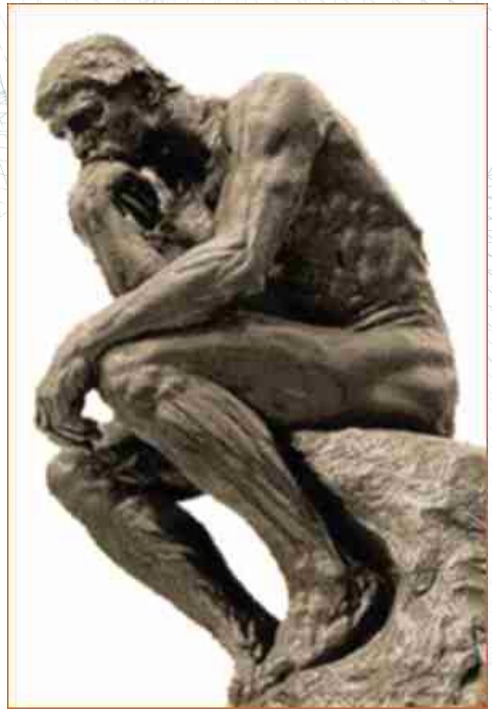
- INNOVIMAX (small French company)
- W3C Member (XSL, XProc, XQuery, XML Core …)
- ISO JTC 1/SC 34 French Head of Delegation
- AFNOR (French national body); French Official Publication Office (DJO); OECD Publication
- XML Guild
- Co-organizer of XML Prague **(13/14 March 2010)**

Memory Management in Streaming: Buffering, Lookahead or none

Streaming processing

What is streaming processing?

- Difficult to define
- Related to events
- Related to memory
- Related to latency
- Eventually able to process infinite input (Turing machine ?)



What is it?

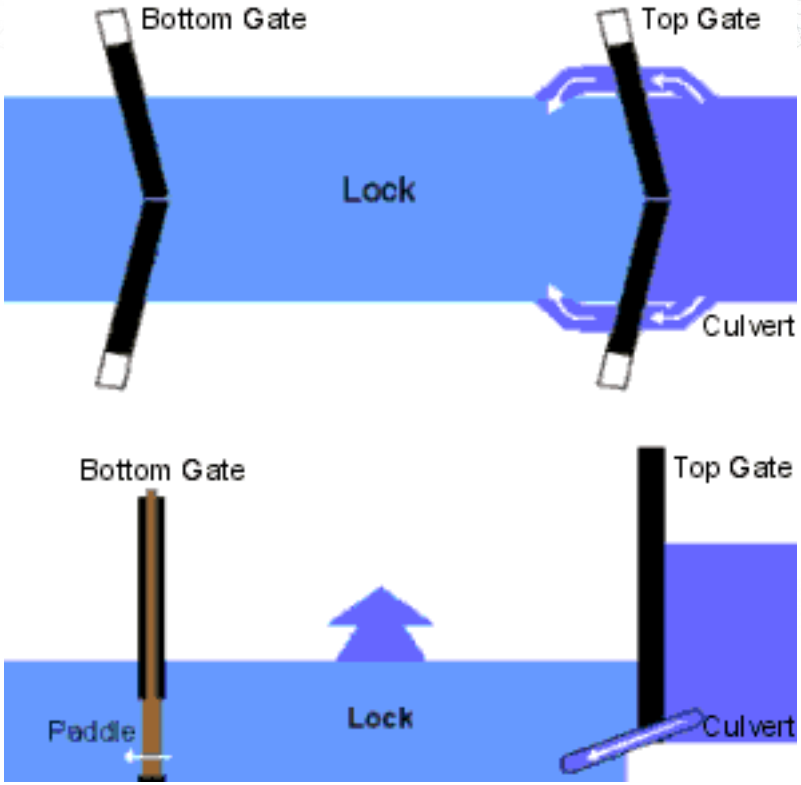
- Proposal (very broad):

  A process that do not mandates a blocking preloading of the whole main input document. This process will heavily rely on assertion

- Rule out Tree like input (DOM, etc.)
- Rule out Frozen stream like input (VTD-XML, etc.)
- Rule out Full Validation (sic!)
- Generate Dynamic Invalid Assertion Error in middle or at the end of the process (not well formed, Schema invalid, duplicate id/key, invalid assertion, etc.)

Why?

- Well, this is XML processing today



Why?

- ...You see what I mean....

Why ?

- ...But we want this: less or no latency !

Events

- The documents is provided a sequence of events
  - StartDocument
  - StartElement
  - Text
  - PI
  - Comment
  - EndElement
  - EndDocument

Latency

- The unit of distance in events between
  - First input event/First output event
  - First input event/First **significative** output event
  - Last input event/Last output event (with finite document)
  - Then it depends on relation between the input and the output (if there is some repitition/motif/pattern)

Memory

- Many way to see that
- Extreme way: no extra memory at all
- Moderate way: allow some bounded lookahead
- Allow buffering of different categories:
  - Buffering the input or the output
  - Bounded by a constant size
  - $O(X^{\varepsilon})$ where $0 < \varepsilon < 1$ and X is the input size (growing slower than linear)

Extreme way

Extreme way: No memory at all

- What does it mean?
- Impossible in real life (not really useful)
- Caching is everywhere (CPU caching, Disk Caching, IO caching, Parser caching, etc.)
- So we end up having a bounded memory volume (small but not null)
- Interestingly, it is what most of people define as « streaming »
  - It just more restrictive than real life!

Moderate way

Memory

- Many way to see that
- ~~Extreme way: no extra memory at all~~
- Moderate way: allow some bounded lookahead
- Allow buffering of different categories:
  - Buffering the input or the output
  - Bounded by a constant size
  - $O(X^{\varepsilon})$ where $0 < \varepsilon < 1$ and X is the input size (growing slower than linear)

Moderate way: allow some bounded lookahead

- Definitively related to the input document
- All the trick is based on reading ahead the input file to detect some cases (but not all the document...)
- Less general than buffering **(forward memory)**
- Hard to bound
- The limit rely on assertion (→ dynamic error when exceeded)

Example of lookahead (1/2)

- A[B] ==> two cases:
  - Structure imposes that B must be the first child of A => lookahead of 1 event
  - General case lookahead of **at most** entire A
- A[count(*) mod 2 = 1] ==> lookahead of **exactly** entire A *(unless very particular schema)*
- A[last()] ==> lookahead of entire A **at least** and sometimes even more depending on the structure of the files (<root><A/><B/><B/>....<B/></root>); at **most the rest of the parent element**

Example of lookahead (2/2)

- These were moderate examples
- Most of the time there is simultaneous/concurrent lookahead on the input file
- They could be combined in multistate automata
- Again, preempting the size needed for the lookahead is hard in many cases
- Schemas are not sufficient

For profiling the lookahead

- You need some information on the document
- You need to make a lot of assertions
- You must be ready to fail during or at the end of the process

Extra Information on input document

- Schemas
- Assertions
- Some information on ordering of some elements
- Some information on uniqueness and key
- Average/min/max size of particular elements in the document (→ Statistical informations)

Buffering

Why do we need buffering?

- To make the software simpler to write
- To make ordering or sorting
- To make preemptive uniqueness/key checking
- To avoid having errors at the end of the process

Well I should precise « buffering of the input »

- Indeed, the computation itself may need buffer
- And you may end up finding Pathological case of Complexity & Computability Theory:
  - EXPSPACE: Comparing two languages through RE
  - Recursive non primitive function: Ackermann

Buffering

- Proposal:

  Buffering of the input is always related to sorting and particular ordering of information, or for global preemptive validation.

- Following this, every process should be transformed into

  $$Process = Pre\text{-}odering \circ Streaming$$

  Or $$Process = Streaming \circ Post\text{-}ordering$$

Buffering and ordering

- We can replace

```
<xsl:for-each-group select='...' group-by='...'>...</>
```

by

```
<xsl:for-each-group select='...' group-adjacent='...'>...</>
```

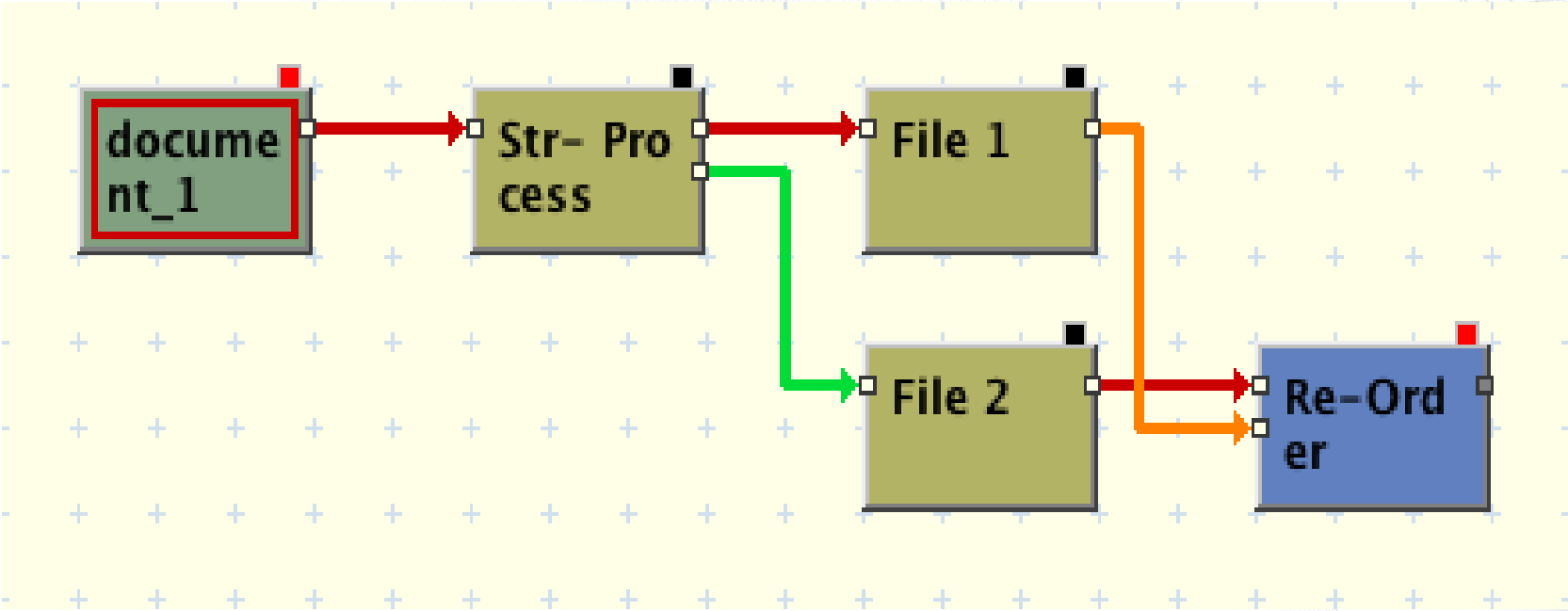
- But first we have to know...
- ... that the input is already ordered …
- … with respect to the right ordering !!

Buffering and ordering

If the input is already sorted then	
<xsl:sort ....>	Ø
<xsl:perform-sort ...>....</>	<xsl:sequence ...>...</>
<xsl:for-each-group select=”...” **group-by**=”...”>...</>	<xsl:for-each-group select=”...” **group-adjacent**=”...”>...</>

Buffering and ordering

- One way to solve the case of post sorting is to write the different chucks in different files using `<xsl:result-document`



Buffering and ordering

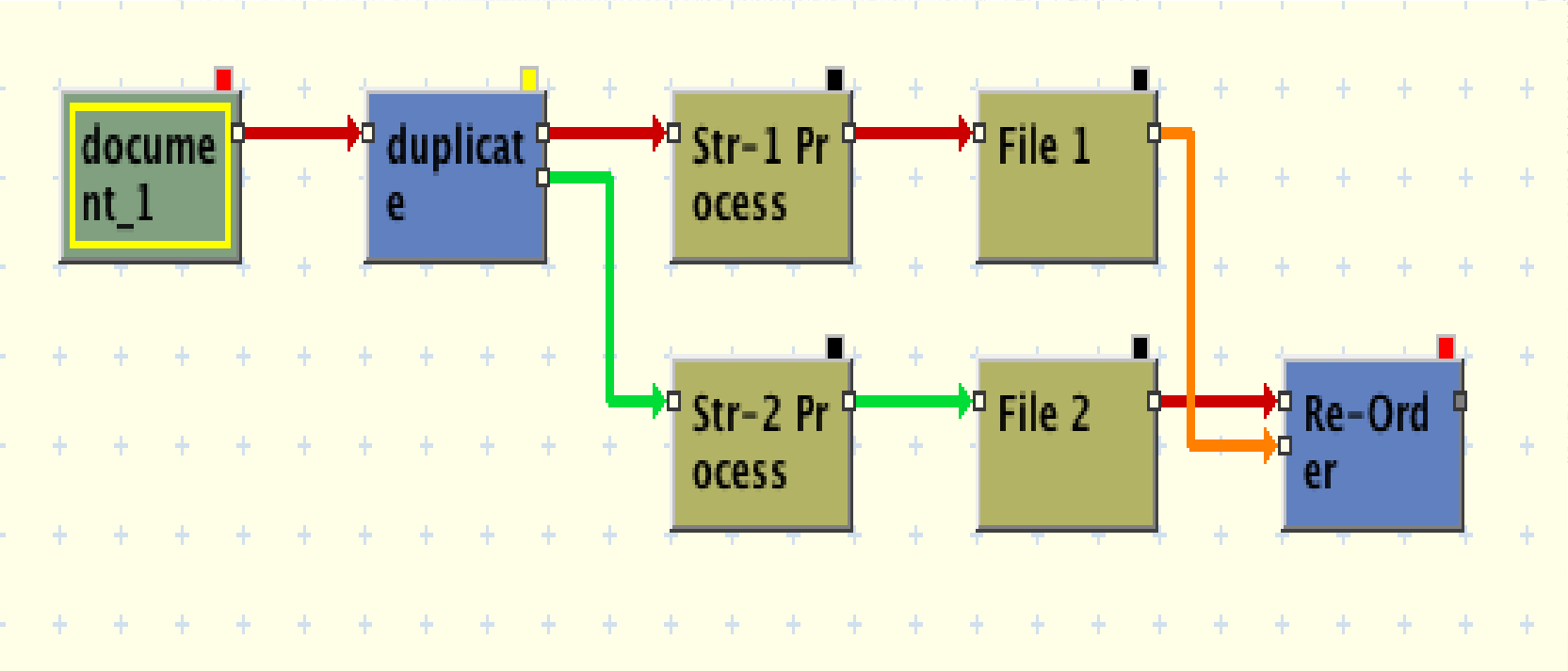
```
<xsl:variable name="a">sequence constructor a</>
<xsl:variable name="b">sequence constructor b</>
<xsl:copy-of select="$b"/>
<xsl:copy-of select="$a"/>
```

Could be transformed to

```
<xsl:result-document href='2.xml'>sequence
constructor a</>
<xsl:result-document href='1.xml'>sequence
constructor b</>
```

Buffering and ordering

- One other way to solve it is to has TWO differents processing in PARALLEL



Buffering and ordering

```
<xsl:variable name="a">sequence constructor a</>
<xsl:variable name="b">sequence constructor b</>
<xsl:copy-of select="$b"/>
<xsl:copy-of select="$a"/>
```

Could be transformed to

```
Process 1 : sequence constructor a
Process 2 : sequence constructor b
```

Categorisation

Small lookahead process

- Parsing
- Validation (DTD, RelaxNG)
- Typing (only for 1-pass preorder typing 1PPT, Martens & Neven 2006)
- Simple identity like processing: mapping, decorator, almost-copy
- Aggregate processing: summation, mean.
- Grouping adjacent with simple rules

Lookahead process

- Conditionnal Type Assignment (XML Schema 1.1)
- Full Typing (may need huge lookahead depending on the schema)
- Almost sorting: if you allow multiple consecutive process until fixed point

Long Lookahead process

- <r><a/><b/><c/><d/><e/></r>
- <r>
  <n>
  <n>
  <n>
  <a/>
  <b/>
  </n>
  <n>
  <c/>
  <d/>
  </n>
  </n>
  <n>
  <e/>
  </n>
  </n>
  </r>

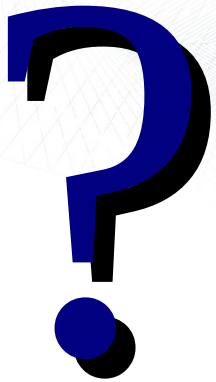
Streaming – Next – 1/2

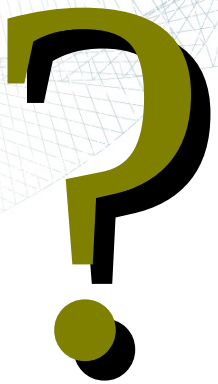
Next

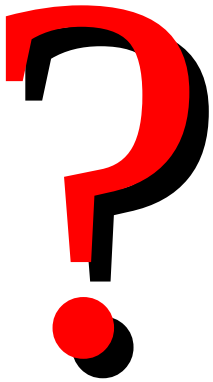
CODEX Project

- INRIA, CNRS, ENS, Jussieu (Paris), Lille, Tours, Lyon...
- Innovimax
- ANR public project
- Implementation of efficient processing tooling and of state of the art algorithm
- Be in sync with standards (XSLT 2.1, XProc, NVDL, etc.)

Questions ? – 1/1







innovimax