Frontal and multi-frontal solvers: Graph grammar based model of concurrency

Maciej Paszynski
Department of Computer Science
AGH University of Science and Technology, Krakow, Poland
maciej.paszynski@agh.edu.pl
http://home.agh.edu.pl/paszynsk
http://www.ki.agh.edu.pl/en/staff/paszynski-maciej
http://www.ki.agh.edu.pl/en/research-groups/a2s

Main collaborators

Victor Calo (KAUST)
Leszek Demkowicz (ICES, UT)
David Pardo (IKERBASQUE)
GENERATION OF 1D ELIMINATION TREE

1D elimination tree obtained by executing productions \((P1)-(P2)^2-(P2)^2-(P3)^6\)
We assign indices to grammar productions in order to localize the places where the graph grammar productions were fired:

$$(P1)\rightarrow (P2)_1\rightarrow (P2)_2\rightarrow (P2)_3\rightarrow (P2)_4\rightarrow (P3)_1\rightarrow (P3)_2\rightarrow (P3)_3\rightarrow (P3)_4\rightarrow (P3)_5\rightarrow (P3)_6$$
TRACE THEORY BASED SCHEDULER

Alphabet:
\[ A = \{ (P1), (P2)_1, (P2)_2, (P2)_3, (P2)_4, (P3)_1, (P3)_2, (P3)_3, (P3)_4, (P3)_5, (P3)_6 \} \]

Dependency relation for construction of the elimination tree
\[ (P1)D\{(P2)_1,(P2)_2\} \]
\[ (P2)_1D\{(P2)_3,(P2)_4\} \]
\[ (P2)_3D\{(P3)_1,(P3)_2\} \]
\[ (P2)_4D\{(P3)_3,(P3)_4\} \]
\[ (P2)_2D\{(P3)_5,(P3)_6\} \]
TRACE THEORY BASED SCHEDULER

Dependency graph

(P1) 
(P2)1 
(P2)2 
(P2)3 
(P2)4 
(P3)1 
(P3)2 
(P3)3 
(P3)4 
(P3)5 
(P3)6
TRACE THEORY BASED SCHEDULER

Dependency graph
Scheduling according to Foata Normal Form:

\[
\begin{bmatrix}
a_1^1 a_2^1 \ldots a_l^1 \\ a_1^2 a_2^2 \ldots a_l^2 \\ \vdots \\ a_1^n a_2^n \ldots a_l^n
\end{bmatrix}
\]

where

\[
a_i^k \in A \text{ (alphabet)}
\]

\[
\forall k \forall i, j \in \{1, \ldots, l_k\} \quad a_i^k I a_j^k \quad i<>j \quad \text{where } I = AxA \setminus D
\]

\[
\forall k \forall i \in \{1, \ldots, l_k\} \exists j \in \{1, \ldots, l_{k-1}\} \quad a_i^{k-1} Da_j^k
\]

Thus, the execution of the solver consists of several steps, where independent tasks are executed in concurrent, interchanged with the synchronization barriers.
PROCESS OF THE ELIMINATION
EXPRESSED BY GRAPH GRAMMAR PRODUCTIONS

Graph grammar production construction local matrix for the first sub-interval

Graph grammar production construction local matrix for the i-th sub-interval

Graph grammar production construction local matrix for the last sub-interval
PROCESS OF THE ELIMINATION
EXPRESSED BY GRAPH GRAMMAR PRODUCTIONS

Generation of frontal matrices at leaves of the elimination tree expressed as
the execution of graph grammar productions (A1)-(A)^4-(AN)
PROCESS OF THE ELIMINATION
EXPRESSED BY GRAPH GRAMMAR PRODUCTIONS

\[
\begin{bmatrix}
-\frac{1}{h^2} & \frac{1}{h^2} \\
\frac{1}{h^2} & -\frac{1}{h^2}
\end{bmatrix}
\begin{bmatrix}
x_{i-1} \\
x_i
\end{bmatrix}
= \begin{bmatrix}
0 \\
0
\end{bmatrix}
\bigoplus
\begin{bmatrix}
-\frac{1}{h^2} & \frac{1}{h^2} \\
\frac{1}{h^2} & -\frac{1}{h^2}
\end{bmatrix}
\begin{bmatrix}
x_i \\
x_{i+1}
\end{bmatrix}
= \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

Exemplary merging of two internal contributions

Graph grammar production expressing the merging process
Expression of the solver execution by graph grammar productions

(A1)-(A)^4-(AN) (generation of frontal matrices at leaves of the elimination trees)

(A2)^3 (merging contributions at father nodes)
After merging of the two internal contributions, \([x_{i-1}, x_i]\) and \([x_i, x_{i+1}]\) the i-th equation is fully assembled, and can be eliminated

\[
\begin{bmatrix}
  -\frac{1}{h^2} & \frac{1}{h^2} \\
  \frac{1}{h^2} & -\frac{1}{h^2}
\end{bmatrix}
\begin{bmatrix}
  x_{i-1} \\
  x_i
\end{bmatrix}
= \begin{bmatrix}
  0 \\
  0
\end{bmatrix}
\bigoplus
\begin{bmatrix}
  -\frac{1}{h^2} & \frac{1}{h^2} \\
  \frac{1}{h^2} & -\frac{1}{h^2}
\end{bmatrix}
\begin{bmatrix}
  x_i \\
  x_{i+1}
\end{bmatrix}
= \begin{bmatrix}
  0 \\
  0
\end{bmatrix}
\]

Graph grammar production expressing the elimination process

Expression of the solver execution by graph grammar productions

(A1)-(A)\(^4\)-(AN) (generation of frontal matrices at leaves of the elimination trees)

(A2)\(^3\) (merging contributions at father nodes)

(E2)\(^3\) (elimination of fully assembled nodes)
Finally, we reach the root of the elimination tree

At the root node, all three equations are fully assembled, and the local system can be solved now
Expression of the solver execution by graph grammar productions
(A1)-(A)^4-(AN) (generation of frontal matrices at leaves of the elimination trees)
(A2)^3 (merging contributions at father nodes)
(E2)^3 (elimination of fully assembled nodes)
(A2) – (E2) (merging at parent node followed by elimination)
(Aroot) – (Eroot) (merging at root node followed by full forward elimination)
At the last stage of the solver execution, we execute partial backward substitutions
TRACE THEORY BASED SCHEDULER

Alphabet:
\[ A=\{(A_1), (A_1), (A_2), (A_3), (A_4), (AN), (A_2)_1, (A_2)_2, (A_2)_3, (E_2)_1, (E_2)_2, (E_2)_3, (A_2)_4, (E_2)_4, (A_{\text{root}}), (E_{\text{root}}), (B_{\text{S}})_1, (B_{\text{S}})_2, (B_{\text{S}})_3, (B_{\text{S}})_4 \} \]

Dependency relation for the solver algorithm
\[
\begin{align*}
(A_1), (A_1) & \to (A_2)_1 \\
(A_2), (A_3) & \to (A_2)_2 \\
(A_4), (AN) & \to (A_2)_3 \\
(A_2)_1 & \to (E_2)_1 \\
(A_2)_2 & \to (E_2)_2 \\
(A_2)_3 & \to (E_2)_3 \\
(E_2)_1, (E_2)_2 & \to (A_2)_4 \\
(A_2)_4 & \to (E_2)_4 \\
(E_2)_3, (E_2)_4 & \to (A_{\text{root}}) \\
(A_{\text{root}}) & \to (E_{\text{root}}) \\
(E_{\text{root}}) & \to (B_{\text{S}})_1, (B_{\text{S}})_2 \\
(B_{\text{S}})_1 & \to (B_{\text{S}})_3, (B_{\text{S}})_4 \\
\end{align*}
\]
TRACE THEORY BASED SCHEDULER

Dependency graph
TRAC theory based scheduler

(A1)-(A)1-(A)2-(A)3-(A)4-(AN)-(A2)1-(A2)2-(A2)3-(E2)1-(E2)2-(E2)3-(A2)4-(E2)4-(Aroot)-(Eroot)-(BS)1-(BS)2-(BS)3-(BS)4

Foata Normal Form

\[
\left[ a_1^1 a_2^1 \ldots a_l^1 \right] \left[ a_1^2 a_2^2 \ldots a_l^2 \right] \ldots \left[ a_1^n a_2^n \ldots a_l^n \right]
\]

\( a_i^k \in A \) (alphabet)

\( \forall k \, \forall i, j \in \{1, \ldots, l_k\} \quad a_i^k Ia_j^k \)

\( \forall k \, \forall i \in \{1, \ldots, l_k\} \, \exists j \in \{1, \ldots, l_{k-1}\} \quad a_i^{k-1} Da_j^k \)

Scheduling according to Foata Normal Form:

\[
[(A1)(A)_1(A)2(A)3(A)4(AN)][(A2)_1(A2)_2(A2)_3][(E2)_1(E2)_2(E2)_3] [(A2)_4][(E2)_4]

[(Eroot)][(Aroot)][(Eroot)][(BS)_1(BS)_2][(BS)_3(BS)_4]

Thus, the execution of the solver consists of several steps, where independent tasks are executed in concurrent, interchanged with the synchronization barriers.
NUMERICAL EXPERIMENTS

NVIDIA GeForce 8800 gt with 16 multiprocessors, each having 8 cores (128 cores total)

1D solver $O(\log N)$

2D solver $O(N \log N)$

When the number of leaves $n$ is larger than number of processors, the execution time must be multiplied by $n/p$
PAPERS

Paweł Obrok, Paweł Pierzchała, Arkadiusz Szymczak, Maciej Paszyński
GRAPH GRAMMAR BASED MULTI-THREAD MULTI-FRONTAL PARALLEL SOLVER WITH THRACE THEORY BASED SCHEDULER