Combining Parallel and Distributed Search in Automated Equational Deduction

Carsten Sinz, Jörg Denzinger, Jürgen Avenhaus and Wolfgang Küchlin

PPAM 2001
Outline of Talk

- Unfailing completion procedure
- Parallelization and distribution schemes
- Fine-grained parallelization: PaReDuX
- Distributed cooperating agents: TEAMWORK
- Combination of both approaches: TEAMWORK-PaReDuX
- Experimental results
- Conclusions
Unfailing Completion

- Calculus for equational reasoning by Bachmair, Dershowitz and Plaisted (1989)
- Basic objects: equations $s \leftrightarrow t$ and rewrite rules $l \rightarrow r$
- Basic inference rule: replace equals by equals

Example:  \[ \mathcal{E} = \{ f(x, y) \leftrightarrow f(y, x) \} \]
\[ \mathcal{R} = \{ f(x, n) \rightarrow x, f(x, i(x)) \rightarrow n \} \]
\[ f(f(i(x), n), x) \rightarrow_{\mathcal{R}} f(i(x), x) \leftrightarrow_{\mathcal{E}} f(x, i(x)) \rightarrow_{\mathcal{R}} n \]

- Proof of $\mathcal{T} \models a \leftrightarrow b$ by converting $\mathcal{T}$ into two sets $\mathcal{E}, \mathcal{R}$ by which $a$ and $b$ can be reduced to a common term $c$, i.e.

\[ a \rightarrow_{\mathcal{R} \cup \mathcal{E}}^{*} c \leftarrow_{\mathcal{R} \cup \mathcal{E}}^{*} b \]
Unfailing Completion: Deduction Rules

Orient \[ \frac{(P \cup \{s \leftrightarrow t\}; \mathcal{E}; \mathcal{R})}{(P; \mathcal{E}; \mathcal{R} \cup \{s \rightarrow t\})} \] if \( s \succ t \)

Unfail \[ \frac{(P \cup \{s \leftrightarrow t\}; \mathcal{E}; \mathcal{R})}{(P; \mathcal{E} \cup \{s \leftrightarrow t\}; \mathcal{R})} \] if \( s \nless t, \ t \nless s \)

Collapse\_\mathcal{E} \[ \frac{(P; \mathcal{E} \cup \{s \leftrightarrow t\}; \mathcal{R})}{(P \cup \{u \leftrightarrow t\}; \mathcal{E}; \mathcal{R})} \] if \( s \rightarrow^\mathcal{D}_{\{u \rightarrow r\}} u, \ l \rightarrow r \in \mathcal{R} \cup \mathcal{E}_\succ \)

Collapse\_\mathcal{R} \[ \frac{(P; \mathcal{E}; \mathcal{R} \cup \{s \rightarrow t\})}{(P \cup \{u \leftrightarrow t\}; \mathcal{E}; \mathcal{R})} \] if \( s \rightarrow^\mathcal{D}_{\{u \rightarrow r\}} u, \ l \rightarrow r \in \mathcal{R} \cup \mathcal{E}_\succ \)

Compose \[ \frac{(P; \mathcal{E}; \mathcal{R} \cup \{s \rightarrow t\})}{(P; \mathcal{E}; \mathcal{R} \cup \{s \rightarrow u\})} \] if \( t \rightarrow_{\mathcal{R} \cup \mathcal{E}_\succ} u \)

Simplify \[ \frac{(P \cup \{s \leftrightarrow t\}; \mathcal{E}; \mathcal{R})}{(P \cup \{s \leftrightarrow u\}; \mathcal{E}; \mathcal{R})} \] if \( t \rightarrow_{\mathcal{R} \cup \mathcal{E}_\succ} u \)

Delete \[ \frac{(P \cup \{s \leftrightarrow s\}; \mathcal{E}; \mathcal{R})}{(P; \mathcal{E}; \mathcal{R})} \]

Deduce \[ \frac{(P; \mathcal{E}; \mathcal{R})}{(P \cup \{s \leftrightarrow t\}; \mathcal{E}; \mathcal{R})} \] if \( s \leftrightarrow t \in \text{CP}_\succ(\mathcal{R} \cup \mathcal{E}) \)
Parallelization and Distribution Schemes

- High degree of non-determinism in deduction rules, complexity of problem $\rightarrow$ Heuristics
- Parallelization applicable on different levels:
  1. Individual deduction steps in parallel (fine-grained)
  2. (Large) groups of deduction steps in parallel (medium-grained)
  3. Independent or communicating calculi with different heuristics (coarse-grained)
- Parallelization scheme has to match hardware architecture:
  - (Symmetric) multi-processor computers (SMPs)
  - Clusters of workstations
Sequential Algorithm and PaReDuX

• Sequential algorithm (Huet, 1981):
  1. Select next equation, add it to \( R \) or \( E \) (Orient, Unfail).
  2. Perform simplifications (Collapse, Compose, Simplify, Delete).
  3. Derive new consequences (Deduce).
  4. Simplify goal and goto 1. until proof is found.

• PaReDuX: Parallel execution of deduction rules in steps 2 and 3:
  – Concurrently: Collapse&Compose, Simplify&Delete&Deduce
  – Instances of Simplify&Deduce executed in parallel for each equation resp. equation pair
Cooperating Agents: Teamwork

- Equation selection heuristic determines performance, but no universally best one → Use multiple competing heuristics
- Experts run instances of sequential algorithm with different heuristics.
- Team meetings allow exchange of positive/negative information.
- Referees evaluate success of each expert.
Cooperation in Teamwork

1. Compute Assessment
2. Send Short Reports
3. Choose New Supervisor
4. Send Full Reports
5. Distribute New Problem Description
# Comparison PaReDuX vs. TEAMWORK

<table>
<thead>
<tr>
<th></th>
<th>PaReDuX</th>
<th>TEAMWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallelization</td>
<td>fine-grained</td>
<td>coarse-grained</td>
</tr>
<tr>
<td>Equation selection heuristics</td>
<td>single, fixed</td>
<td>multiple, dynamic</td>
</tr>
<tr>
<td>Information exchange</td>
<td>all equations</td>
<td>“best” equations</td>
</tr>
<tr>
<td>Strategy-compliant?</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Timing dependencies</td>
<td>none</td>
<td>considerable</td>
</tr>
<tr>
<td>Max. (theoretical) speed-ups</td>
<td>linear</td>
<td>super-linear</td>
</tr>
<tr>
<td>Communication</td>
<td>shared memory</td>
<td>network</td>
</tr>
<tr>
<td>Suitable HW platform</td>
<td>SMPs</td>
<td>clusters</td>
</tr>
</tbody>
</table>
**Integration:** TEAMWORK-PaReDuX

- TEAMWORK method with each expert running adapted PaReDuX algorithm

- Software architecture:

  ![TEAMWORK-PaReDuX Diagram]

  - PaReDuX and TEAMWORK approaches counteractive or complementary? → Empirical judgement
### Experimental Results

<table>
<thead>
<tr>
<th>Problem</th>
<th>Runtimes</th>
<th></th>
<th></th>
<th>Speed-up</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BOO002-2</td>
<td>411.95</td>
<td>154.02</td>
<td>194.80</td>
<td>76.19</td>
<td>2.67</td>
<td>2.11</td>
</tr>
<tr>
<td>BOO007-4</td>
<td>-</td>
<td>-</td>
<td>554.31</td>
<td>163.68</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BOO022-1</td>
<td>148.50</td>
<td>56.34</td>
<td>226.44</td>
<td>57.08</td>
<td>2.64</td>
<td>0.66</td>
</tr>
<tr>
<td>COL003-12</td>
<td>364.89</td>
<td>111.21</td>
<td>262.08</td>
<td>86.56</td>
<td>3.28</td>
<td>1.39</td>
</tr>
<tr>
<td>GRP002-4</td>
<td>205.93</td>
<td>83.36</td>
<td>14.00</td>
<td>6.22</td>
<td>2.47</td>
<td>14.71</td>
</tr>
<tr>
<td>GRP119-1</td>
<td>867.70</td>
<td>258.45</td>
<td>423.86</td>
<td>99.81</td>
<td>3.36</td>
<td>2.07</td>
</tr>
<tr>
<td>GRP122-1</td>
<td>752.73</td>
<td>218.73</td>
<td>155.31</td>
<td>49.23</td>
<td>3.32</td>
<td>4.85</td>
</tr>
<tr>
<td>GRP175-3</td>
<td>1422.15</td>
<td>555.19</td>
<td>63.70</td>
<td>29.04</td>
<td>2.56</td>
<td>22.33</td>
</tr>
<tr>
<td>GRP175-4</td>
<td>464.33</td>
<td>163.02</td>
<td>243.50</td>
<td>87.61</td>
<td>2.85</td>
<td>1.91</td>
</tr>
<tr>
<td>LUKA3</td>
<td>-</td>
<td>-</td>
<td>135.73</td>
<td>33.17</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RA007</td>
<td>-</td>
<td>-</td>
<td>67.22</td>
<td>25.58</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ROB004-1</td>
<td>1683.82</td>
<td>670.16</td>
<td>40.73</td>
<td>14.60</td>
<td>2.51</td>
<td>41.34</td>
</tr>
</tbody>
</table>

2 Sun ES450, each with 4 UltraSparcIi processors @400MHz, 1GB
Conclusions

1. **TEAMWORK-PaReDuX** integrates two different parallelization/distribution schemes.

2. **TEAMWORK-PaReDuX** reflects two-tired HW architecture of clusters of SMP computers.

3. Speed-up factors almost multiply compared to individual approaches.

4. Proposed combination presumably applicable to other search problems (e.g. Gröbner bases, resolution-based theorem provers).