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.

$$a \ \longrightarrow^*_{\mathcal{R}\cup\mathcal{E}_\succ} \ c \ \longleftarrow^*_{\mathcal{R}\cup\mathcal{E}_\succ} \ b$$

Unfailing Completion: Deduction Rules

Orient	$\dfrac{(\mathcal{P}\cup\{s\leftrightarrow t\};\, \mathcal{E};\, \mathcal{R})}{(\mathcal{P};\, \mathcal{E};\, \mathcal{R}\cup\{s\rightarrow t\})}$	if $s \succ t$
Unfail	$\dfrac{(\mathcal{P}\cup\{s\leftrightarrow t\};\, \mathcal{E};\, \mathcal{R})}{(\mathcal{P};\, \mathcal{E}\cup\{s\leftrightarrow t\};\, \mathcal{R})}$	if $s \not\succ t$, $t \not\succ s$
Collapse$_{\mathcal{E}}$	$\dfrac{(\mathcal{P};\, \mathcal{E}\cup\{s\leftrightarrow t\};\, \mathcal{R})}{(\mathcal{P}\cup\{u\leftrightarrow t\};\, \mathcal{E};\, \mathcal{R})}$	if $s \rightarrow^{\triangleright}_{\{l\rightarrow r\}} u$, $l \rightarrow r \in \mathcal{R} \cup \mathcal{E}_{\succ}$
Collapse$_{\mathcal{R}}$	$\dfrac{(\mathcal{P};\, \mathcal{E};\, \mathcal{R}\cup\{s\rightarrow t\})}{(\mathcal{P}\cup\{u\leftrightarrow t\};\, \mathcal{E};\, \mathcal{R})}$	if $s \rightarrow^{\triangleright}_{\{l\rightarrow r\}} u$, $l \rightarrow r \in \mathcal{R} \cup \mathcal{E}_{\succ}$
Compose	$\dfrac{(\mathcal{P};\, \mathcal{E};\, \mathcal{R}\cup\{s\rightarrow t\})}{(\mathcal{P};\, \mathcal{E};\, \mathcal{R}\cup\{s\rightarrow u\})}$	if $t \rightarrow_{\mathcal{R}\cup\mathcal{E}_{\succ}} u$
Simplify	$\dfrac{(\mathcal{P}\cup\{s\leftrightarrow t\};\, \mathcal{E};\, \mathcal{R})}{(\mathcal{P}\cup\{s\leftrightarrow u\};\, \mathcal{E};\, \mathcal{R})}$	if $t \rightarrow_{\mathcal{R}\cup\mathcal{E}_{\succ}} u$
Delete	$\dfrac{(\mathcal{P}\cup\{s\leftrightarrow s\};\, \mathcal{E};\, \mathcal{R})}{(\mathcal{P};\, \mathcal{E};\, \mathcal{R})}$	
Deduce	$\dfrac{(\mathcal{P};\, \mathcal{E};\, \mathcal{R})}{(\mathcal{P}\cup\{s\leftrightarrow t\};\, \mathcal{E};\, \mathcal{R})}$	if $s \leftrightarrow t \in \mathrm{CP}_{\succ}(\mathcal{R} \cup \mathcal{E})$

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

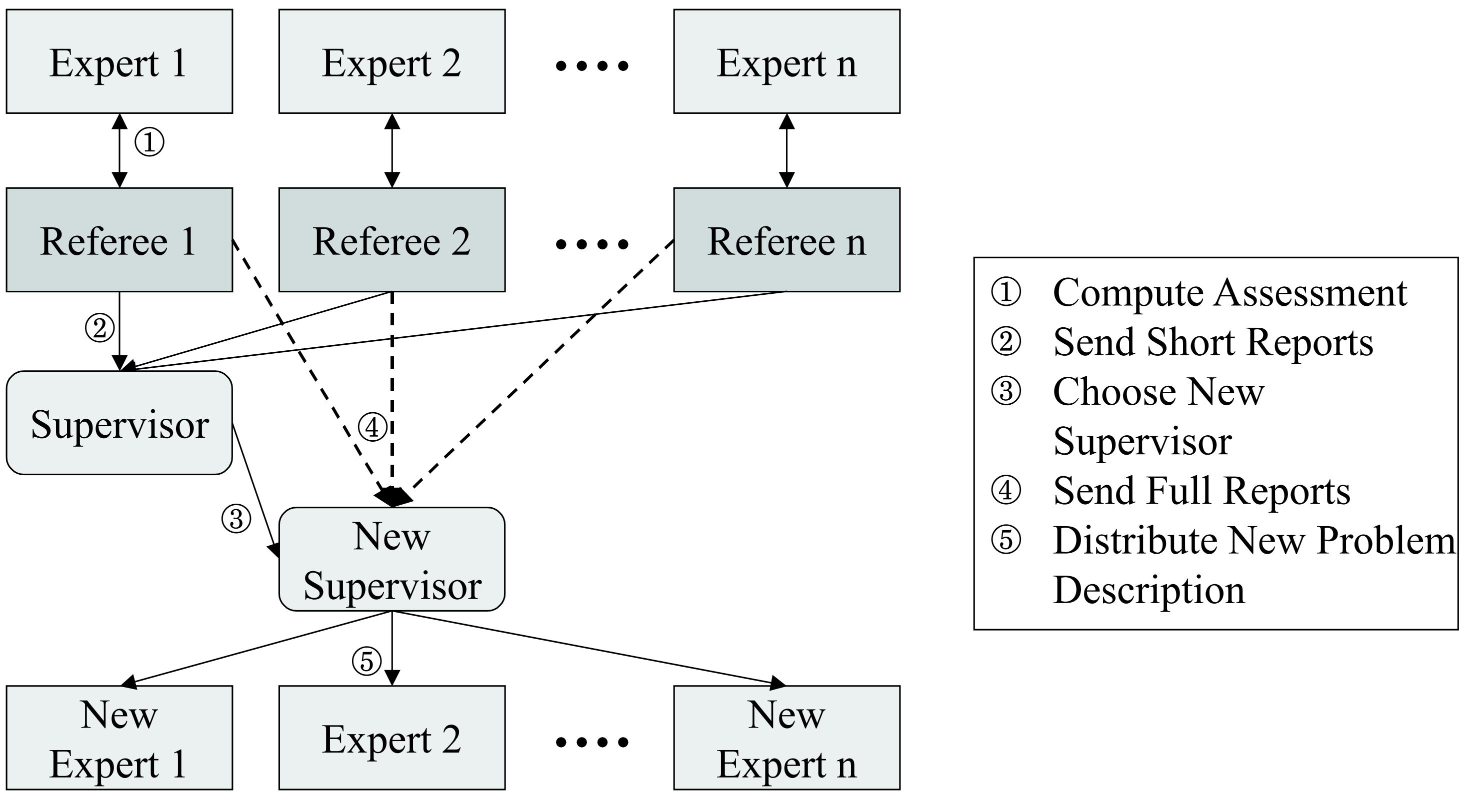
Sequential Algorithm and PaReDuX

- Sequential algorithm (Huet, 1981):
  1. Select next equation, add it to $\mathcal{R}$ or $\mathcal{E}$ (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 $\rightarrow$ 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

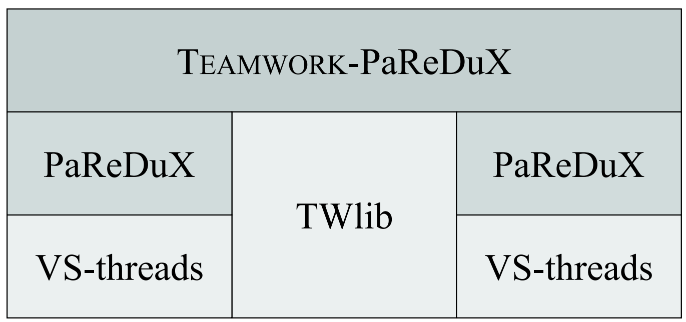


Comparison PaReDuX vs. Teamwork

	PaReDuX	Teamwork
Parallelization	fine-grained	coarse-grained
Equation selection heuristics	single, fixed	multiple, dynamic
Information exchange	all equations	“best” equations
Strategy-compliant?	yes	no
Timing dependencies	none	considerable
Max. (theoretical) speed-ups	linear	super-linear
Communication	shared memory	network
Suitable HW platform	SMPs	clusters

Integration: Teamwork-PaReDuX

- Teamwork method with each expert running adapted PaReDuX algorithm
- Software architecture:



- PaReDuX and Teamwork approaches counteractive or complementary? $\rightarrow$ Empirical judgement

Experimental Results

Problem	Runtimes				Speed-up			
	Seq.	Par.	Dist.	P.&D.	Par.	Dist.	P.×D.	P.&D.
BOO002-2	411.95	154.02	194.80	76.19	2.67	2.11	5.63	5.41
BOO007-4	-	-	554.31	163.68	-	-	-	-
BOO022-1	148.50	56.34	226.44	57.08	2.64	0.66	1.74	2.60
COL003-12	364.89	111.21	262.08	86.56	3.28	1.39	4.60	4.21
GRP002-4	205.93	83.36	14.00	6.22	2.47	14.71	36.33	33.11
GRP119-1	867.70	258.45	423.86	99.81	3.36	2.07	6.96	8.69
GRP122-1	752.73	218.73	155.31	49.23	3.32	4.85	16.10	15.29
GRP175-3	1422.15	555.19	63.70	29.04	2.56	22.33	57.17	48.97
GRP175-4	464.33	163.02	243.50	87.61	2.85	1.91	5.44	5.30
LUKA3	-	-	135.73	33.17	-	-	-	-
RA007	-	-	67.22	25.58	-	-	-	-
ROB004-1	1683.82	670.16	40.73	14.60	2.51	41.34	103.76	115.33

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).