Distributed Theorem Proving by Peers

Maria Paola Bonacina
Dept. of CS
University of Iowa

William Mc Cune
MCS Div.
Argonne Nat. Lab.

Overview of Peers

- Logic: equational
  (also AC-theories).

- Strategies: refutational,
  contraction-based
  (simplification-based,
  rewriting-based,
  completion-based).

- Parallelization: Clause-Diffusion
  methodology.

- Environment: network of
  workstations.

Overview of Peers

Clause - Diffusion:

- coarse - grain parallelism:
  parallel search,
- concurrent, asynchronous, deductive processes ("peers"):
  - same theorem proving problem,
  - same strategy (default)
    (different strategies possible)
    no special - purpose processes,
  - separate data bases
    (distributed memory),
  - independent derivations,
  - one succeeds, all halt.

Overview of Peers

Clause - Diffusion:

- subdivision of work:
  - subdivision of clauses (distributed allocation algorithm),
  - subdivision of inferences (data-driven)

  - ⊛ subdivision of the search space,
  - ⊛ balance of work-load.

- communication:

  exchange of clauses

  by message-passing.

The structure of Peers

- Initialization phase
- Distributed computation:

  3 states : - incoming messages,
  - inferences,
  - idle.

  Basic work-loop:

  - select pending message
    messages ⇒ actions,
  - else do unit of inference work,
  - else become idle
    (till a message arrives).

The structure of Peers

- Termination :
  - finds a proof ⇒ success
    broadcast "halt message."
  - raise an exception
    broadcast "halt message"
    (fault-tolerance possible).
  - all idle,
    all sent messages received,
    (e.g. for Knuth-Bendix completion
    or no refutation)

Dijkstra-Pnueli global termination detection algorithm.

The "Types of work"

- Receive new settler
- Receive inference message
- Do inference work:
  - select given-clause or given-pair
  - broadcast inference message
  - generate new clauses
  - forward contraction
  - allocation
  - backward contraction
  - send new settlers

More features

- Selection of strategies:
  flags / parameters.
- Distributed allocation algorithm:
  - "rotate",
  - "syntax",
  - "select-min".
- Treatment of new clauses
  generated by
  backward contraction.

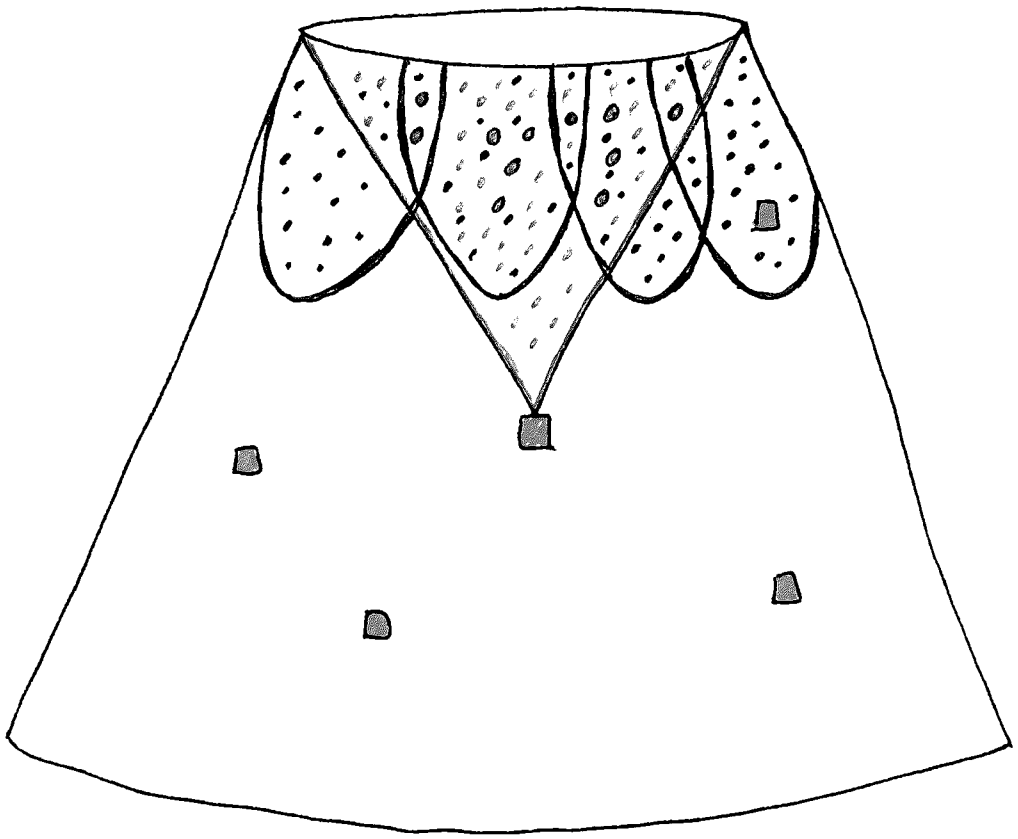
Experiments

Problem	*1-Peers*	*2-Peers*	*4-Peers*	*6-Peers*	*8-Peers*
x3	96.45	50.29	43.28	30.66	7.51
r2	40.04	16.51	18.74	34.97	22.31
sa1	15.99	7.30	16.06	12.96	9.65
sa2	24.28	20.09	12.76	81.05	20.34

Problem	*1-Peers*	*2-Peers*	*4-Peers*	*6-Peers*	*8-Peers*
x3-a	96.28	53.58	46.87	54.04	25.95
x3-b	96.45	50.29	43.28	30.66	7.51
x3-c	96.06	51.37	44.06	43.52	28.06
x3-d	95.86	49.16	44.52	31.65	8.60
x3-e	96.36	87.64	38.34	24.93	31.02

Problem	*1-Peers*	*2-Peers*	*4-Peers*	*6-Peers*	*8-Peers*
kbcomm	5.14	1.62	0.55	0.55	0.58
x3	62.42	49.40	24.09	23.80	14.03
sa1	25.84	10.05	20.32	4.60	5.29
sa2	12.97	20.73	2.24	1.56	2.15

Discussion on the experiments



■ : solutions

blue: sequential search

green / black: parallel search by Clause - Diffusion