Distributed On-the-Fly Equivalence Checking

Christophe Joubert and Radu Mateescu

INRIA / VASY

http://www.inrialpes.fr/vasy

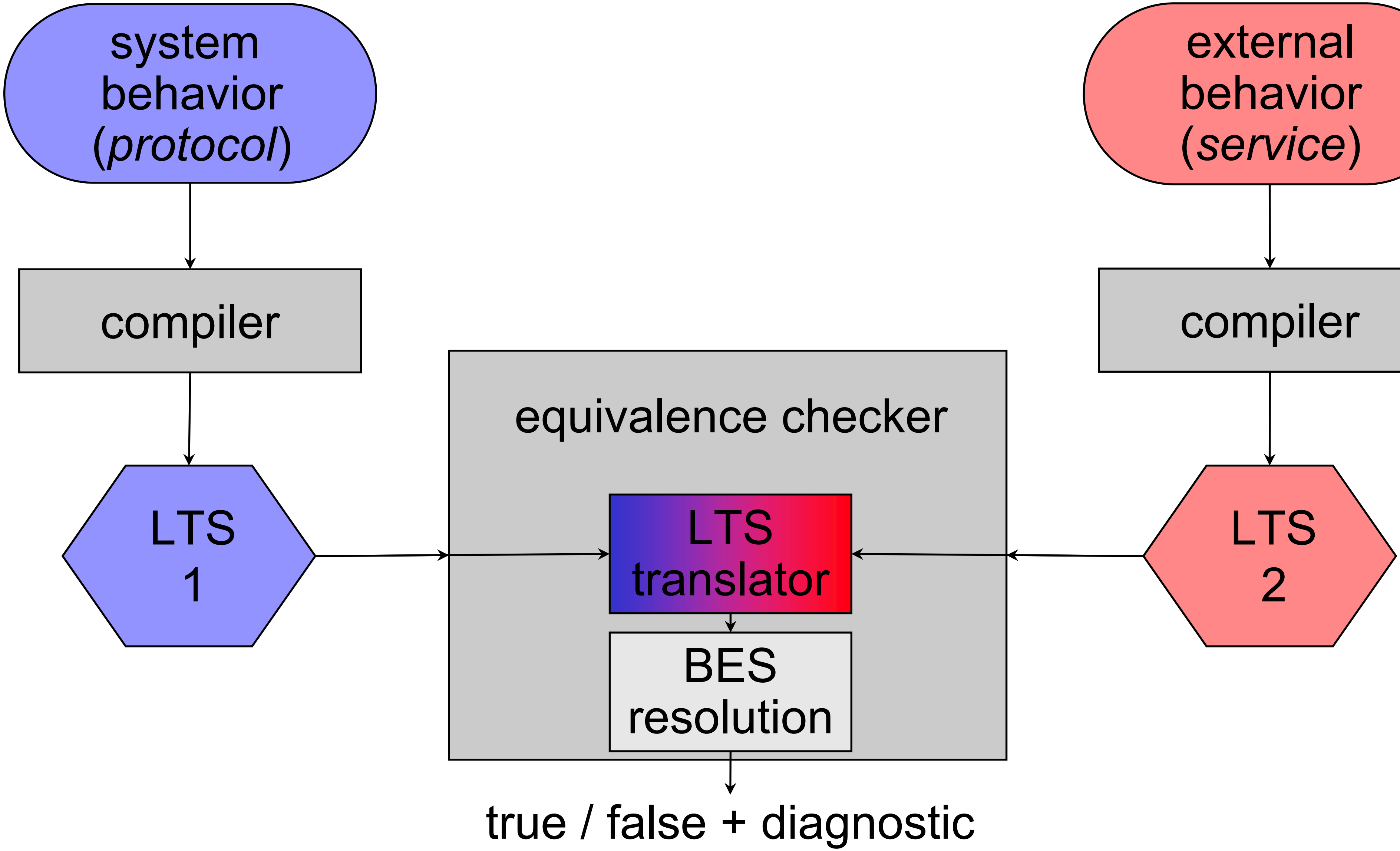
INRIA RHÔNE-ALPES

© 2004 Christophe Joubert

Outline

- Equivalence checking as a Boolean Equation System (BES) resolution problem
- BES distributed resolution algorithm
- Implementation and experiments
- Conclusion and future work

Equivalence checking using BES resolution



Equivalence relations in terms of BES

- $LTS_1 = (Q_1, A, T_1, q_{01})$, $LTS_2 = (Q_2, A, T_2, q_{02})$

  strong equivalence $\approx\ \subseteq Q_1 \times Q_2$ is the max relation s.t. $p \approx q$ iff

$$\boxed{\begin{array}{l} \forall a \in A, \forall p \xrightarrow{a} p' \in T_1, \exists q \xrightarrow{a} q' \in T_2, p' \approx q' \\ \wedge \\ \forall a \in A, \forall q \xrightarrow{a} q' \in T_2, \exists p \xrightarrow{a} p' \in T_1, p' \approx q' \end{array}}$$

- $LTS_1 \approx LTS_2$ iff $q_{01} \approx q_{02}$
- Principle: $p \approx q$ iff $X_{p,q}$ is **true**
- Translation in BES:

$$\boxed{X_{p,q} \;=_\nu\; \left(\bigwedge_{p \xrightarrow{b} p'} \bigvee_{q \xrightarrow{b} q'} X_{p',q'}\right) \wedge \left(\bigwedge_{q \xrightarrow{b} q'} \bigvee_{p \xrightarrow{b} p'} X_{p',q'}\right)}$$

© 2004 Christophe Joubert

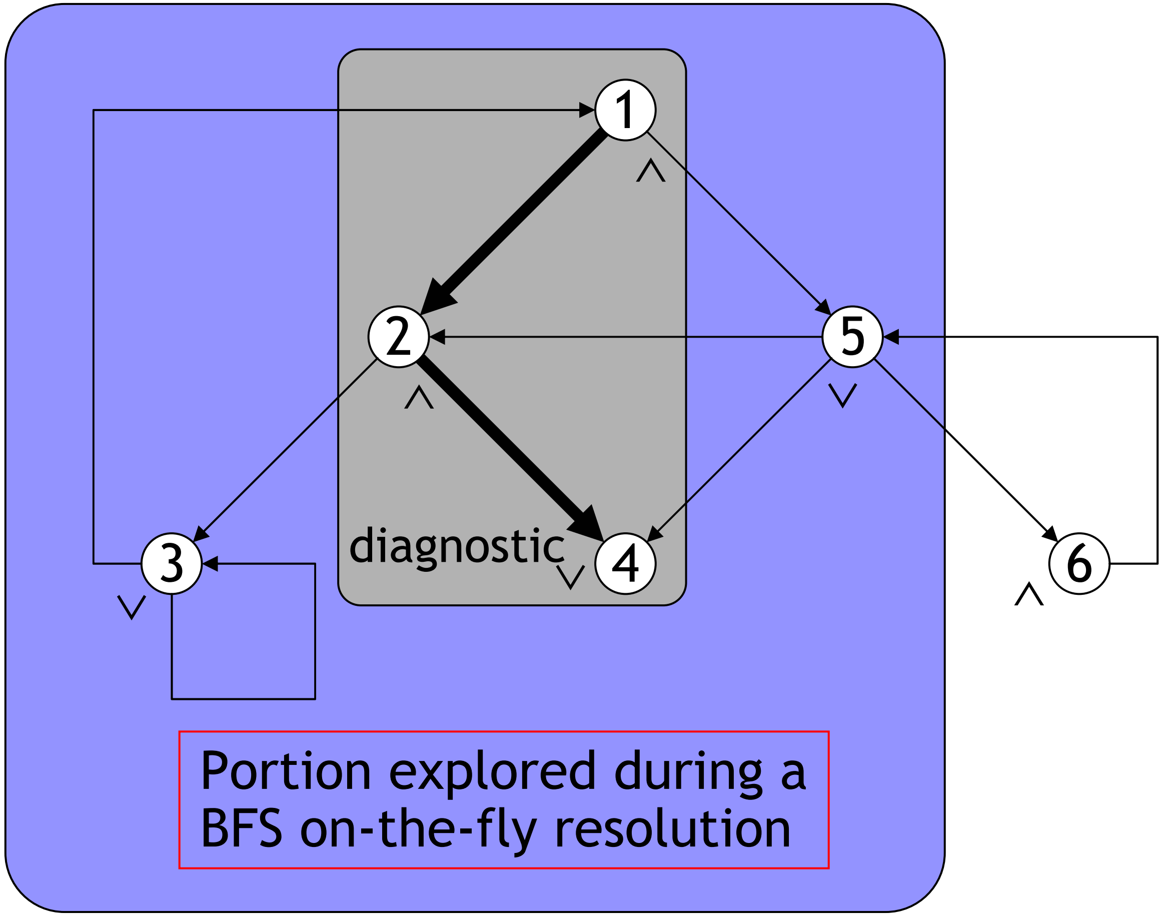
BES / boolean graph sequential resolution

BES

$$\begin{cases} x_1 =_\nu x_2 \wedge x_5 \\ x_2 =_\nu x_3 \wedge x_4 \\ x_3 =_\nu x_1 \vee x_3 \\ x_4 =_\nu F \\ x_5 =_\nu x_2 \vee x_4 \vee x_6 \\ x_6 =_\nu x_5 \end{cases}$$

- Theory of boolean graphs [Andersen-Vergauwen-95][Vergauwen-Lewi-94]
- CAESAR_SOLVE library [Mateescu-03]

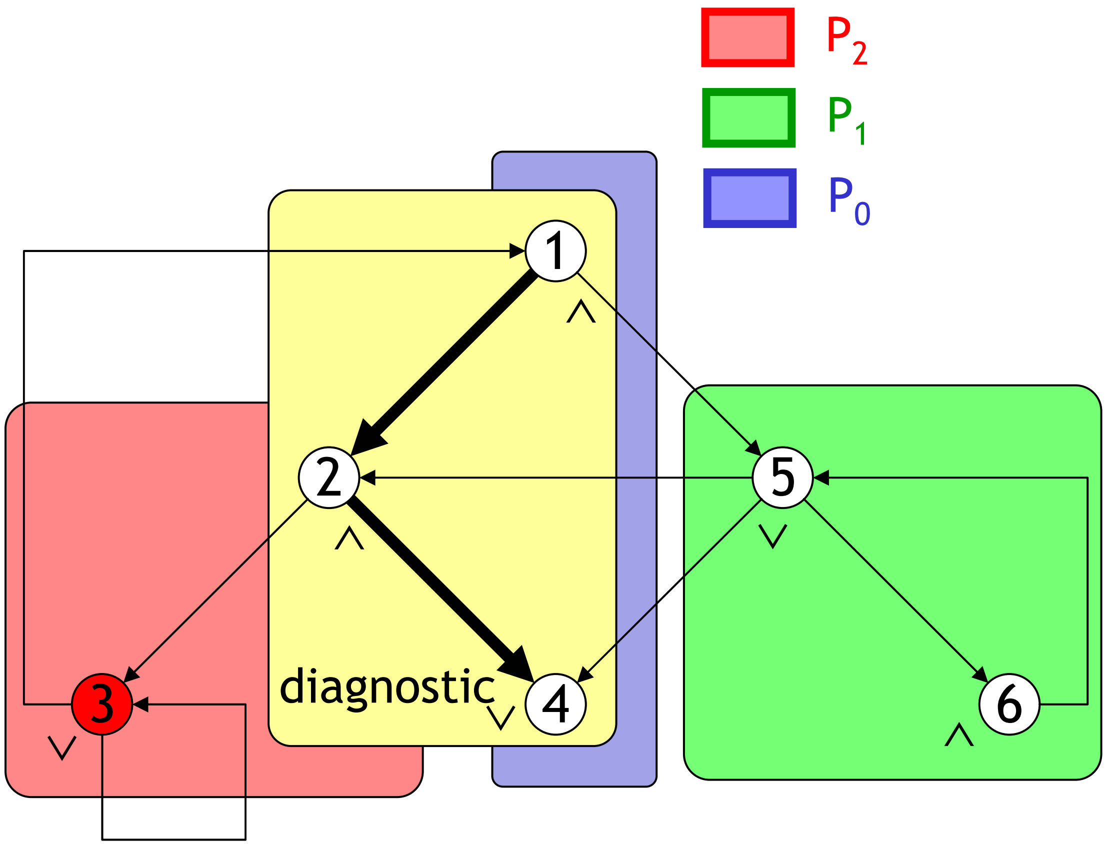
boolean graph



BES / boolean graph distributed resolution

- Limitations of sequential resolution:
  - Memory (BES with more than 10^8 variables to solve)
  - Time (traversals of very large BES)

- Reasons for distribution:
  - Regular problem prone to balanced distribution of task and data
  - Running faster with few memory used per machine



Distributed BFS on-the-fly resolution on 3 processes

Distributed model

- Parallel architecture:
  - Distributed computers with own CPU and memory
  - NOW and cluster of PC
- Network:
  - Strongly connected topology
  - Loosely coupled
  - FIFO channels
- SPMD (Single Program Multiple Data) programming model:
  - Several processes performing the distributed BES resolution
  - 1 *coordinator* process (configuration, launching, collection of statistical data, termination detection)

© 2004 Christophe Joubert

DSOLVE algorithm

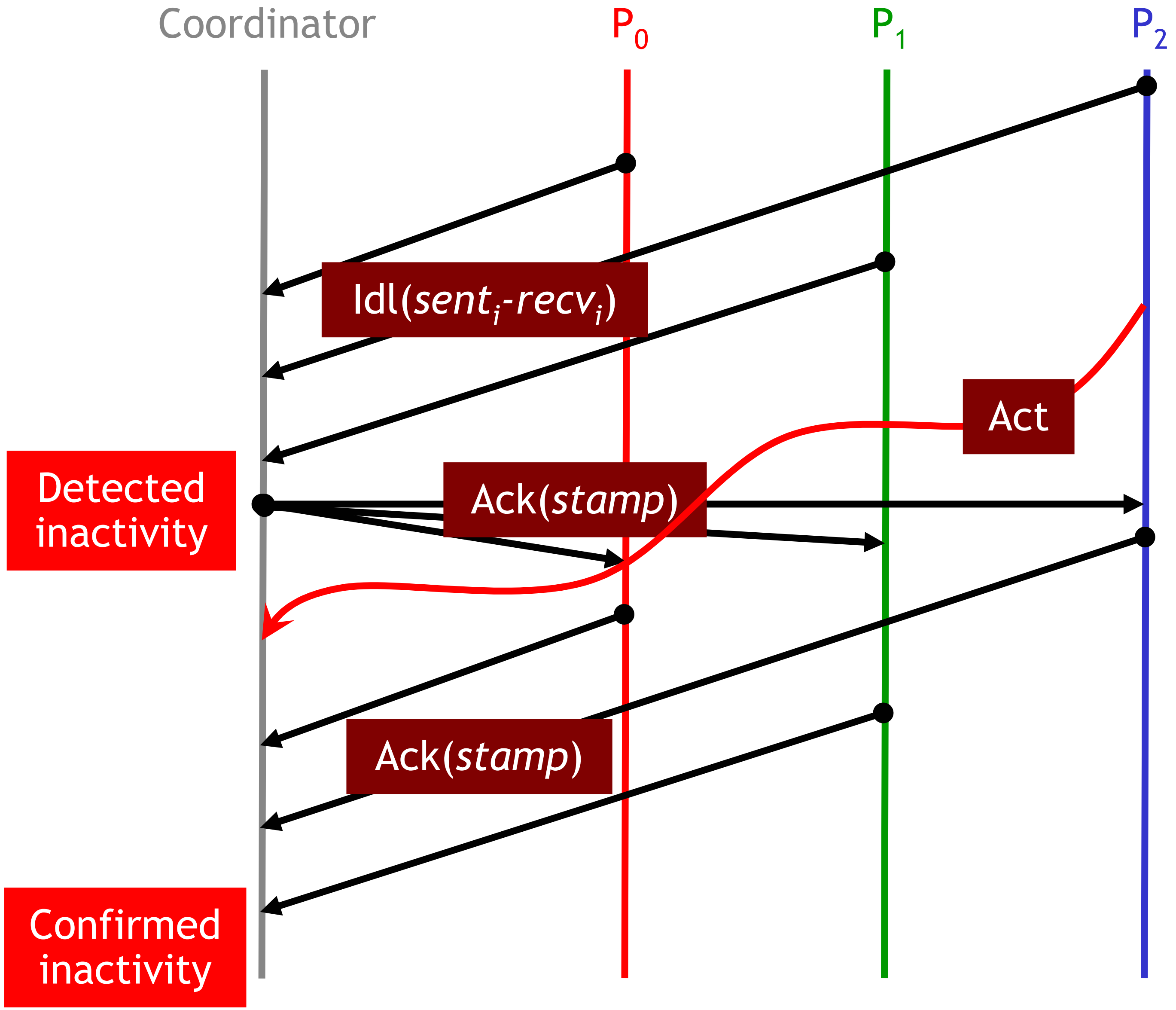
- Each process solves a subset of boolean variables determined by a static hash function
- Inputs:
  - Variable of interest *x*
  - Implicit boolean graph *(V,E,L)* (successor function)
  - Static hash function *h* (data partitioning)
  - Index of current process *i* ($i \in [0, P-1]$)
- Steps:
  - BFS forward exploration of boolean graph (V,E,L) starting at $x \in V$
  - Backward propagation of stable (computed) variables
  - Distribution (communication) of variables to be solved or stabilized
  - Termination when *x* is stable or the entire boolean graph has been explored
- Outputs:
  - Boolean value of *x*
  - Diagnostic by keeping relevant successors

Communication primitives

- Problems:
  - Reducing memory consumption
  - Maximizing communication and computation overlapping
  - Avoiding busy waiting on emission failures
  - Preventing communication deadlocks

- Solution:
  - Asynchronous (overlapping of communication with computations)
  - Both blocking and non-blocking communication (avoiding synchronization and busy waiting)
  - Fine tuned loosely coupled distributed communication library (CAESAR_NETWORK)
    - UNIX sockets with bounded buffers
    - TCP/IP protocol

Termination detection



Conditions of termination:

- Stabilized variable of interest *x*

or

- Boolean graph completely explored =
  - all local working sets of variables empty
  - No more messages transiting through the network

$\left(\sum_{i=0}^{P}(sent_i - recv_i) = 0\right)$

=> 2 broadcast waves of global inactivity detection between the coordinator and the resolution processes

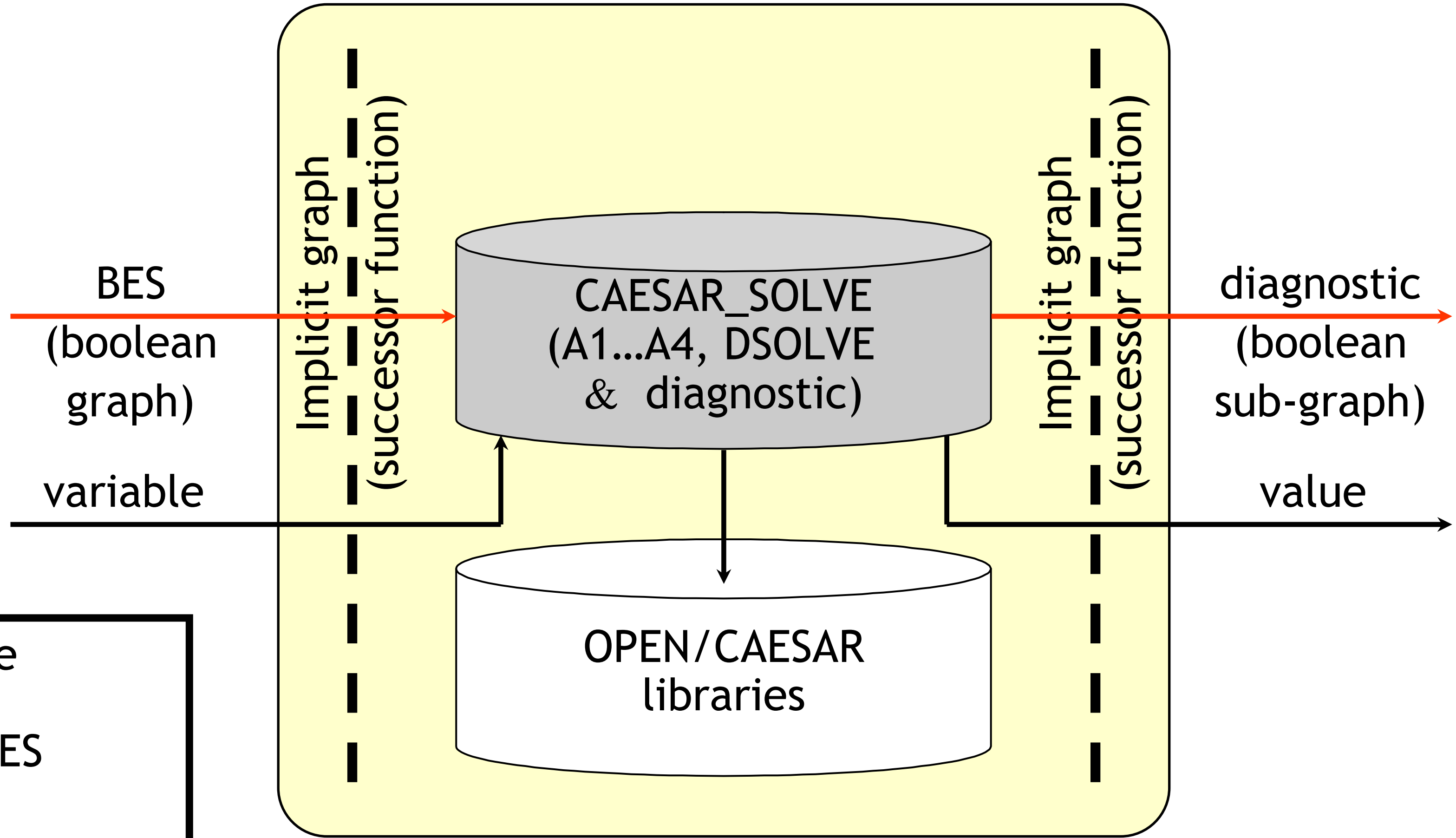
Complexity of the distributed resolution

For a boolean graph (V,E,L) and P running processes:

- Worst case time complexity = O (|V|+|E|)
  - 2 intertwined graph traversals (forward and backward)
- Worst case memory complexity = O (|V|+|E|)
  - Dependencies stored during graph exploration
- Worst case message complexity = O (2.|E|.(P-1)/P)
  - 2 messages (expansion and stabilization) exchanged by edges
- Distributed termination detection = O (|E|)
  - Practically, only 0.01% of total exchanged messages used for termination detection (processes rarely inactive)

© 2004 Christophe Joubert

DSOLVE implementation

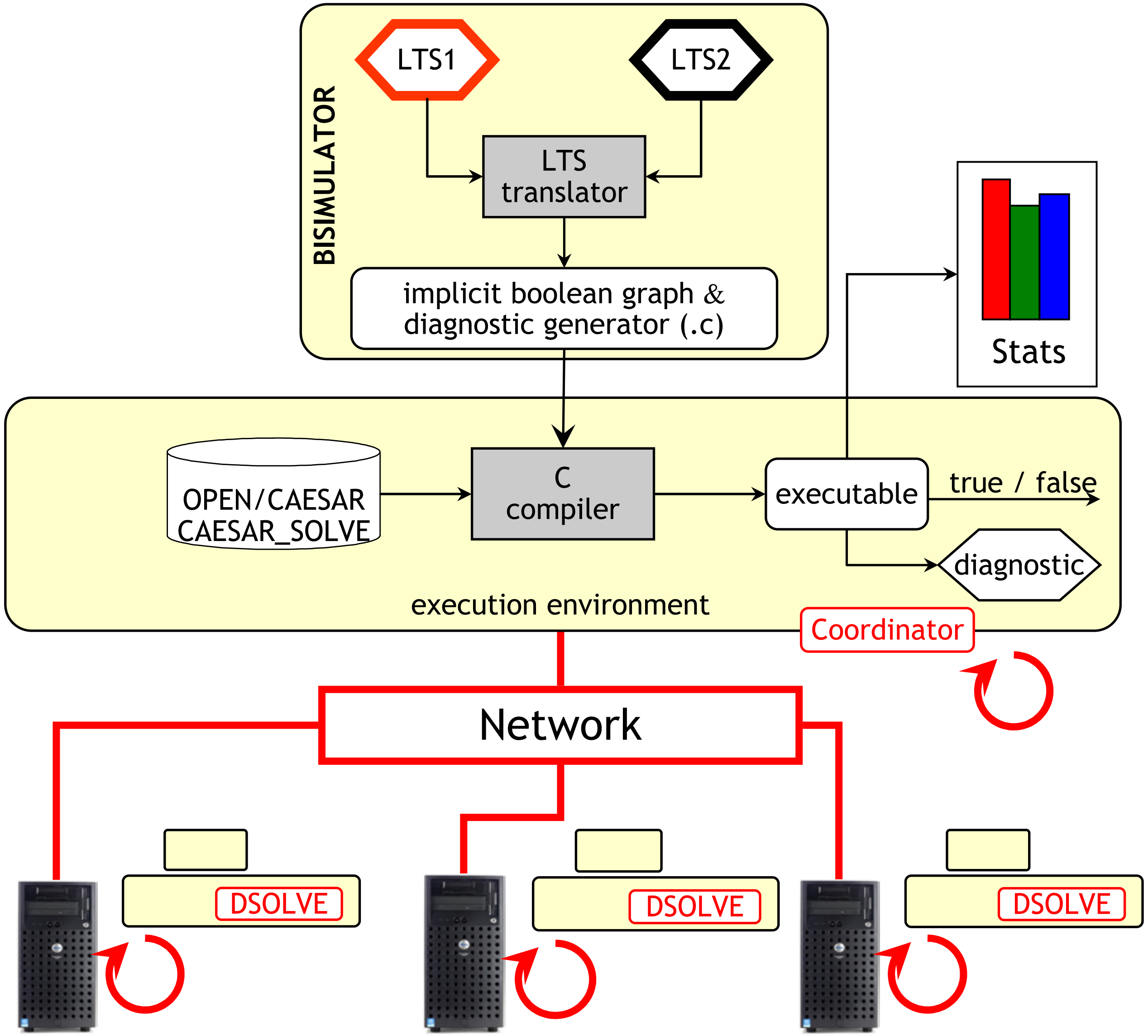


- 8500 lines of C code
- Integrated to the BES resolution library CAESAR_SOLVE
- Developed using the OPEN/CAESAR environment

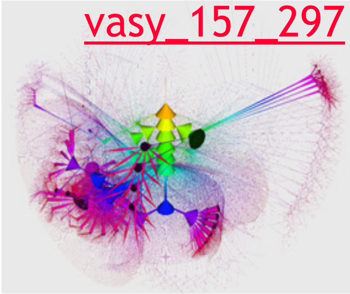
© 2004 Christophe Joubert

Distributed equivalence checking using DSOLVE

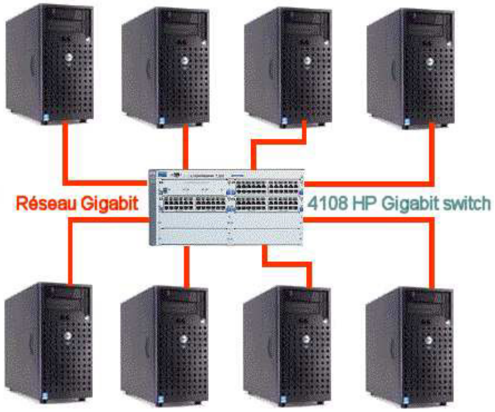
- Front-end (BISIMULATOR):
  - Called sequentially and independently on each worker
  - Encodes the equivalence relations as BES
  - Performs transitive closures on tau-transitions for weak equivalences
- Back-end:
  - BES resolution
  - CAESAR_SOLVE
    - A1, …, A4 (sequential)
    - DSOLVE (parallel)



Performance measures



- Benchmark:
  - 65 LTS taken from CADP demos and VLTS suite
  - From (9.10^3 states, 2.10^4 transitions) to (8.10^6 states, 4.10^7 transitions)
  - http://www.inrialpes.fr/vasy/cadp

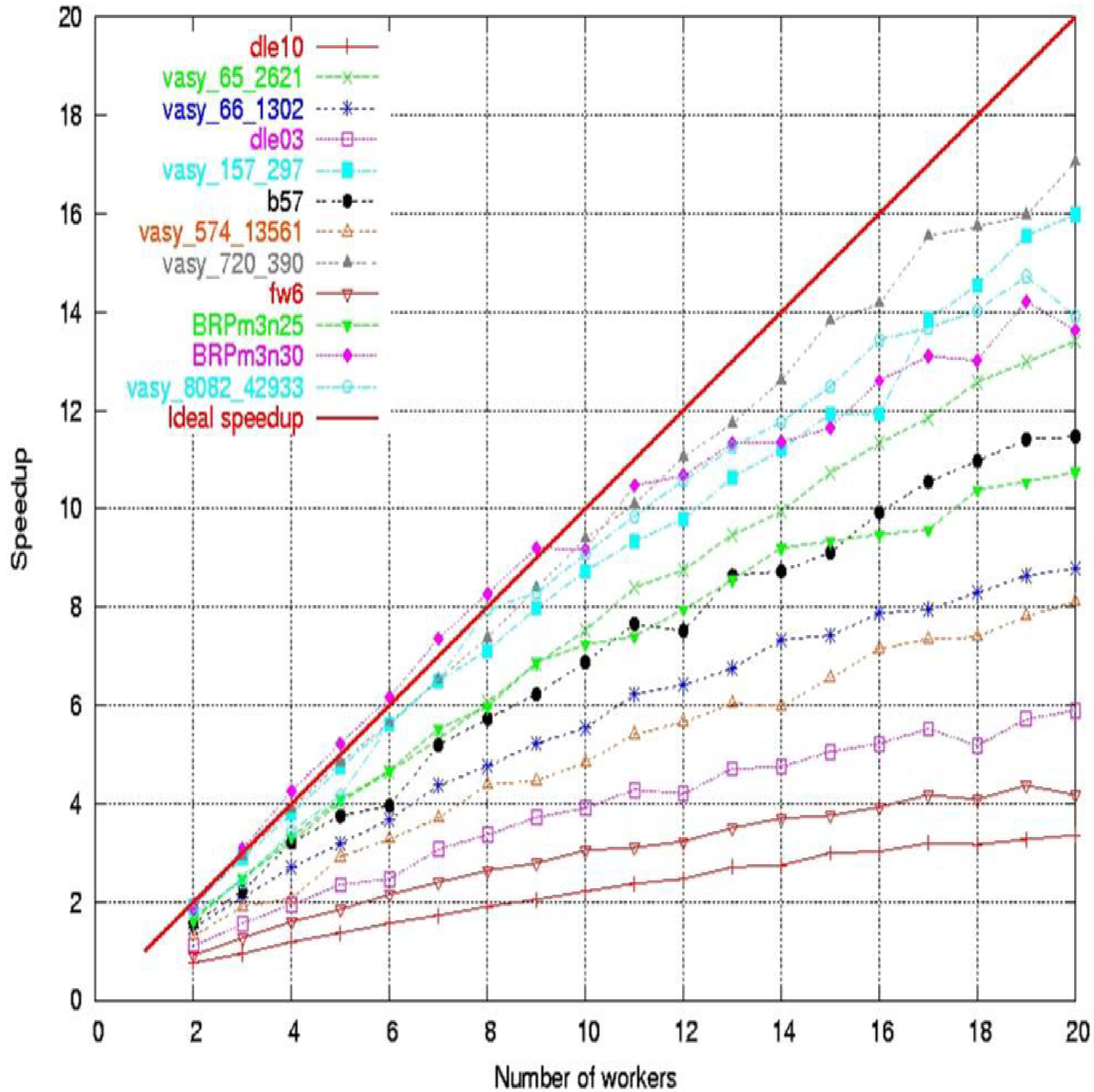


- Parallel architecture:
  - 20 * Xeon 2.4 GHz + 1.5 GB of RAM + 80 GB + Gigabit network
  - Debian 2.4.26
  - OAR batch scheduler

- Experimentation:
  - Distributed BISIMULATOR (using DSOLVE) vs. sequential BISIMULATOR (using BFS resolution algorithm) for different equivalences
  - Worst-case: comparison of an LTS with its minimized version
  - Exclusion of system-dependent fixed costs (code loading, LTS copying, connection initialization)

Speedup – strong equivalence

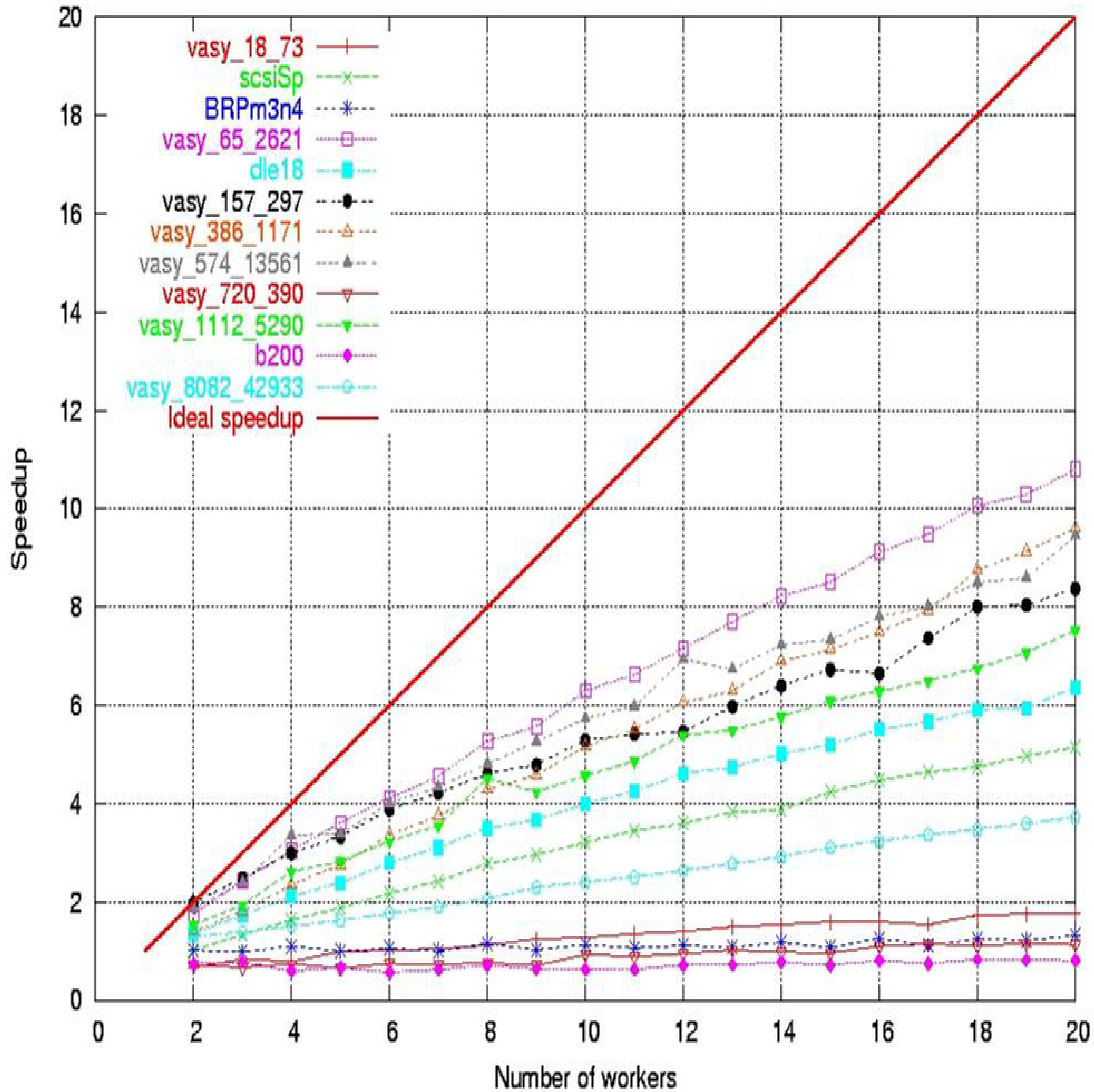
- Best behavior among all equivalences (very few time spent in the front-end)
- Linear speedups
- BRPm3n30:
  - 332.53 s. in seq
  - 29.06 s. with 13 processors (speedup of 11.5)



Speedup – observational equivalence

- Large BES encoding
- Better for LTS with few tau-transitions or deterministic behavior
- Vasy_65_2621:

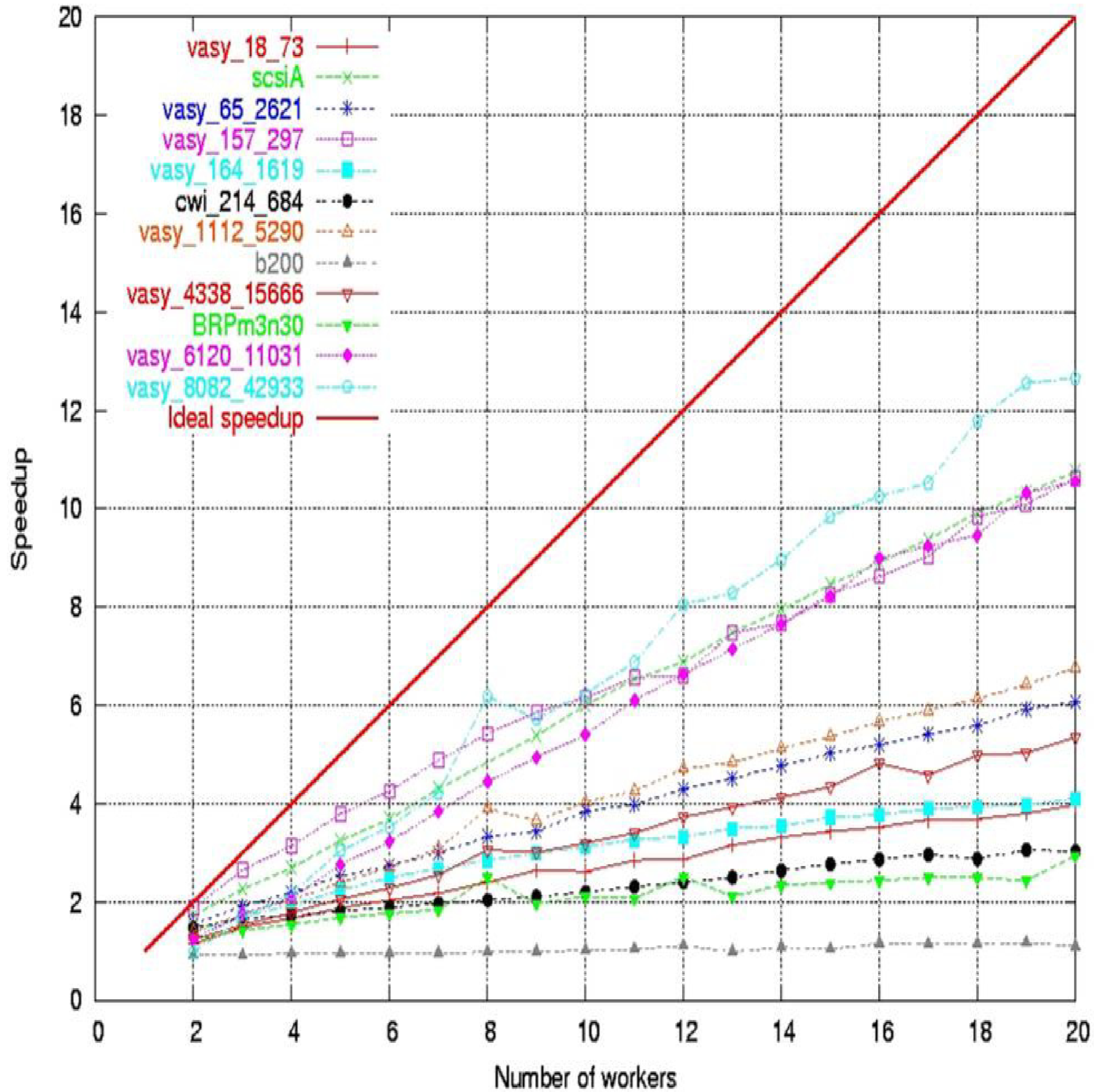
  Speedup of 7.86 with 13 processors
- Branching equivalence gives similar results



Speedup – tau*.a equivalence

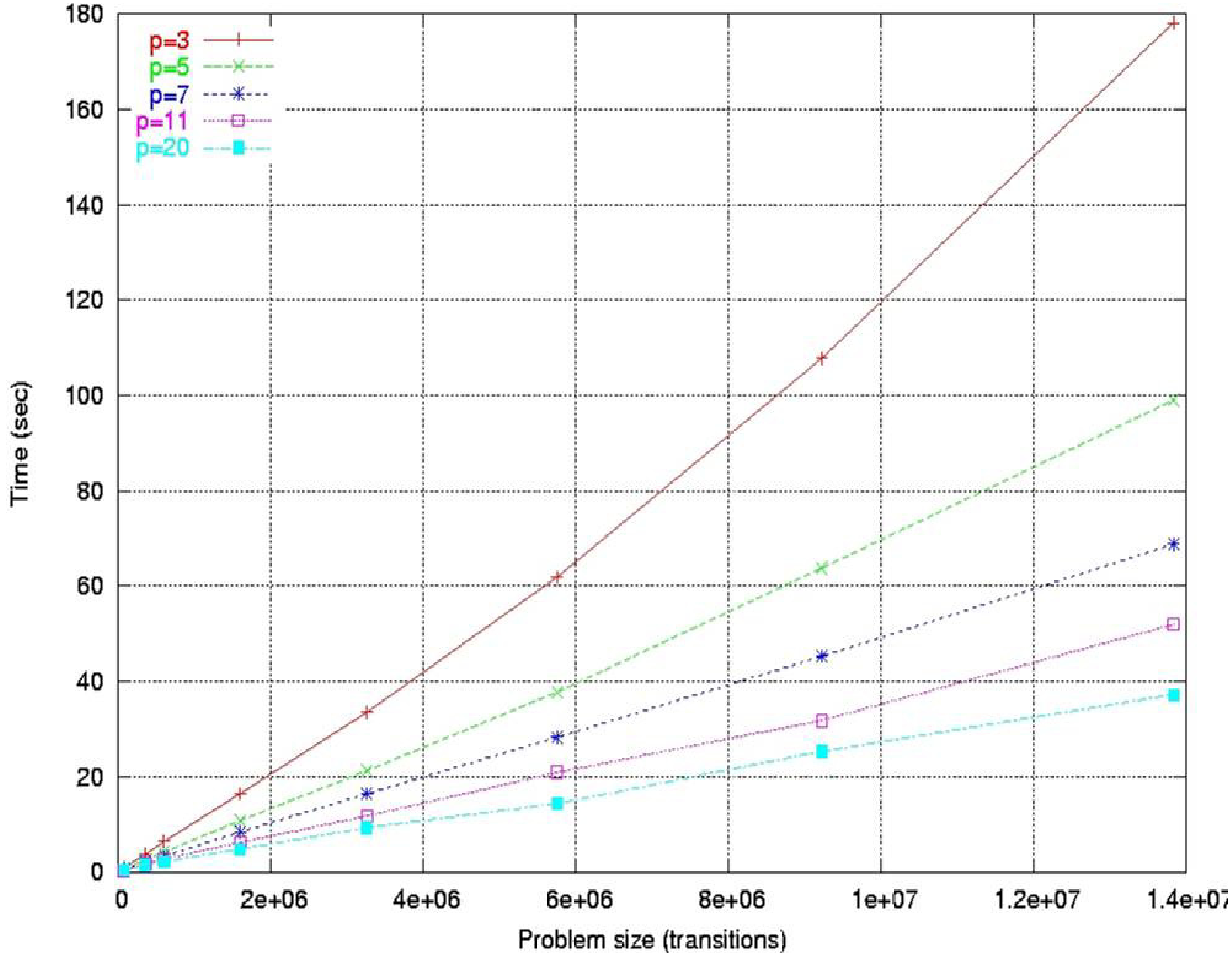
- Worst behavior (extensive transitive closures on tau transitions in the front-end on each worker)
- Very small BES encoding for high % of tau transitions
- Vasy_8082_42933:

  Speedup of 8.22 with 13 processors
- Similar results for safety equivalence
- 3 factors:
  - Size of LTSs
  - % of Tau transitions
  - Degree of non-determinism



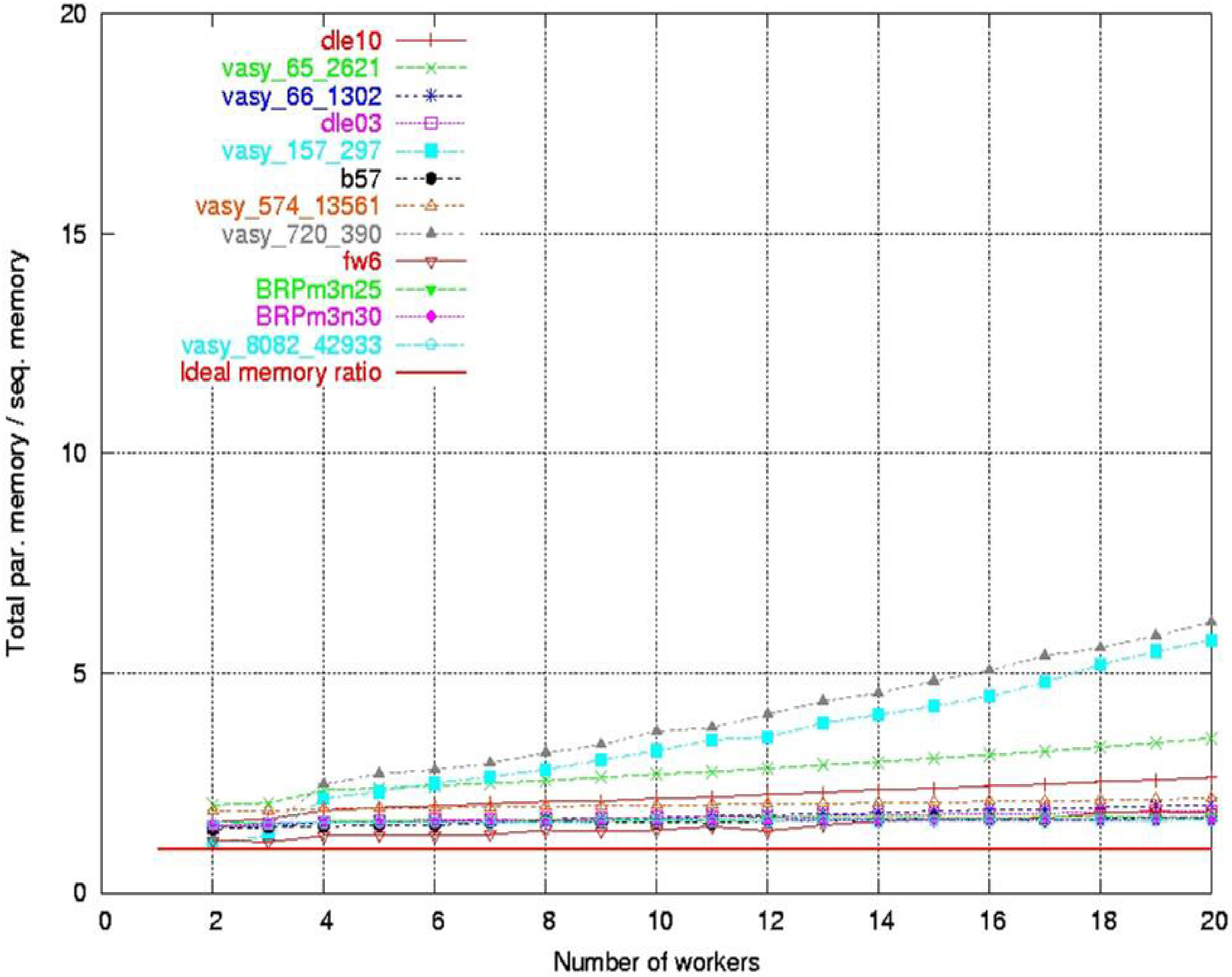
Scalability - problem size / time

- BRP with packet length K ∈ [4,35]:
  - Strong equivalence
  - Fixed p number of processors (p ∈ [3,20])
  - Adapted to increases in problem size



Scalability – memory / # processes

- Main bottleneck for verification problems
  = main motivation for distribution
- Memory equally divided amongst processes
  (hash tables and communication buffers)
- Small increases due to distributed resolution exploring more boolean variables (edges) than its sequential counterpart (bigger hash table)



© 2004 Christophe Joubert

Conclusion

- DSOLVE, a distributed algorithm for local resolution of BES
- A distributed version of BISIMULATOR and a distributed generation of diagnostic for equivalence checking
- Generic implementation running on widely-used loosely-coupled parallel machines (clusters and NOW)
- Extensive set of experiments performed on large BES (VLTS benchmark suite)
  - Linear speedups (even superlinear for large BES with particular forms)
  - Scalability w.r.t. BES size and number of processors

© 2004 Christophe Joubert

Future work

- Verification:
  - Tau-confluence reduction [Pace-Lang-Mateescu-03]
  - Alternation-free mu-calculus model-checking [Mateescu-03]
  - Markovian bisimulation [Hermanns-Siegle-99]

- Potential applications:
  - Propositional Horn clauses resolution [Liu-Smolka-98]
  - Abstract interpretation [Chen-94]
  - Data flow analysis [Fecht-Seidl-96]

For more information …

Christophe
Joubert

Radu
Mateescu

http://www.inrialpes.fr/vasy

© 2004 Christophe Joubert