Distributed On-the-Fly Verification of Large State Spaces

Christophe Joubert

INRIA Rhône-Alpes / VASY
http://www.inrialpes.fr/vasy

December 12th, 2005

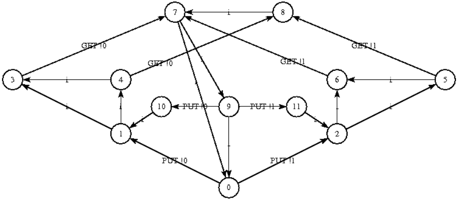
Joint work with Radu Mateescu and Hubert Garavel

Formal Verification

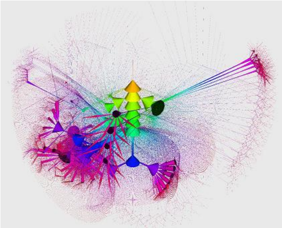
- Goal : to produce reliable softwares
- Technique : using formal models and computation capacities of computers to analyze their behaviour
- Targets : critical computer systems, implying high human or financial costs
- Example : lost of Cryosat satellite - 08/10/05 - software error on Rockot Launcher - 136 M €

Formal Model : Labeled Transition System (LTS)

- Simplified behaviour of a data exchange protocol between 2 computers :

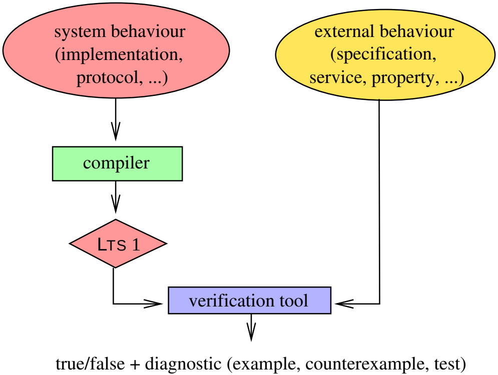


- Real size LTS (10^5 states, 10^5 transitions) extracted from the VLTS benchmark :



- Software support (CADP) for LTS representation :
  - *explicit* (predecessor/successor function) – BCG (Binary Coded Graph)
  - *implicit* (successor function) – OPEN/CÆSAR [Garavel-98]

Enumerative Verification

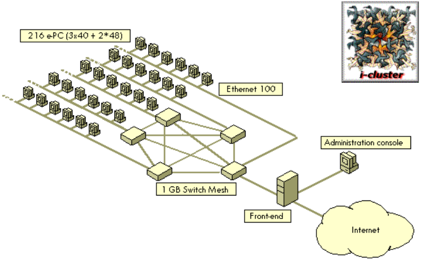


- **Global verification**
  - LTS constructed before verification
- **On-the-fly verification**
  - LTS constructed during verification
  - Possibility of partial exploration of LTS to obtain a result
- **Problem of state space explosion**

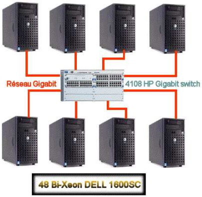
Distributed Verification

- To use the computation power and memory space of interconnected machines to solve complex problems

- ICLUSTER (INRIA/ID)
  216 PIII 733 MHz 256 Mb



- IDPOT
  48 Bi-Xeon 2.5 GHz 1.5 Gb



http://www.grid5000.org/

Grid'5000

Four large problems treated in this research work

Enumerative Verification

- On-the-fly equivalence checking
- On-the-fly minimization (τ-confluence)
- On-the-fly model-checking of temporal logic formulae

Test generation

- On-the-fly generation of conformance test cases

Generic approach to the four large problems

⇒ Resolution of boolean equation systems (BES) with diagnostic

Enumerative Verification

- Equivalence relations
  [Andersen-Vergauwen-95],[Mateescu-03]
- τ-confluence [Pace-Lang-Mateescu-03]
- μ-calculus formulae
  [Andersen-94],[Mateescu-Sighireanu-02]

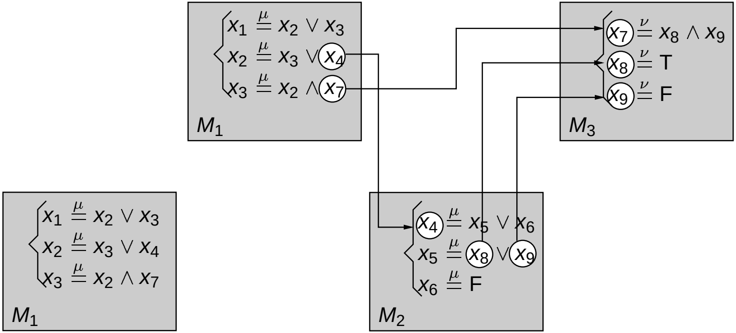
Test generation

- Conformance test cases

Outline

Outline

Monoblock and multiblock BES



- Set of fixed point boolean equations ($M_i = \{x_{ij} \stackrel{\sigma_i}{=} op_{ij} X_{ij}\}_{1 \leq j \leq m_i, 1 \leq i \leq n}$)
- Pure disjonctive or conjonctive formulae (simple BES)
- n blocks M_i ($i \in [1..n]$) with acyclic interblock dependencies

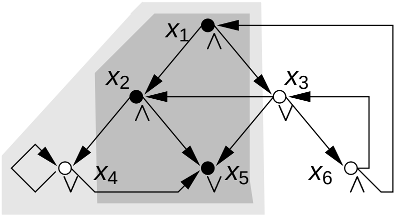
Boolean graph and BES resolution

○ : *true*

● : *false*

: explored portion during an on-the-fly DFS resolution

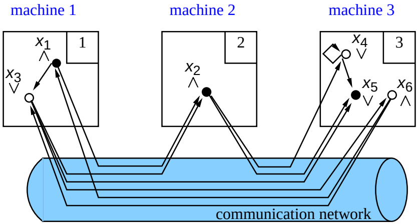
: diagnostic



- **Boolean graph $G = (V, E, L)$ associated to a BES (of sign ν)**
  - V = variables set
  - E = edges set
  - L = variables sign ($\vee$, $\wedge$)
- **Local sequential resolution `[Mateescu-03]`**
  - Truth value of main variable
  - Diagnostic generation (boolean subgraph)

Distribution of BES Resolution

- Goal : to spread memory cost over several machines (current limit 10^7 variables) and to decrease resolution time (with respect to BES size)
- Method : natural and balanced distribution of BES resolution problem by variable assignment on different processes



Outline

1. Boolean Equation Systems

2. Distributed On-the-Fly Resolution of BES
   - Resolution of Monoblock BES
   - Resolution of Multiblock BES
   - Generic Library CAESAR_SOLVE_2

3. Three Applications in Enumerative Verification

4. Application to Test Generation

5. Conclusion and Future Work

Resolution of Monoblock BES

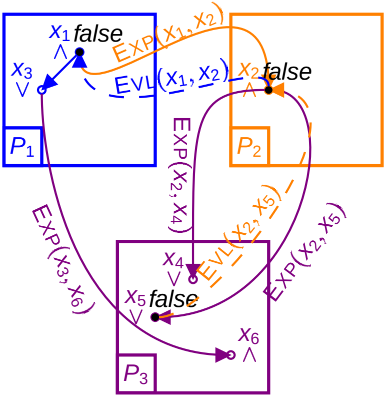
Computation model

- Distributed memory architecture (message passing) : cluster of PCS
- P SPMD processes and 1 supervisor process
- Each process solves a subgraph of boolean variables (static hash function)

Distributed Algorithm : DSOLVE

- Forward exploration of boolean graph (V, E, L) starting from main variable $x \epsilon V$
- Backward propagation of stable variables
- Distribution of variables through remote dependencies
- Termination detection : x stable or completely solved boolean graph

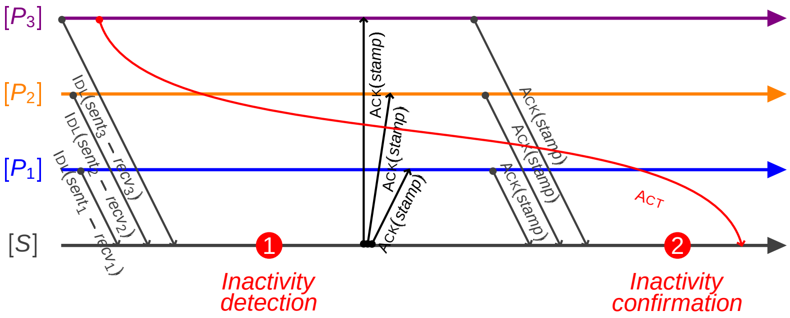
DSOLVE Execution



1. Initialization (main variable x_1)
2. Local expansion and remote expansion (EXP message)
3. Conjonctive variable without successor (i.e., **false** constant)
4. Backward local and remote (EVL message) propagation of stablized (i.e., computed) variables
5. If main variable stabilizes, then resolution terminates

Distributed Termination Detection Algorithm (DTD)

- Two waves of global inactivity detection between supervisor process and resolution processes



Complexity results

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

- Time complexity in the worst case = $O(|V| + |E|)$
  - two intertwined graph traversals (forward and backward)
- Memory complexity in the worst case = $O(|V| + |E|)$
  - dependencies stored during graph exploration
- Complexity in number of messages = $O(|E|)$
  - two messages (expansion and stabilization) at most exchanged per transition
- Distributed termination detection = $O(|E|)$
  - two waves with at most $3P$ messages exchanged per transition

Resolution of Multiblock BES

- Sequential approach [Mateescu-03] :
  - recursive resolution calls per block
  - call stack bounded by the number of blocks
- Naive distributed approach (DSOLVE) :
  - a single resolution for the entire BES
  - termination detection of the entirely solved BES

→ incompatible or inefficient with distributed resolution of multiblock BES

- Adopted solution :
  - distinction between variables of different blocks
  - distributed termination detection per block
  - two traversals (forward and backward) per block

Distributed Resolution of Multiblock BES

- Conservative extension of DSOLVE algorithm $\Longrightarrow$ identical computation model

Distributed algorithm MB-DSOLVE

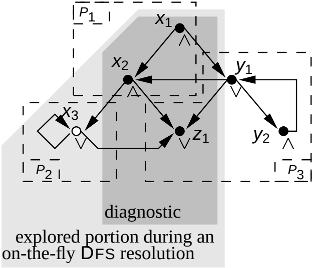
- Choice of block number among those waiting to be explored or stabilized
- Priority to stabilization of blocks with highest level in the dependency graph between blocks
- Limitation of exploration requests : only one block portion explored at a time, and priority to blocks with lowest level
- Management of interblock unstabilized transitions : residual propagations
- Distributed detection of solved block portion

Example of Distributed On-the-Fly Resolution of Multiblock BES

bloc 1
$$\begin{cases} x_1 \overset{\nu}{=} x_2 \wedge y_1 \\ x_2 \overset{\nu}{=} x_3 \wedge z_1 \\ x_3 \overset{\nu}{=} x_3 \vee z_1 \end{cases}$$

bloc 2
$$\begin{cases} y_1 \overset{\mu}{=} x_2 \vee z_1 \vee y_2 \\ y_2 \overset{\mu}{=} y_1 \end{cases}$$

bloc 3
$$\begin{cases} z_1 \overset{\nu}{=} F \end{cases}$$



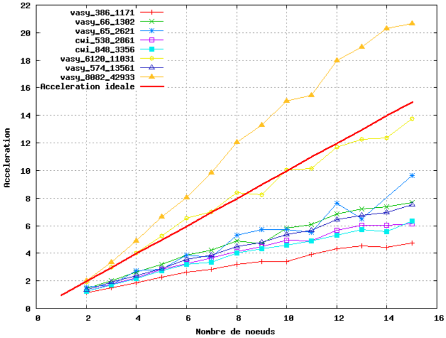
- Fixed point can be different between blocks
- Interblock transition need to be stabilized

Generic Library CAESAR_SOLVE_2

- Distributed on-the-fly resolution of alternation free BES and distributed on-the-fly generation of diagnostics (boolean subgraph)
  - Monoblock BES - DSOLVE (10 000 lines of C code)
  - Multiblock BES - MB-DSOLVE (7 000 complementary lines of C code)
- Tested with a parameterized generator (1000 lines of C code) of random BES
- Connected to a generic and prototype communication library using TCP/IP sockets
- Generic and independant boolean resolution API, given by the library CÆSAR_SOLVE_1 [Mateescu-03]

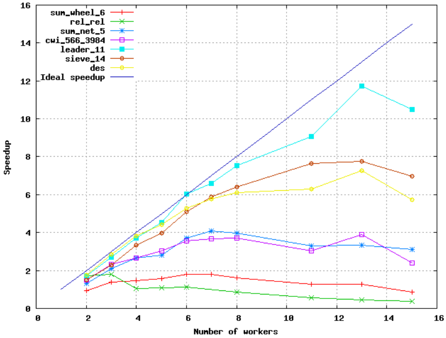
Outline

Distributed vs. Sequential BISIMULATOR



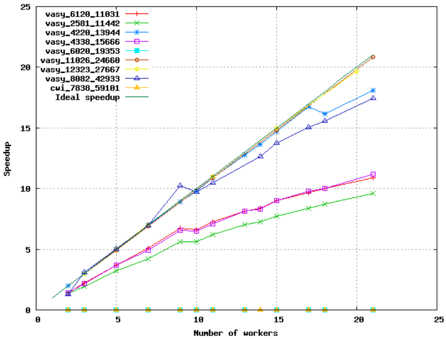
- Strong equivalence : best behaviour among all equivalences (very few time spent in the computation of successors)
- Linear speedups
- vasy_6120_11031 (VLTS) :
  - 169.47 s. in sequential
  - 11.69 s. with 15 processes, speedup of 14.5
- Constant memory overhead (4 times sequential)
  - for all number of computation nodes
  - for a fixed problem size

Distributed vs. Sequential TAU_CONFLUENCE



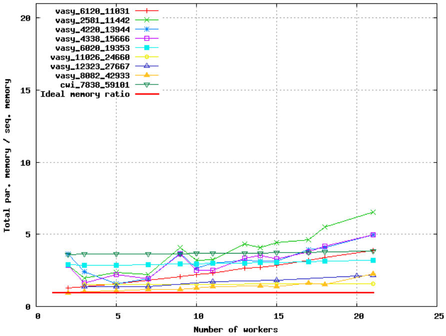
- Speedup close to linear in the number of nodes
- Reduction between one and four orders (similarly in sequential)
- Limitation in few cases :
  - BFS traversal with resolution call for each τ-transition
  - DTD that forces nodes to synchronize often
  - Alternative solution : call over a set of τ-transitions
- Constant memory overhead (3 times sequential)
  - for all number of computation nodes
  - with few dependency to the problem size

Distributed vs. Sequential EVALUATOR 3.5 (time)



- Speedup close to linear
- Comparable in time and memory to UppDMC (distributed model-checker)
- Significative gain in time for the example *vasy_12323_27667* (VLTS) and livelock detection :
  - $>$ 2 days in optimised DFS sequential
  - $<$ 3h in distributed over 20 nodes, speedup of 19.7
- Immediate detection of diagnostics

Distributed vs. sequential EVALUATOR 3.5 (memory)

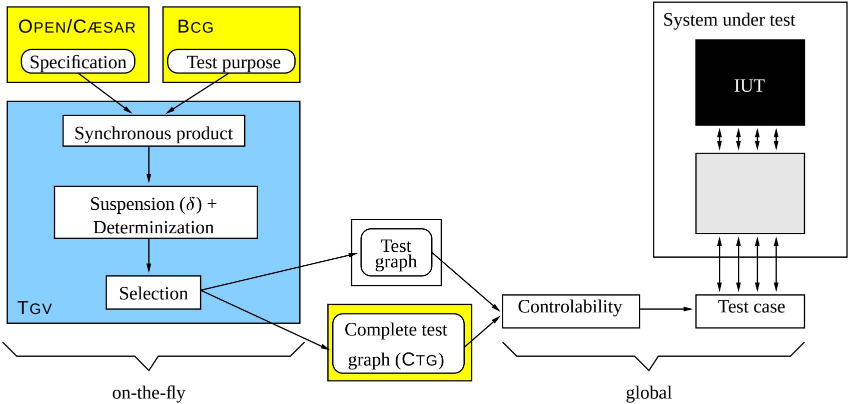


- Constant memory overhead (4 times the one in sequential) :
  - for all number of computation nodes
  - for a formula and its truth value (detection of counterexample or not)
- Distributed model-checker for other temporal logics :
  - ACTL, by encoding in alternation free μ-calculus `[Fantechi-Gnesi-Ristori-92]`

Outline

TGV : On-the-Fly Test Case Generator

- [Fernandez-Jard-Jeron-Viho-96], [Jard-Jeron-05]



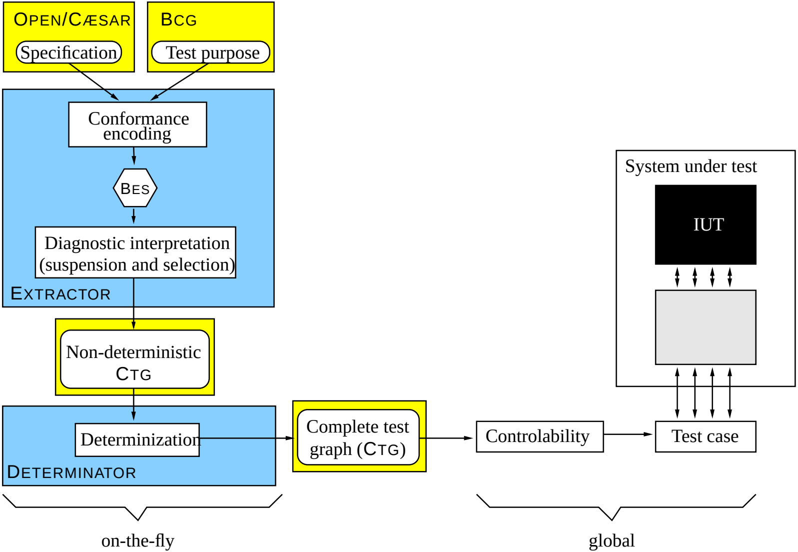
Encoding of Test Cases in terms of BES

- Test generation =
  - particular case of diagnostic generation for an alternation free μ-calculus formula
  - particular case of diagnostic generation for a multiblock BES
- Definition of corresponding multiblock BES :

$$\begin{array}{lll}\{X_s & =_\nu & Y_s \wedge \bigwedge_{s \to s'} (Z_{s'} \vee X_{s'})\} \\ \{Y_s & =_\mu & \bigvee_{s \stackrel{acc}{\to} s'} \mathrm{T} \vee \bigvee_{s \to s'} Y_{s'}\} \\ \{Z_s & =_\nu & \bigwedge_{s \stackrel{acc}{\to} s'} \mathrm{F} \wedge \bigwedge_{s \to s'} Z_{s'}\}\end{array}$$

- Advantages :
  - generic solution
  - direct creation of a distributed on-the-fly generator of test cases

EXTRACTOR : On-the-Fly Test Case Generator



EXTRACTOR vs. TGV

- Speedup :

$$\frac{\sum_{\text{LTSs}} \mathit{time}(\text{TGV})}{(\sum_{\text{LTSs}} \mathit{time}(\text{EXTRACTOR}) + \sum_{\text{CTGs } \mathit{interm.}} \mathit{time}(\text{DETERMINATOR}))} = 1.82$$

- Memory consumption :

$$\frac{\sum_{\text{LTSs}} \mathit{memory}(\text{TGV})}{(\sum_{\text{LTSs}} \mathit{memory}(\text{EXTRACTOR}) + \sum_{\text{CTGs } \mathit{interm.}} \mathit{memory}(\text{DETERMINATOR}))} = 1.05$$

- Size of CTGs :

$$\frac{\sum_{\text{LTSs}} \mathit{stateNumber}(\text{TGV})}{\sum_{\text{CTGs } \mathit{interm.}} \mathit{stateNumber}(\text{DETERMINATOR})} = 0.71$$

$$\frac{\sum_{\text{STEs}} \mathit{transNumber}(\text{TGV})}{\sum_{\text{CTGs } \mathit{interm.}} \mathit{transNumber}(\text{DETERMINATOR})} = 0.53$$

- Treated examples on which TGV fails :

EXAMPLE	10^3 states	10^3 trans.	EXTRACTOR + DETERMINATOR
cwi_214_684	214	684	8 s., 19 Mb, no test case
cwi_566_3984	566	3 984	1195 s., 145 Mb, (32 states, 49 trans.)

Outline

Summary

1. Generic engine for distributed on-the-fly verification :
   - Resolution of monoblock BES (DSOLVE)
   - Resolution of multiblock BES (MB-DSOLVE)
2. Connection to real tools for formal verification :
   - On-the-fly equivalence checking (BISIMULATOR)
   - On-the-fly partial-order reduction (TAU_CONFLUENCE)
   - On-the-fly model-checking of temporal logic formulae (EVALUATOR)
3. Application to test generation :
   - Encoding of on-the-fly conformance test case generation in terms of BES (EXTRACTOR)
4. Massive tool experimentation on industrial study-cases and real parallel machines

Future Work

- Completing existing applications :
  - Encoding of other equivalences : Markovian bisimulation [Hermanns-Siegle-99], abstract relation [Holzmann-Joshi-04]
  - Encoding of other reductions : tau-inertness [Groote-Sellink-90], weak tau-confluence [Groote-vandePol-00]
- Developing other applications over DSOLVE and MB-DSOLVE :
  - Horn clauses resolution `[Liu-Smolka-98]`
  - Workflow analysis and abstract interpretation `[Fecht-Seidl-96]`
- Study other strategies of BES resolution
- Generalizing the approach to heterogeneous architectures, such as NOWS, and computation grids

Bibliography

- H. Garavel, M. Mateescu, I. Smarandache, A. Curic, D. Bergamini, N. Descoubes, C. Joubert and G. Stragier.
  DISTRIBUTOR and BCG_MERGE : Tools for Distributed Explicit State Space Generation.
  TACAS'2006, To appear.
- C. Joubert and R. Mateescu.
  Distributed On-the-Fly Model Checking and Test Case Generation.
  SPIN'2006, LNCS 3925 :126–145.
- D. Bergamini, N. Descoubes, C. Joubert and R. Mateescu.
  BISIMULATOR : A Modular Tool for On-the-Fly Equivalence Checking.
  TACAS'2005, LNCS 3440 :581–585.
- C. Joubert and R. Mateescu.
  Distributed Local Resolution of Boolean Equation Systems.
  PDP'2005, IEEE 264–271.
- C. Joubert and R. Mateescu.
  Distributed On-the-Fly Equivalence Checking.
  PDMC'2004, ENTCS 128(3).
- Christophe Joubert.
  Distributed Model-Checking : From Abstract Algorithms to Concrete Implementations.
  PDMC'2003, ENTCS 89(1).