Fault Tolerant PVM Application? — A Case Study with D-CICADA(FT)

Luděk Matyska[1,2] **and Karel Taft**[1]
[1]Department of Information Technology
Faculty of Informatics
and
[2]Institute of Computer Science
Masaryk University
Brno, Czech Republic

1. D-CICADA
2. Master-Slave distributed paradigm
3. “Robust” PVM?
4. “Robust” Master and Slave
5. Conclusion

D-CICADA

Distributed travelling on potential energy hypersurfaces

Potential energy (hyper)surface (PES)

$$E = f(\textbf{3D coordinates})$$

3n–6 degrees of freedom

Minimum = stable conformation
Conversion of conformations = **paths in PES**

Basic problem:
To find all minima and all low energy paths between them.

Algorithm

Set C of all known minima on PES
Set T of torsions

1. Take a vertex $v \in C$ with active torsion
2. Drive since
   (a) new minimum is found
   (b) already known minimum is found

   Store the path found and go back to step 1

Decomposition

Step 2b needs the knowledge of the whole graph
Possible solutions:

1. The graph is shared
2. The graph is kept on one node, other are asking

Distributed algorithm

Slave: one path

Master: boundary conditions, paths scheduling

Weak Points of Master–Slave Paradigm

Sources:

1. pvmd daemon
2. Master looses connection with slave(s)
3. Slave looses connection with the master

Solutions:

1. No simple solution, only suggestions:
   (a) pvmd running on a separate node, where no “hard” task runs.
   (b) supermaster

2. (a) detection of possible communication break
   (b) “roll-back” in case of confirmed slave crash
   (c) retargetting of “slow” nodes/slaves

3. (a) check pointing
   (b) master restarts under the supervising role of the supermaster
   (c) collection of internal information from slaves

Conclusion

- Enhancing the robustness of the PVM itself
- Program modification
  (under the master-slave paradigm)
- D-CICADA(FT) — robust PVM implementation