Contribution ID: **106** Type: **Oral**

Chip Test Generation Based on Black-Box Model

Thursday, 5 March 2009 16:20 (20 minutes)

Large and complex chip model testing is a computationally intensive task. It is not possible to test every single input and get output correlations for all tests. The search space is too big to fit any supercomputer. However, using Monte Carlo methods and splitting computations across the grid it is still possible to accumulate enough information about black-box inner workings. Here I am going to present how grid resources were allocated and utilized to achieve superior results.

Impact

The gridification process permitted chip test design time reduction by a couple orders of magnitude. Grid resources enabled larger and more complex chip analysis. The resulting relation matrix for each model accumulated through all computing clusters surpassed any previous results achieved during test design. The resulting data sets got much larger, thus new problems were introduced during data visualization.

URL for further information

http://saulius.org/rmap/

Conclusions and Future Work

The grid is very efficient in solving CPU intensive problems. However, efficient resource allocation is not so intuitive. Future work will be dedicated for chip analysis automation using grid technologies and result visualisation.

Keywords

Analysis, Black-Box, Chip, Resource allocation, Tests

Detailed analysis

Five black-box chip models with at least few hundred inputs and outputs were analyzed using the grid. Computations were split in a parallel matter to cover as much test sets as possible. Single submitted tasks had only one restriction: time limit. Many attempts to finish tasks in a specified time frame had failed before due to long queues. Tasks would get scheduled and were not able to use all proxy time requested by user proxy. In many cases tasks spent more time being Scheduled than Running. So another, more flexible restriction was substituted. Time limit was set only when task started running on the Working Node. Time left for computation was then derived from VOMS proxy information and CPU time limit. In such way it was possible to get the most efficient use of resources to complete model analysis.

Author: Mr PETRAUSKAS, Saulius (Kaunas University of Technology)

Presenter: Mr PETRAUSKAS, Saulius (Kaunas University of Technology)

Session Classification: New Application Areas

Track Classification: Experiences from application porting and deployment