EDX – A Commercial Parallel Simulator Platform

Janis Britals[1]

[1]Incontrol Enterprise Dynamics GmbH, Gustav-Stresemann-Ring 1, 65189 Wiesbaden

Premise

Theory of optimistic parallel simulation algorithms has been explored for quite some time. Successful implementations in commercial simulation packages, however, have so far been missing. Incontrol Enterprise Dynamics has set up a research program directed at creation of such platform based on the latest trends in software engineering.

Key features

During research into improvement aspects for our simulation platform Enterprise Dynamics we gradually implemented various features required for parallel discreet event simulation algorithms: a complete state saving and rollback mechanism, support for efficient event queue driven multithreading and bindings to Lua scripting language. These advances have led us to an implementation of generic parallel simulation environment enabling the model builder to designate parts of his/her model to be executed in a separate simulation thread with a click of a mouse. Thanks to usage of a script language the same code can be executed both in parallel as well as sequential settings allowing the model builder the utmost flexibility in exploring the potential parallelism in his/her model.

Optimistic parallel execution is based on an implementation of Time Warp Operating System (TWOS), described by Fujimoto.

Applications

Immediate applications of our implementation of parallel simulation engine come in the areas of modelling, visualization and hardware emulation. Modelling is improved by adding "undo" function that can use the same rollback mechanism used by TWOS. The simulation can also be run "backwards" if necessary.

Visualization and animation of the simulation model can be performed in a parallel simulation thread, removing performance impact caused by running visualization code as part of the simulation. The visualized state can then be rolled back (when performing "undo" or running backwards for instance) by TWOS rollback mechanism.

For hardware emulation or human-in-the-loop training simulations it is essential that outside interaction with the model can be time stamped with the exact simulation time independently of local computer processing (in case the simulation process takes a lot of computing resources). Again the state of simulation can be brought in sync with the external event using TWOS rollback feature, if necessary.