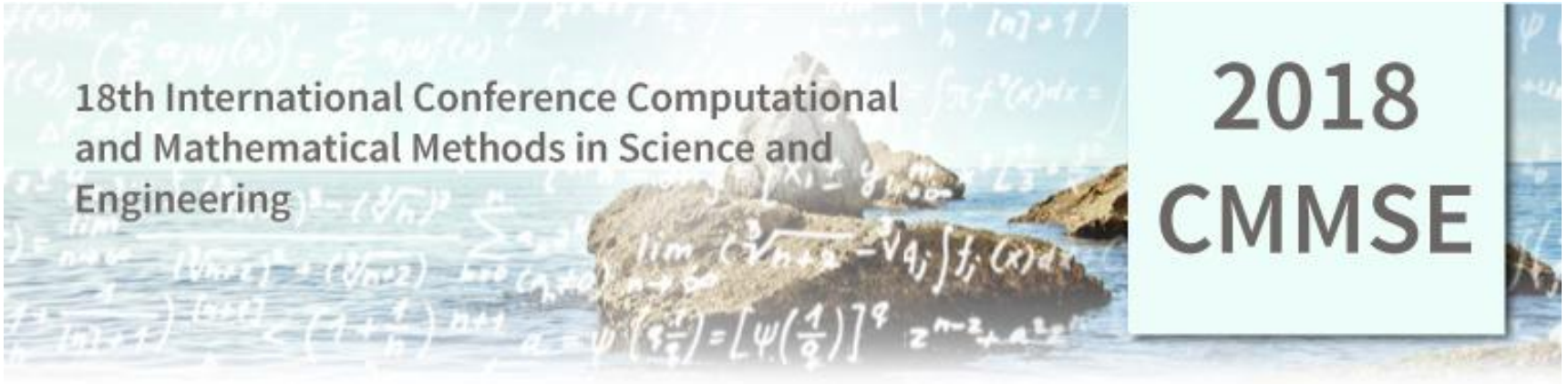


A Parallel Simulator for the Kinematic Analysis of Multibody Systems based on Group Equations

Javier Cuenca
José-Carlos Cano
Domingo Giménez
Mariano Saura-Sánchez
Pablo Segado-Cabezos

UNIVERSIDAD DE MURCIA

parallelum Scientific Computing & Parallel Programming University of Murcia

Motivation

- WHY
  - Recent mathematical formulations of mechanical systems facilitate the application of parallelism techniques to reduce the execution times when solving in computer systems
  - Parallel software requires an appropriate value selection of a set of adjustable parameters (AP) → best performance

- WHERE
  - In today's computational systems: multicore CPU + GPUs

- WHAT :
  - Select the most efficient linear algebra library
  - Maximize the use of available processing units

- HOW
  - Developing the software to simulate the computational workload of the problems to solve
  - Searching for the best adjustable parameters

Contents

- Group Equations formulation
- Simulator:
  - Problem definition
  - Simulation process
- Experimental results
- Conclusions and future work

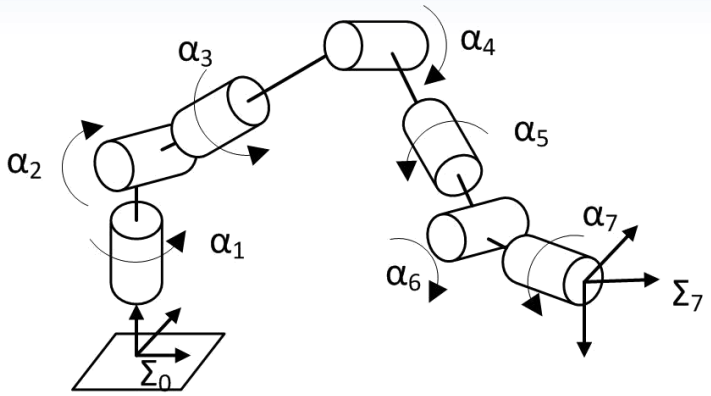
Contents

- **Group Equations formulation**
- Simulator:
  - Problem definition
  - Simulation process
- Experimental results
- Conclusion and future work

Group Equations formulation

Main concepts

- **Mechanical Systems** are composed of rigid and flexible bodies connected through mechanical joints
- The number of independent movements determine the **Degrees of Freedom** of the system (***DOF***)



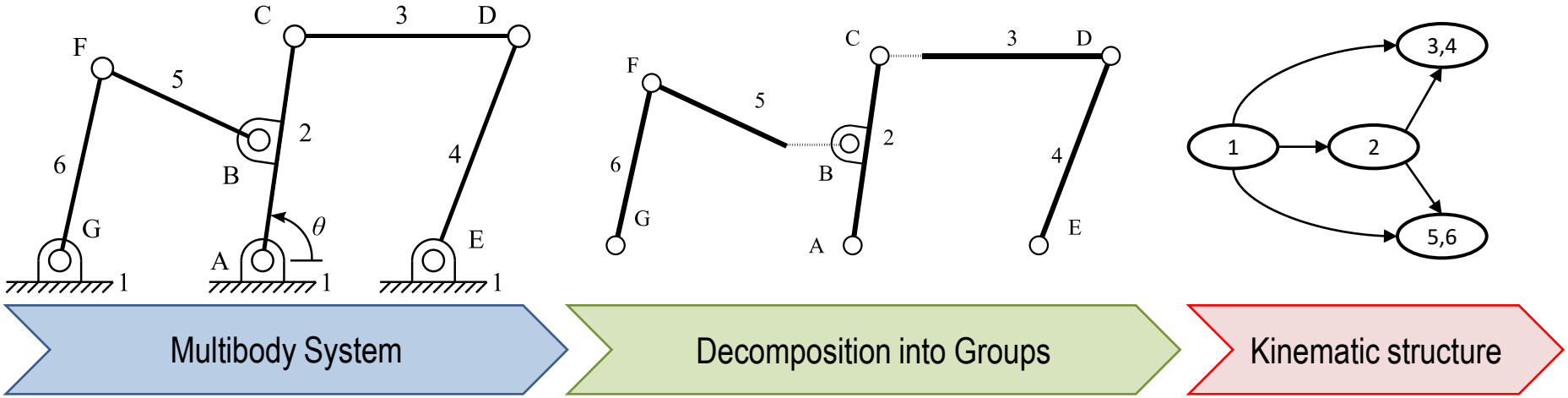
- **Kinematic analysis** of mechanical systems use numerical methods, exploiting **linear algebra** to calculate the **position** and **movement** of its components
- Elements:
  - A vector of ***n*** coordinates
    - Describe the position and orientation of all the bodies
  - A set of ***m = n–DOF*** constraint equations
    - describe the topological structure of the system
  - ***m*** dependent coordinates
    - calculated by means of the ***m*** constraint equations

Group Equations formulation

Description

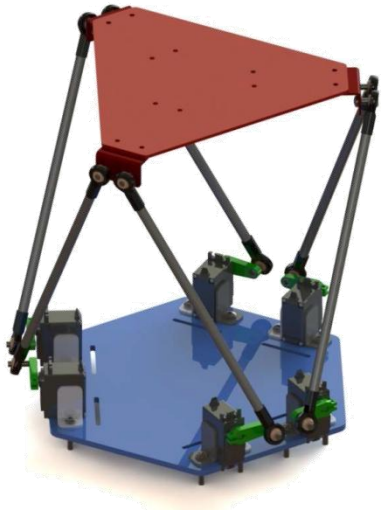
- Classical kinematic formulations that define the position of all the body components are known as **Global** formulations (using a number of coordinates to define the position of each body independently of the position of the other bodies)

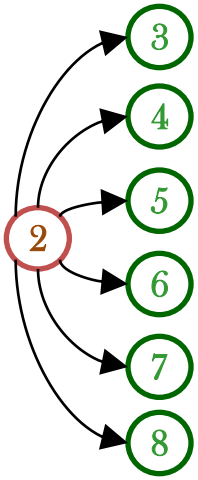
- On the other hand, **Group Equations** formulation divides the system in subsets of (coordinates & constraint equations). One subset for each independent subsystem (or group)



Group Equations formulation

Example: Stewart Platform



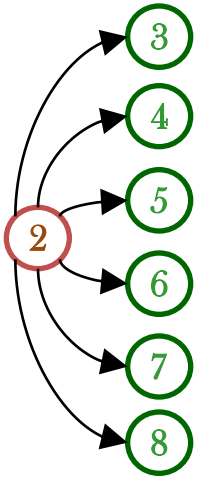



```
1  do while true
2     Solve kinematic of terminal (group 2)
3     for all structural components (groups 3 to 8) do
4        for internal iterations do
5           Solve kinematic of the structural component
6        end for
7     end for
8  end do
```

Multibody System → Decomposition into Groups → Kinematic structure → **Simulation**

Group Equations formulation

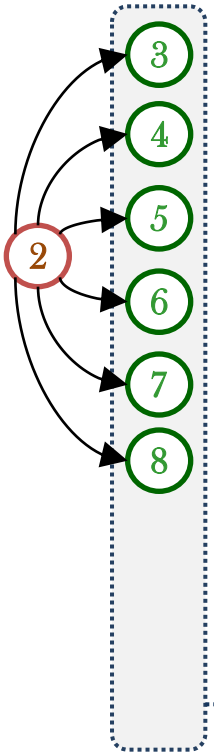
Example: Stewart Platform




```
1  do while true
2     Solve kinematic of terminal (group 2)
3     for all structural components (groups 3 to 8) do
4        for internal iterations do
5           Solve kinematic of the structural component
6        end for
7     end for
8  end do
```

Group Equations formulation

Example: Stewart Platform



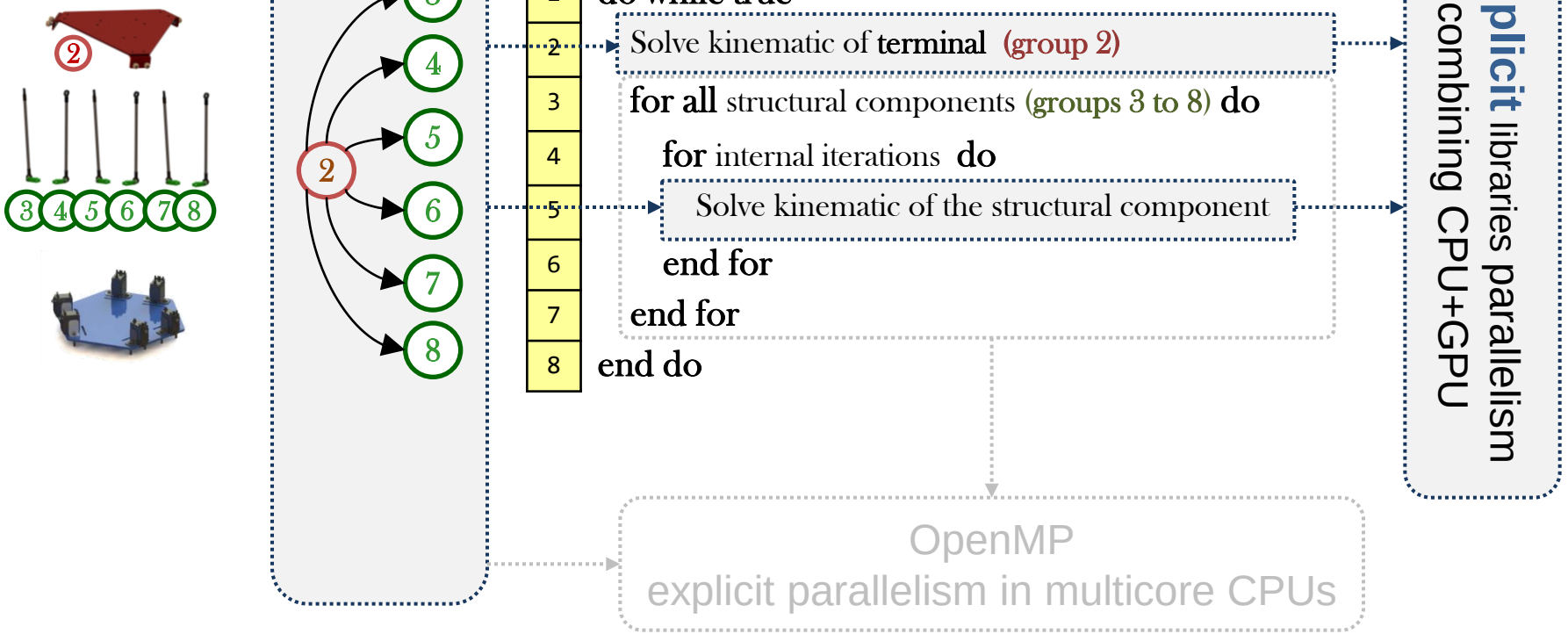

```
1  do while true
2    Solve kinematic of terminal (group 2)
3    for all structural components (groups 3 to 8) do
4      for internal iterations do
5        Solve kinematic of the structural component
6      end for
7    end for
8  end do
```

OpenMP

explicit parallelism in multicore CPUs

Group Equations formulation

Example: Stewart Platform



Contents

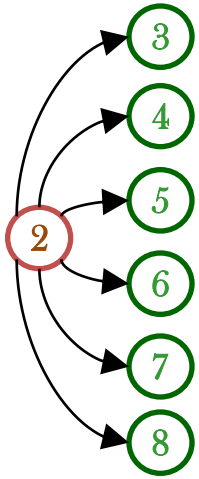
- Group Equations formulation
- **Simulator:**
  - Problem definition
  - Simulation process
- Experimental results
- Conclusion and future work

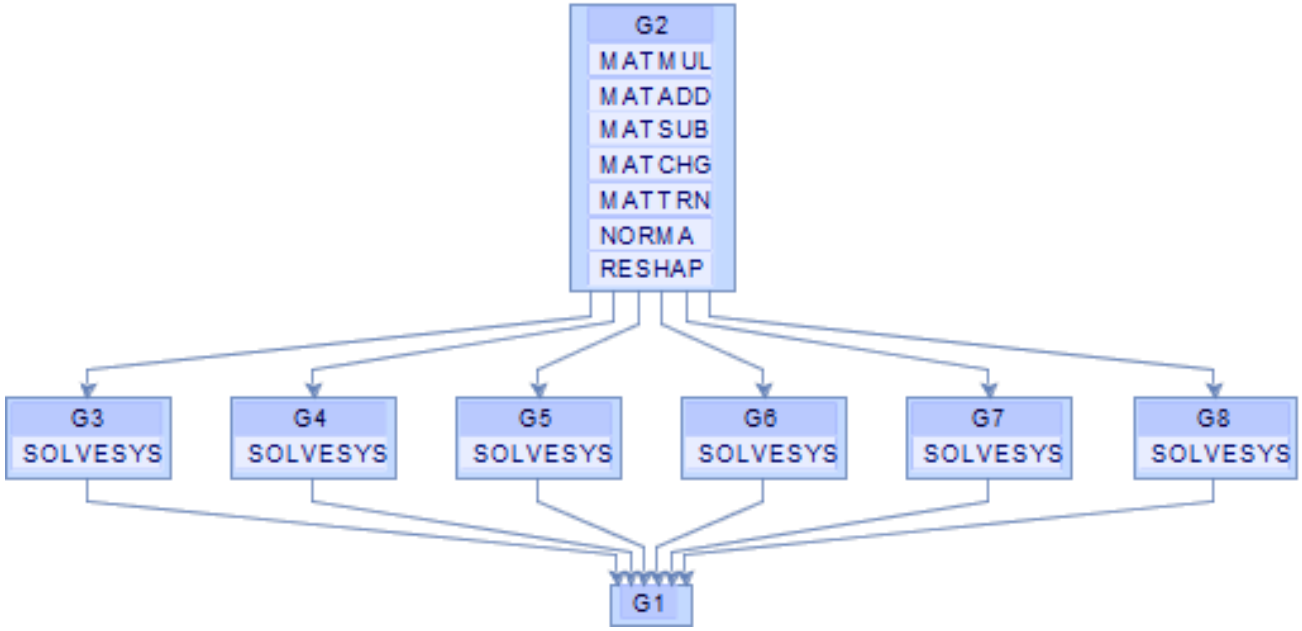
Contents

- Group Equations formulation
- Simulator:
  - Problem definition
  - Simulation process
- Experimental results
- Conclusions and future work

Simulator
Problem definition

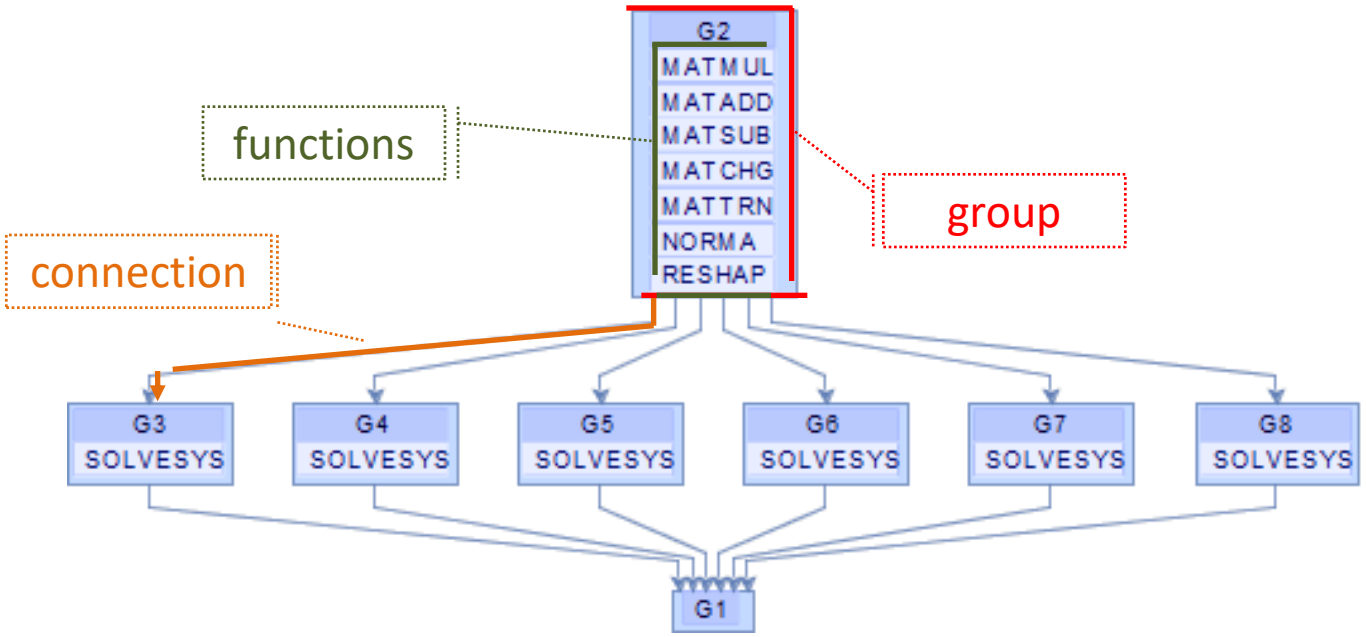
- In order to extend the simulations to a variety of multibody systems, a GUI tool has been developed
- The user can describe graphically the problem to solve, specifying the computation nodes and their connections (dependences)





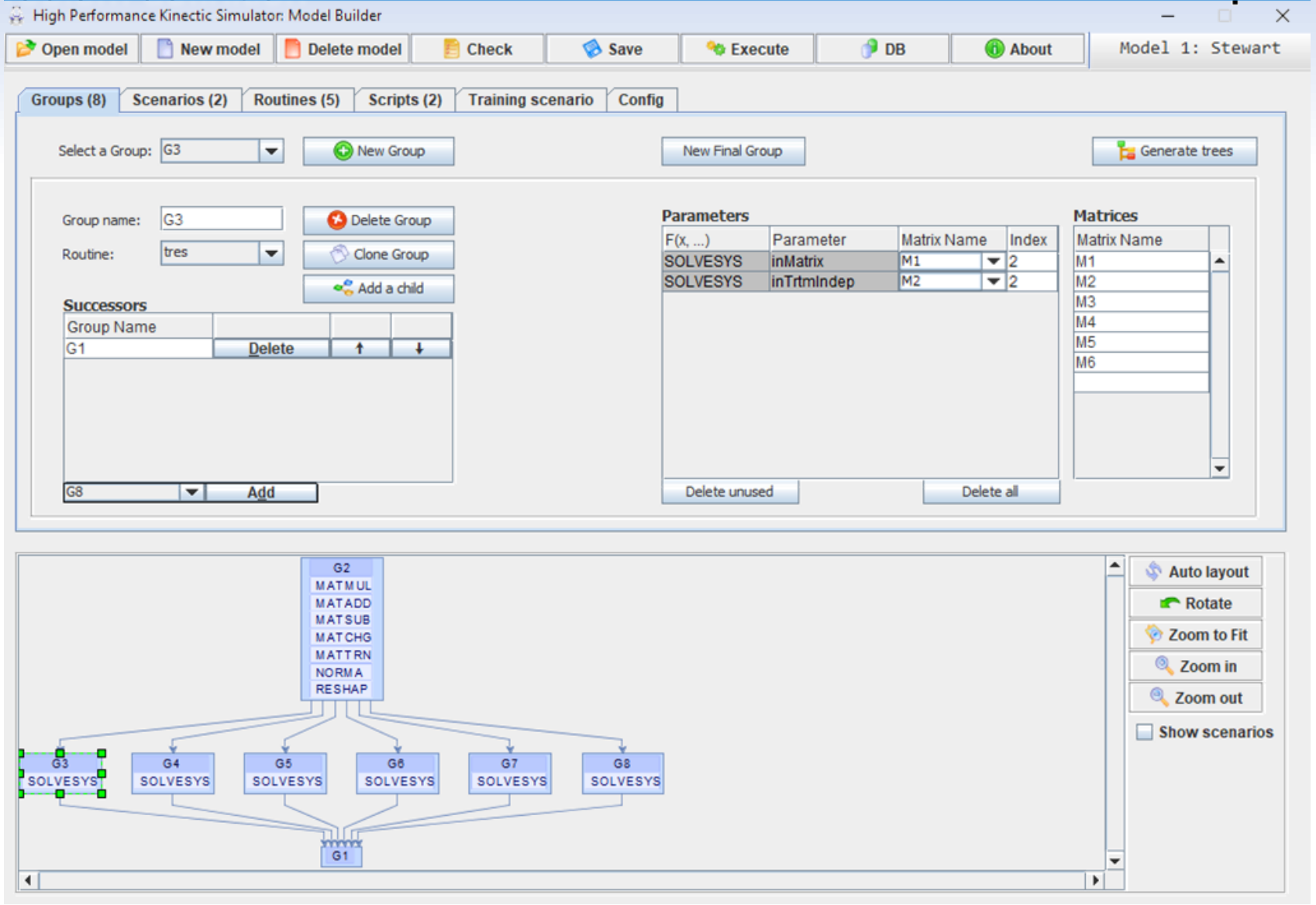
Simulator
Problem definition

- In order to extend the simulations to a variety of multibody systems, a GUI tool has been developed
- The user can describe graphically the problem to solve, specifying the computation nodes and their connections (dependences)

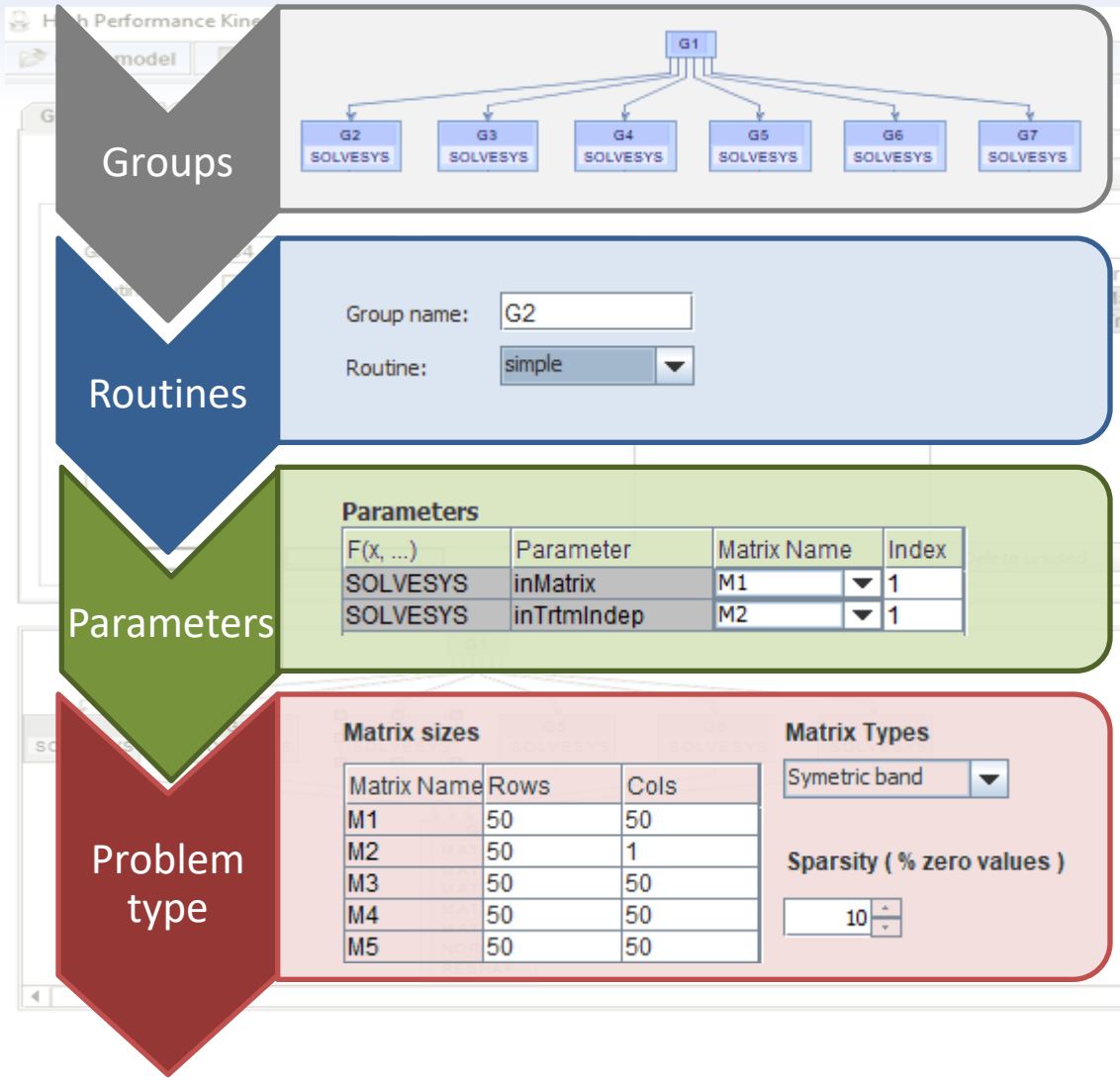


Simulator

Problem definition



Simulator
Problem definition



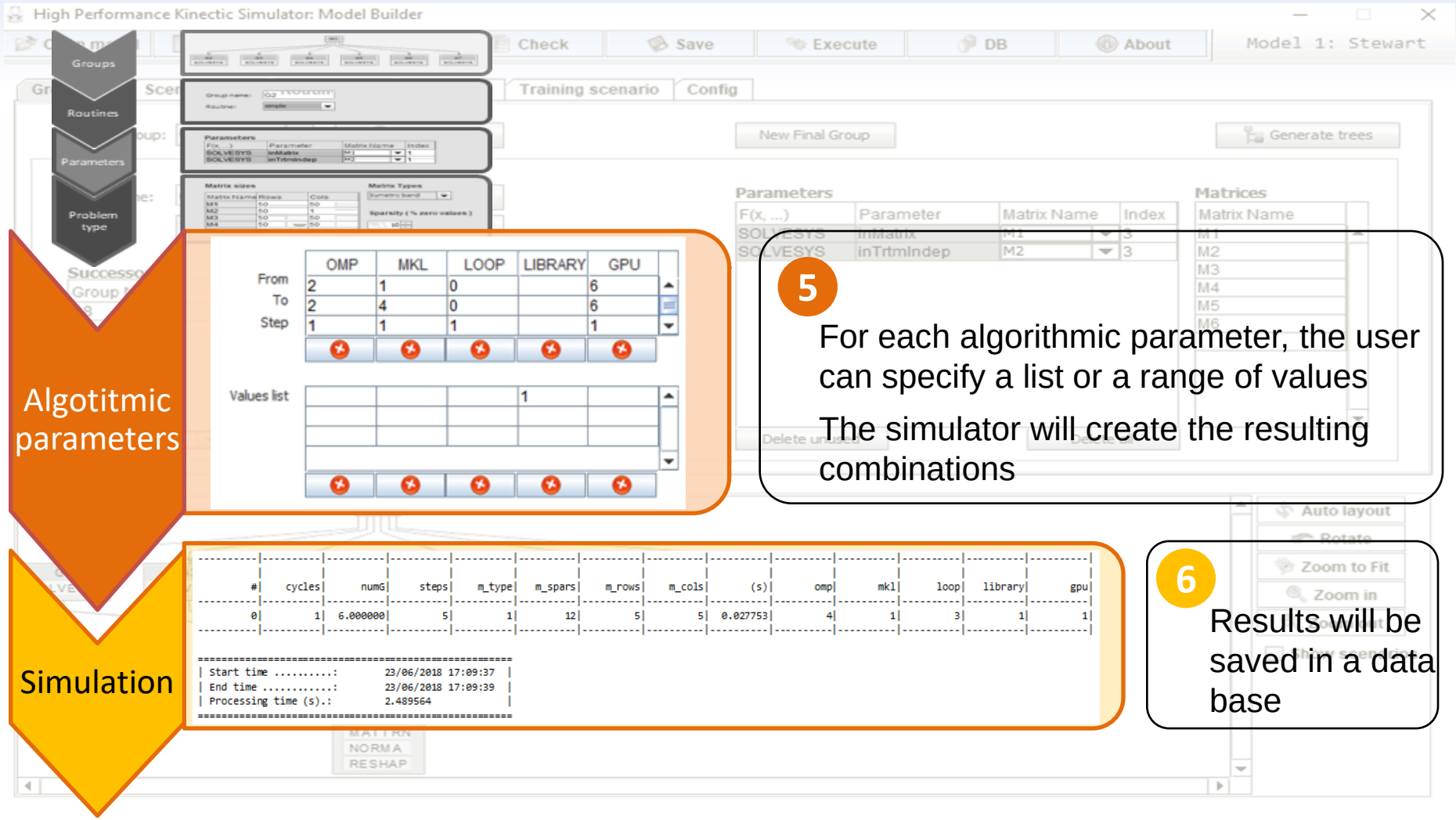
1. Add groups and their depedencies
2. Specify the routines (calculations) in the groups
3. Specify parameters of the routines
4. Specify: size and type of the parameters

Contents

- Group Equations formulation
- Simulator:
  - Problem definition
  - Simulation process
- Experimental results
- Conclusions and future work

Simulator

Simulation Process



Contents

- Group Equations formulation
- Simulator:
  - Problem definition
  - Simulation process
- Experimental results
- Conclusions and future work

Experimental Results

- As an example of use, the following experiments have been carried out in a Stewart Platform model. We analyze:
  1. the influence of the **sparsity of the matrices** in the selection of the most suitable linear algebra library
  2. the best balance between explicit and implicit parallelism in a **two-level parallel configuration**
  3. The type of problems in which the use of **GPUs** leads to lower execution times
- Hardware platforms:

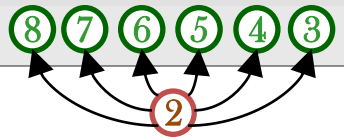
Hardware
❑ **12CORES + 6GPU**: two hexa-cores (12 cores) Intel Xeon E5-2620 + 6 NVIDIA GPUs
❑ **24CORES + 1GPU**: four hexa-cores (24 cores) Intel Xeon E7530 + 1 NVIDIA GPU

Experimental Results

Influence of sparsity

24C+1GPU	sparsity				sparsity			
	10%	20%	50%	85%	10%	20%	50%	85%
size	Sequential				Parallel 24 threads			
12	MA27	MA27	MA27	MA27	MA27	MA27	MA27	MA27
36	LAPACK	LAPACK	LAPACK	MA27	MA27	LAPACK	LAPACK	MA27
60	LAPACK	LAPACK	LAPACK	MA27	LAPACK	LAPACK	LAPACK	MA27
400	LAPACK	LAPACK	LAPACK	MA27	LAPACK	LAPACK	LAPACK	LAPACK
1000	LAPACK	LAPACK	LAPACK	MA27	LAPACK	LAPACK	LAPACK	LAPACK
2000	LAPACK	LAPACK	LAPACK	MA27	LAPACK	LAPACK	LAPACK	LAPACK
3000	LAPACK	LAPACK	LAPACK	PARDISO	LAPACK	LAPACK	LAPACK	LAPACK
4000	LAPACK	LAPACK	PARDISO	PARDISO	LAPACK	LAPACK	LAPACK	PARDISO

Model: Stewart Platform (6 groups)



Libraries with the lowest execution times

Experimental Results

Influence of sparsity

24C+1GPU	sparsity				sparsity			
	10%	20%	50%	85%	10%	20%	50%	85%
size	Sequential				Parallel 24 threads			
12	MA27	MA27	MA27	MA27	MA27	MA27	MA27	MA27
36	LAPACK	LAPACK	LAPACK	MA27	MA27	LAPACK	LAPACK	MA27
60	LAPACK	LAPACK	LAPACK	MA27	LAPACK	LAPACK	LAPACK	MA27
400	LAPACK	LAPACK	LAPACK	MA27	LAPACK	LAPACK	LAPACK	LAPACK
1000	LAPACK	LAPACK	LAPACK	MA27	LAPACK	LAPACK	LAPACK	LAPACK
2000	LAPACK	LAPACK	LAPACK	MA27	LAPACK	LAPACK	LAPACK	LAPACK
3000	LAPACK	LAPACK	LAPACK	PARDISO	LAPACK	LAPACK	LAPACK	LAPACK
4000	LAPACK	LAPACK	PARDISO	PARDISO	LAPACK	LAPACK	LAPACK	PARDISO

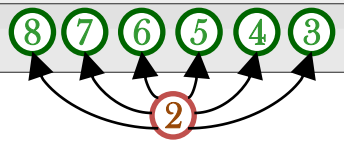
- PARDISO and MA27 are the preferred libraries for high sparsity problems
- For small problems, MA27 is the best option

Experimental Results

Simulation: two-level parallelism

Matriz size	12CORES + 6GPU		24CORES+1GPU	
	MA27 6 x 1	PARDISO 6 x 2	MA27 6 x 1	PARDISO 6 x 4
12	**0,0009**	0,0234	**0,0027**	0,0571
36	**0,0043**	0,0288	**0,0034**	0,0475
60	**0,0089**	0,043	**0,0088**	0,0682
400	**0,1559**	0,4945	**0,2746**	0,6976
1000	**1,6476**	2,4988	**2,6428**	3,5935
2000	10,3387	**7,759**	16,9799	**10,7895**
3000	32,8752	**16,732**	53,5882	**24,8652**
4000	76,5881	**31,6888**	120,8349	**52,3393**

Model: Stewart Platform (6 groups)



Sparsity degree=85%

Only configurations which give best results are shown

Experimental Results

Simulation: two-level parallelism

Matriz size	12CORES + 6GPU		24CORES+1GPU	
	MA27 6 x 1	PARDISO 6 x 2	MA27 6 x 1	PARDISO 6 x 4
12	**0,0009**	0,0234	**0,0027**	0,0571
36	**0,0043**	0,0288	**0,0034**	0,0475
60	**0,0089**	0,043	**0,0088**	0,0682
400	**0,1559**	0,4945	**0,2746**	0,6976
1000	**1,6476**	2,4988	**2,6428**	3,5935
2000	10,3387	**7,759**	16,9799	**10,7895**
3000	32,8752	**16,732**	53,5882	**24,8652**
4000	76,5881	**31,6888**	120,8349	**52,3393**

Model: Stewart Platform (6 groups)

Sparsity degree=85%

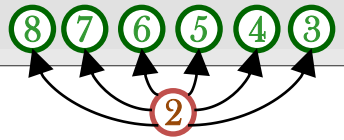
- Explicit parallelism with the MA27 library is the best option for small and medium problems
- For large problems, PARDISO with a two-level parallelism schema is the fastest option

Experimental Results

Simulation: CPU vs GPU

12C+6GPU Matriz size	th.OMP x MA27 6 x 1	th.OMP x th.PARDISO 6 x 2	MAGMA 2	MAGMA 3	MAGMA 6
12	**0,0009**	0,0234	0,0073	0,0064	4,1789
36	**0,0043**	0,0288	0,0073	0,0066	0,1242
60	**0,0089**	0,043	0,0326	0,0269	0,2362
400	**0,1559**	0,4945	1,2254	1,1266	1,262
1000	**1,6476**	2,4988	2,7287	2,6829	2,2188
2000	10,3387	7,759	9,2946	7,8085	**5,9685**
3000	32,8752	16,732	23,7600	18,9474	**14,0435**
4000	76,5881	31,6888	48,3116	38,1400	**28,2109**

Model: Stewart Platform (6 groups)

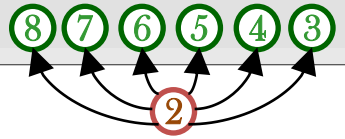


Sparsity degree=85%

Experimental Results
Simulation: CPU vs GPU

Matriz size	th.OMP x MA27 6 x 1	th.OMP x th.PARDISO 6 x 2	MAGMA 2	MAGMA 3	MAGMA 6
12	**0,0009**	0,0234	0,0073	0,0064	4,1789
36	**0,0043**	0,0288	0,0073	0,0066	0,1242
60	**0,0089**	0,043	0,0326	0,0269	0,2362
400	**0,1559**	0,4945	1,2254	1,1266	1,262
1000	**1,6476**	2,4988	2,7287	2,6829	2,2188
2000	10,3387	7,759	9,2946	7,8085	**5,9685**
3000	32,8752	16,732	23,7600	18,9474	**14,0435**
4000	76,5881	31,6888	48,3116	38,1400	**28,2109**

Model: Stewart Platform (6 groups)



Sparsity degree=85%

- For MAGMA, the best results are obtained with each GPU working on a group
- The exploitation of GPUs is advisable only in the case of large problems and with a large number of GPUs

Contents

- Group Equations formulation
- Simulator:
  - Problem definition
  - Simulation process
- Experimental results
- Conclusions and future work

Conclusions and future work

- Mechanical systems that can be broken down into several independent groups offer a good opportunity to exploit parallelism in modern computer systems
- Taking Stewart Platform as a starting point, the generalization to other model structures has been addressed
- Using a new GUI tool, the user can generate and simulate his own problems
- It is possible to command the simulator to use different linear algebra libraries together with the parallelism parameters
- Experiments with several matrix sizes and degrees of sparsity have been presented
- Simulator will be freely available when pending functionalities were added:
  - Management of coprocessors
  - Auto-tuning engine