Introduction to Dimemas

Performance Analysis Tool for Parallel Programs and Platforms

dimemas@cepba.upc.es

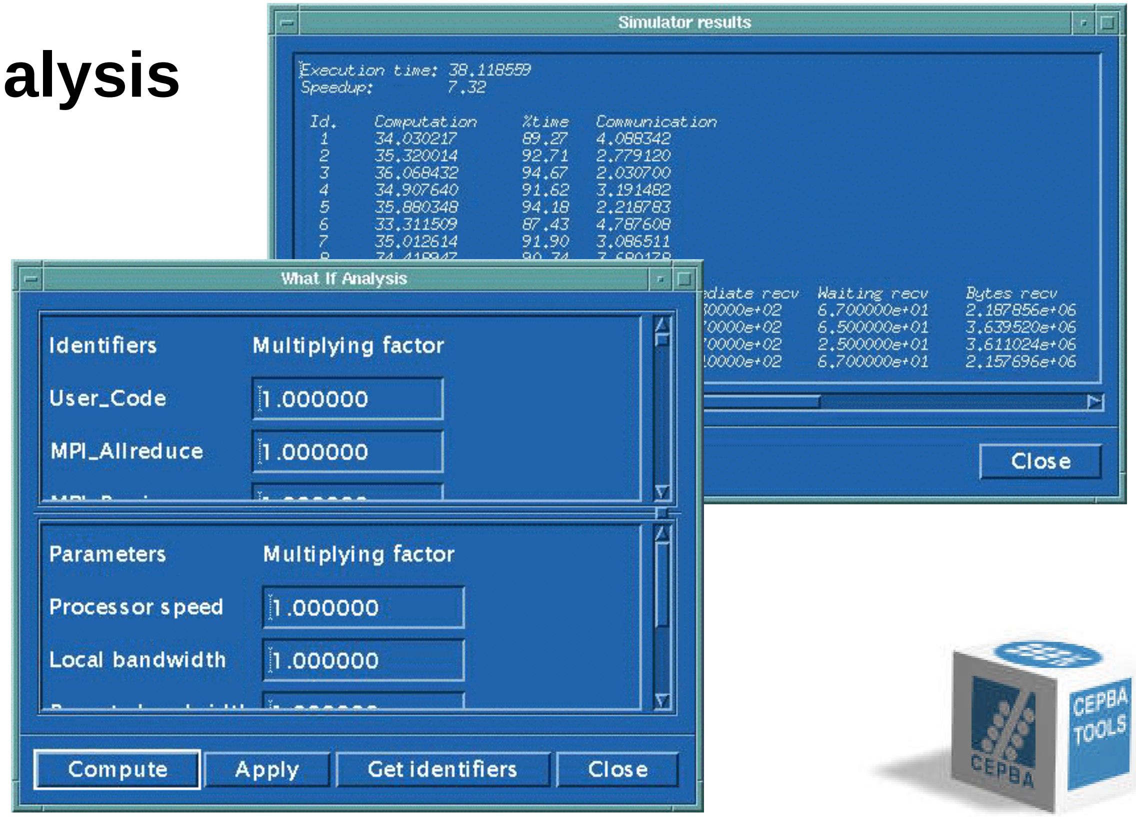
European Center for Parallelism of Barcelona

Technical University of Catalonia

Barcelona, Spain

Dimemas

- Application performance analysis tool for message passing programs
- In development since 1992
- Perform all the analysis on a workstation



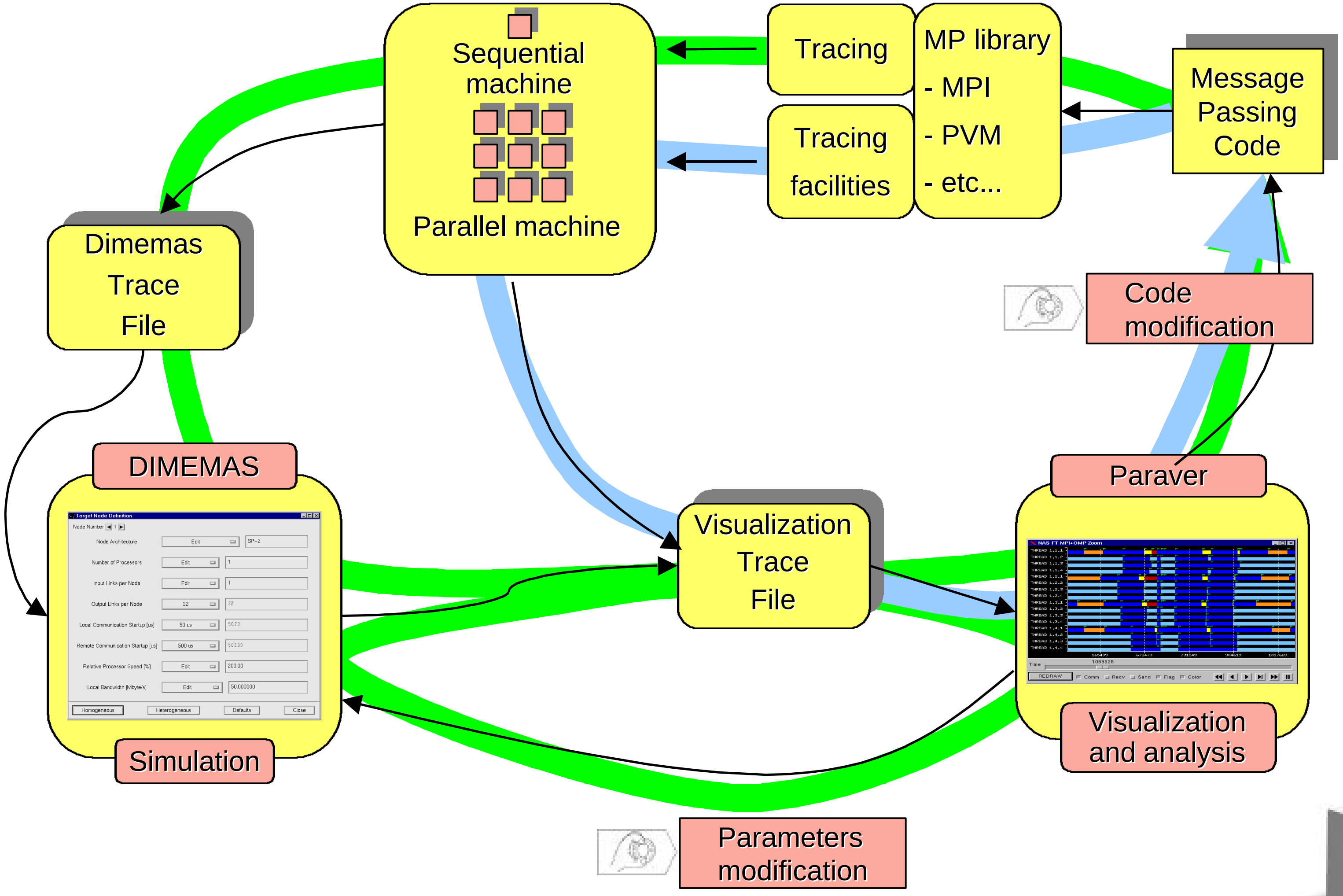
```
Execution time: 38.118559
Speedup:         7.32

 Id.    Computation    %time   Communication
  1     34.030217      89.27   4.088342
  2     35.320014      92.71   2.779120
  3     36.068432      94.67   2.030700
  4     34.907640      91.62   3.191482
  5     35.880348      94.18   2.218783
  6     33.311509      87.43   4.787608
  7     35.012614      91.90   3.086511
  8     34.418947      90.74   3.680178

ediate recv    Waiting recv     Bytes recv
30000e+02      6.700000e+01     2.187856e+06
70000e+02      6.500000e+01     3.639520e+06
70000e+02      2.500000e+01     3.611024e+06
10000e+02      6.700000e+01     2.157696e+06
```

Identifiers	Multiplying factor
User_Code	1.000000
MPI_Allreduce	1.000000

Parameters	Multiplying factor
Processor speed	1.000000
Local bandwidth	1.000000

dimemas@cepba.upc.es

Tuning Methodology



Sequential machine

Parallel machine

Tracing

Tracing facilities

MP library
- MPI
- PVM
- etc...

Message Passing Code

Code modification

Dimemas Trace File

DIMEMAS

Target Node Definition

Node Number 1

Node Architecture	Edit	SP–2
Number of Processors	Edit	1
Input Links per Node	Edit	1
Output Links per Node	32	32
Local Communication Startup [us]	50 us	50.00
Remote Communication Startup [us]	500 us	500.00
Relative Processor Speed [%]	Edit	200.00
Local Bandwidth [Mbyte/s]	Edit	50.000000

Homogeneous Heterogeneous Defaults Close

Simulation

Visualization Trace File

Paraver

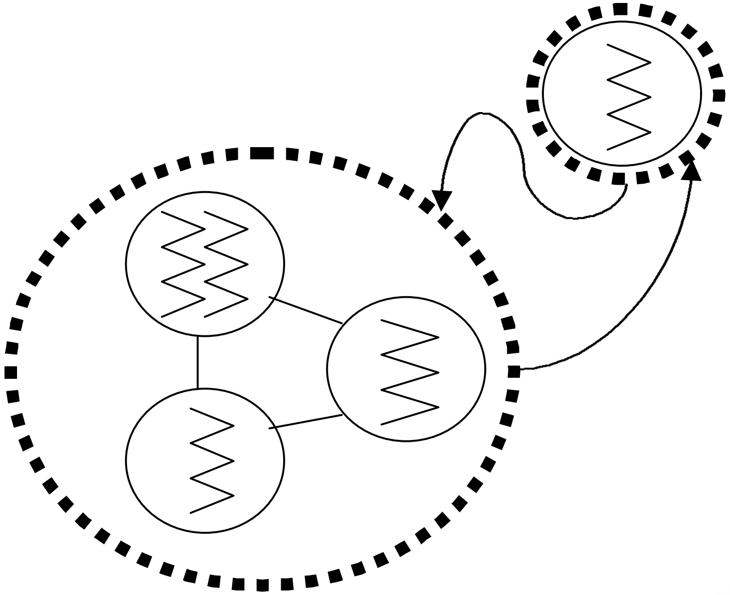
Visualization and analysis

Parameters modification

dimemas@cepba.upc.es

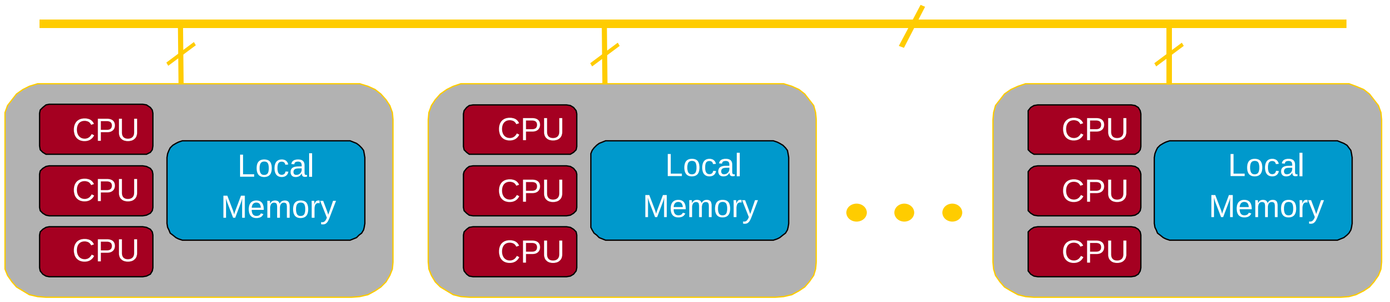
Tracefile

- **Characterizes application**
  - Sequence of resource demands for each task
    - ✓ **CPU bursts**
    - ✓ **Communication**
  - Sequence of events:
    - ✓ **Block/routine entry and exit flags**
- **Application model**



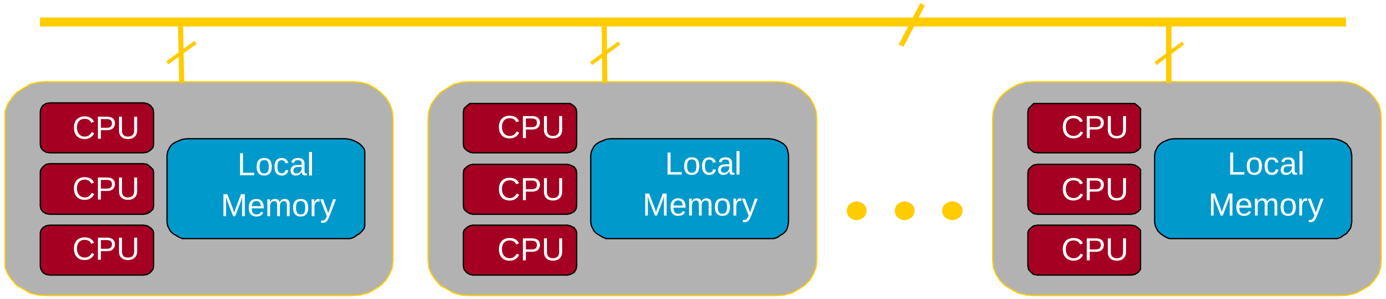
Simulated Architecture

- **"Abstract" architecture**
  - Simple and general
    - ✓ **Network of SMPs**
  - Fast simulation
  - Key factors influencing performance
  - Abstract interconnect
    - ✓ **Local/remote latency/BW**
    - ✓ **Injection mechanism (#links, half/full duplex)**
    - ✓ **Bisection BW, contention**



System

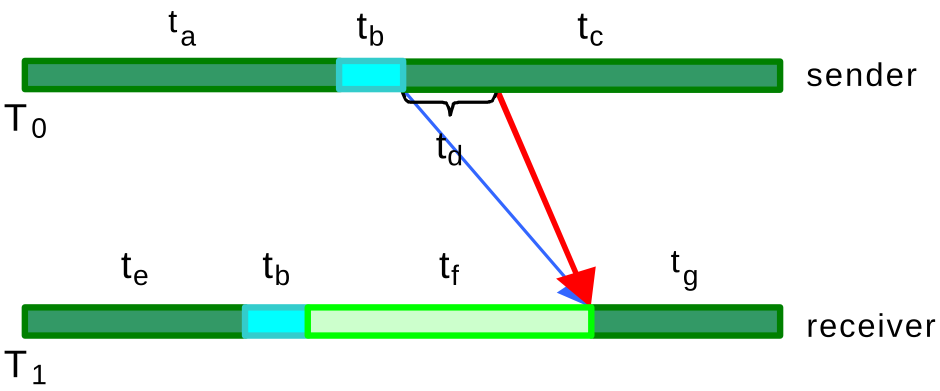
- **Process to processor mapping**
- **Multiprogramming**
  - Tasks sharing node
  - Different applications



Point to Point Communication

$$T = \text{Latency} + \frac{\text{Size}}{\text{Bandwidth}}$$

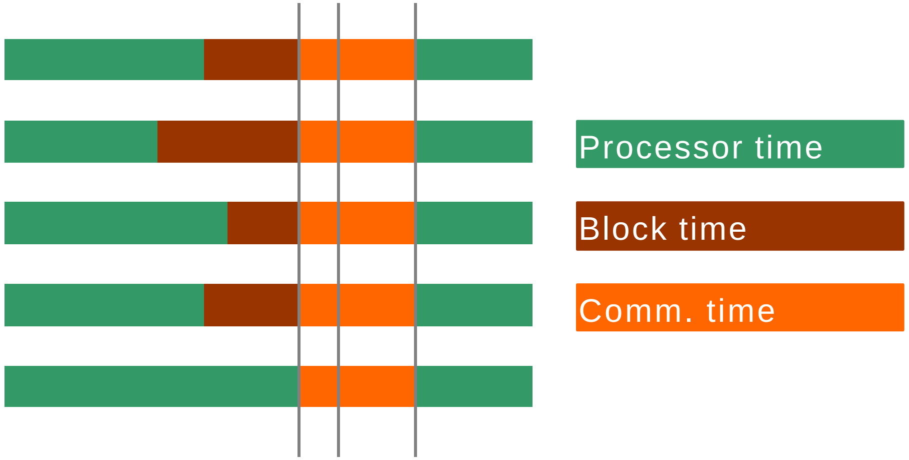
- **Latency**
- **Bandwidth**
- **Resource contention**

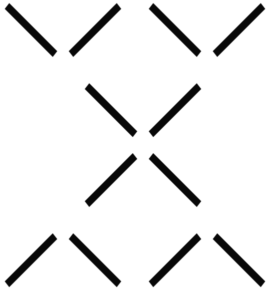


Collective Communication Model

Model hypotheses

- Implicit barrier
- Fan-in/fan-out phases
  - ✓ **Null/Constant/Linear/Logarithmic model for each phase**





Collective Communication Model

- **Communication time**

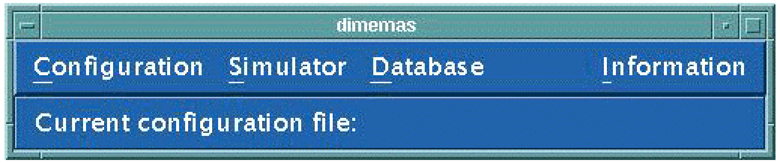
$$\text{Time} = \left(\text{Latency} + \frac{\text{Size}}{\text{Bandwidth}}\right) * \text{MODEL\_FACT} \quad \text{OR}$$

- **Model factor**

Model	Factor
Null	0
Constant	1
Linear	P
Logarithmic	$\text{Nsteps} = \sum_{i=1}^{\lceil \log_2 P \rceil} \text{steps}_i, \text{steps}_i = \left\lceil \frac{C}{B} \right\rceil$

Starting Dimemas

- **Invoke Dimemas gui:** dimemas
- **Dimemas main menu**



Current configuration file:

- **Configuration**
  - First time for each user

    ```
    mkdir $HOME/.DIMEMAS_defaults
    cp $DIMEMAS_HOME/DIMEMAS_defaults/* $HOME/.DIMEMAS_defaults
    ```

  - Every session

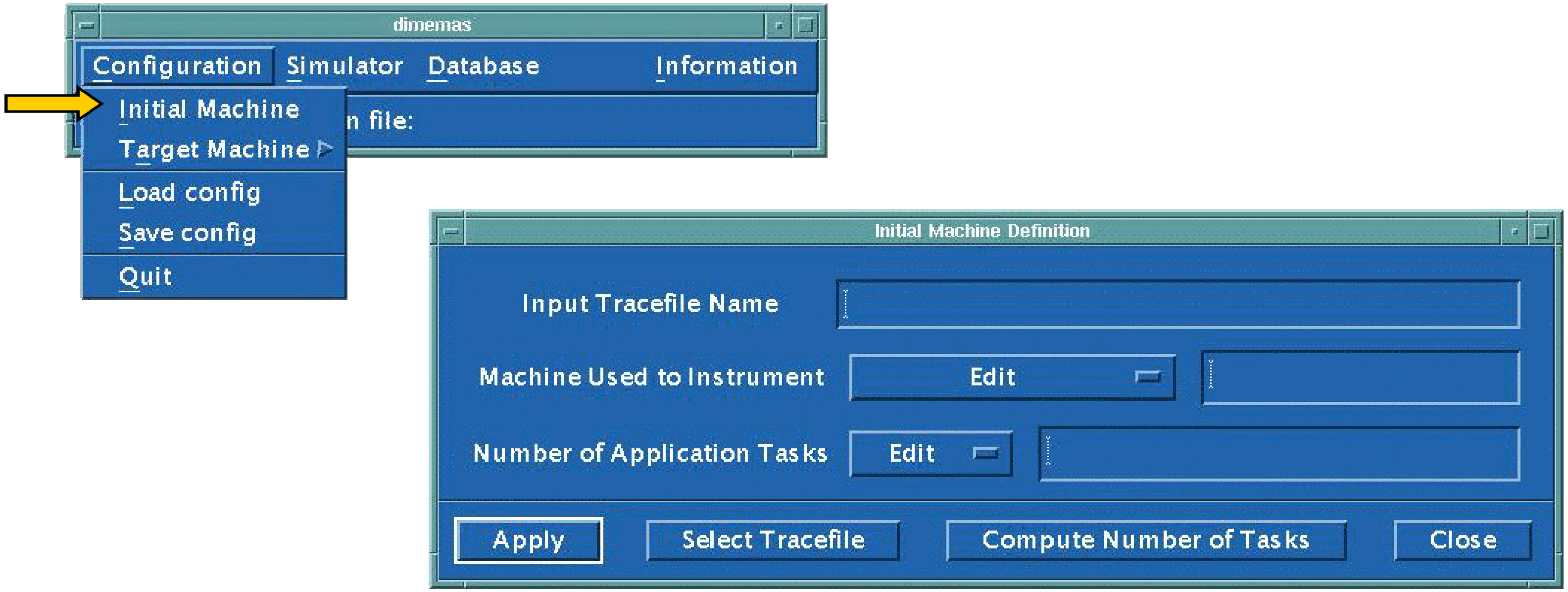
    ```
    setenv DIMEMAS_HOME /aplic/DIMEMAS
    ```

Dimemas files

- **Configuration files, located in $HOME/.DIMEMAS_defaults**
  - Dimemas
    - **Configuration file for X-Windows**
  - Machines.db
    - **Information of machines database**
    - **Some predefined values**
  - Network.db
    - **Information of network database**
    - **Some predefined values**
- **Executables, located in $DIMEMAS_HOME/bin**
- **License file, $DIMEMAS_HOME/etc/license.dat**

Initial machine definition

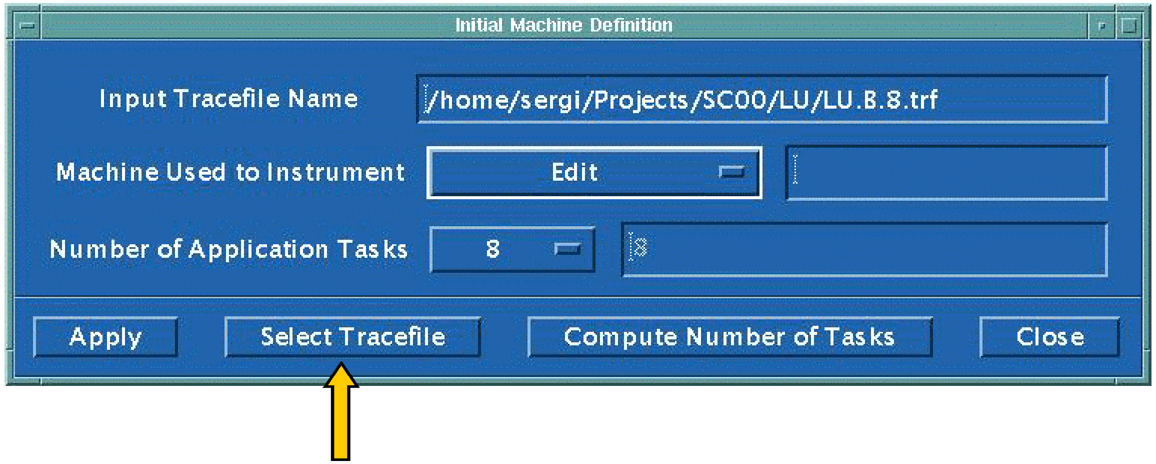
- To define where the tracefile comes from



- Select an input tracefile and define the initial machine

Initial machine definition

- **Select a tracefile**

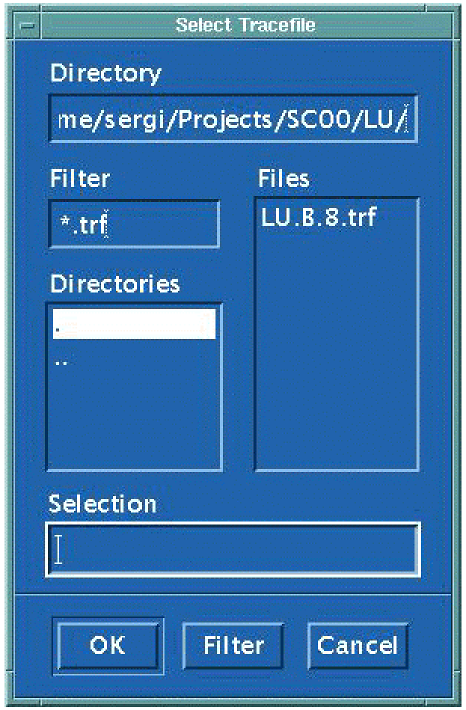


Initial Machine Definition

Input Tracefile Name: /home/sergi/Projects/SC00/LU/LU.B.8.trf

Machine Used to Instrument: Edit

Number of Application Tasks: 8 | 8



Select Tracefile

Directory: me/sergi/Projects/SC00/LU/

Filter: *.trf

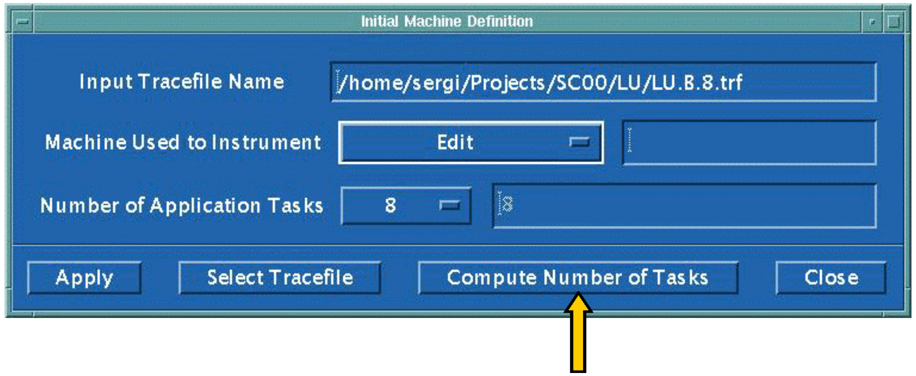
Files: LU.B.8.trf

Directories: . ..

Selection:

Initial machine definition

- **Specify trace file**
  - Type or use Select browser
- **Compute the number of tasks**
  - ✓ **Scans the trace file to compute the number of tasks**



Initial Machine Definition

Input Tracefile Name: /home/sergi/Projects/SC00/LU/LU.B.8.trf

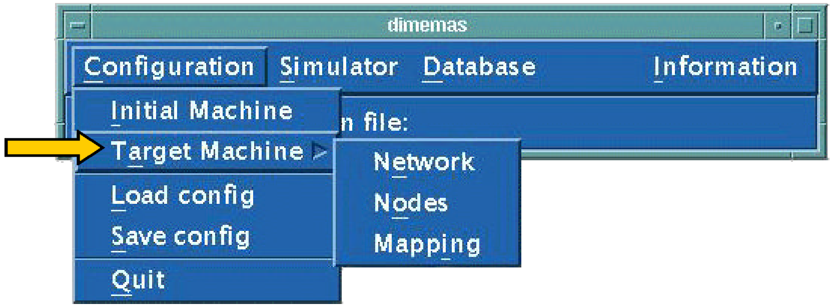
Machine Used to Instrument: Edit

Number of Application Tasks: 8 | 8

- **Enter the initial machine (optional)**
  - Specifies the machine where the trace was obtained
  - If used, the relative speed between initial and target machine can be obtained from the machines.db file
- **Click the Apply button to confirm the input**

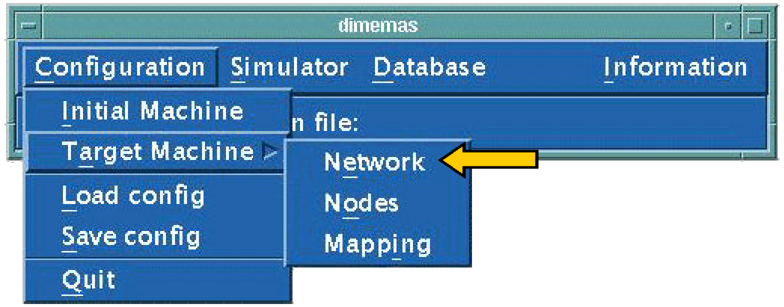
Target machine

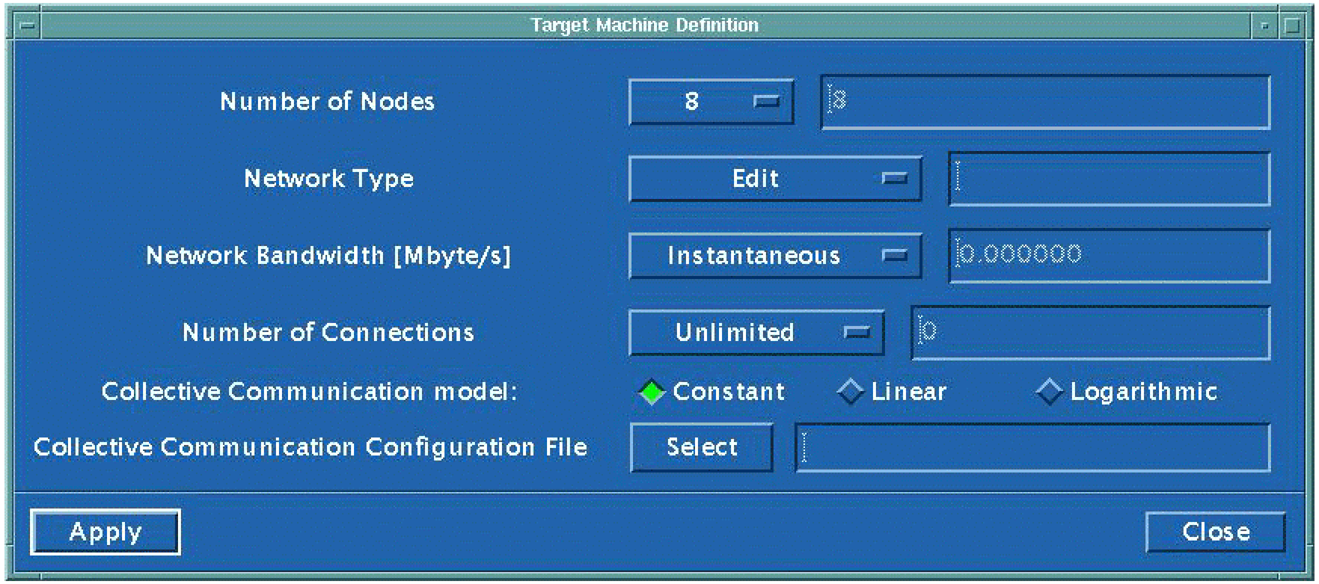
- Machine simulated by Dimemas
- Description composed by



- target network definition
- target node definition
- process to node mapping
- file system parameters

Network definition





Target Machine Definition

Number of Nodes	8	8
Network Type	Edit	
Network Bandwidth [Mbyte/s]	Instantaneous	0.000000
Number of Connections	Unlimited	0

Collective Communication model: ● Constant ◇ Linear ◇ Logarithmic

Collective Communication Configuration File [Select]

Network definition

- **Number of nodes in the system**
- **Network type (optional)**
  - If used, reads parameters from network database
- **Network bandwidth**
  - Bandwidth for inter-node messages
- **Number of connections**
  - Maximum number of simultaneous messages in transit (simple model of network contention)
- **Collective communication model**
  - Default model for all collective operations (constant/linear/log)
- **Collective communication configuration file (optional)**
  - If used, specifies a detailed model for each collective operation

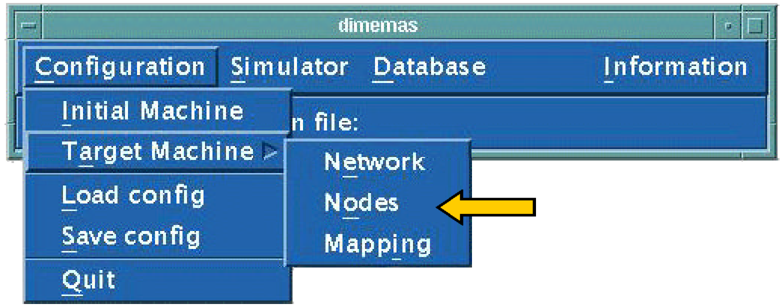
Network definition

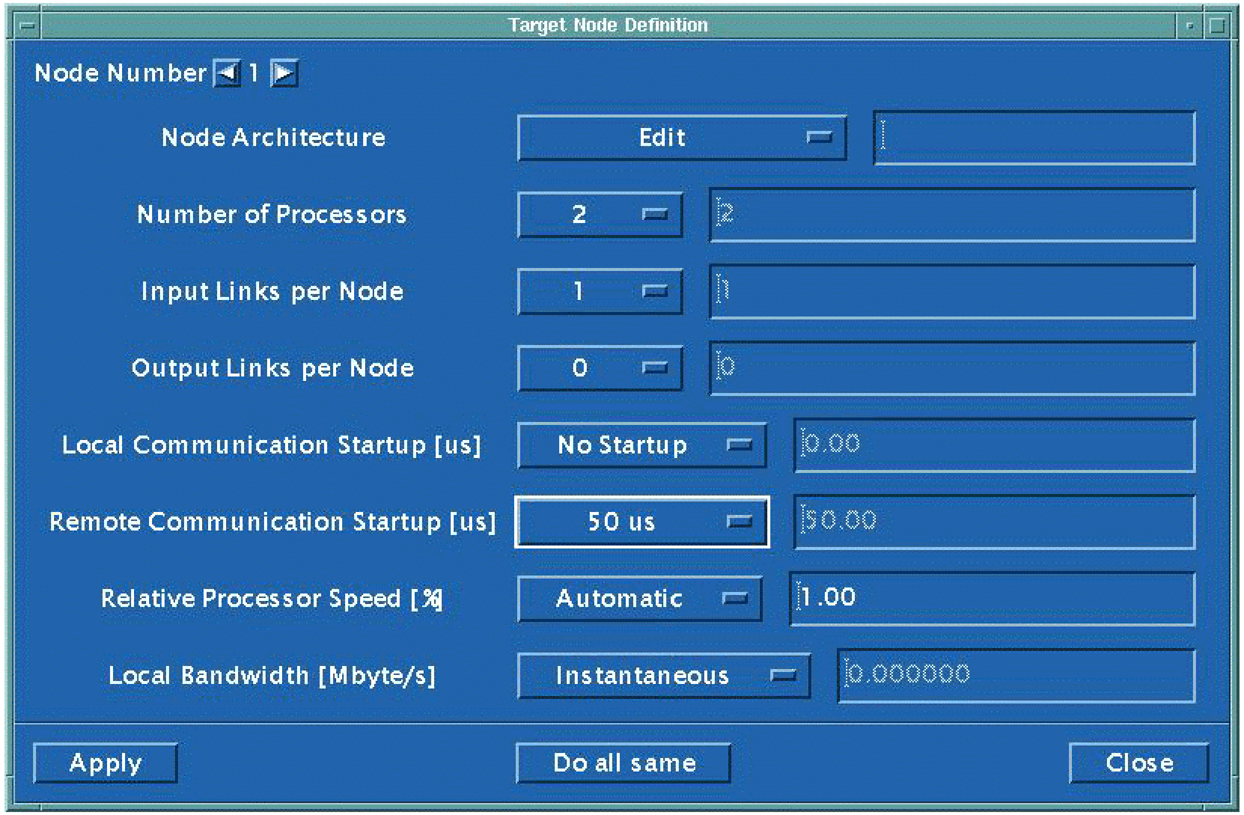
Collective communication configuration file format

- Model and message length to use for each phase
- Example

Funct. ID	Fan-in Model	Fan-in length	Fan-out model	Fan-out length		
Policy: FIFO						This line is mandatory
0	LIN	MAX	0	MAX	1.0	MPI_Barrier
1	LOG	MAX	0	MAX	1.0	MPI_Bcast
2	LOG	MEAN	0	MAX	1.0	MPI_ Gather
3	CTE	MAX	CTE	MAX	1.0	MPI_Gatherv
4	LOG	MEAN	0	MAX	1.0	MPI_Scatter
5	CTE	MAX	CTE	MAX	1.0	MPI_Scatterv
6	LIN	MIN	LIN	MIN	1.0	MPI_Allgather
7	LIN	MIN	LIN	MIN	1.0	MPI_Allgatherv
8	LIN	MIN	LIN	MIN	1.0	MPI_Alltoall
9	CTE	MAX	CTE	MAX	1.0	MPI_Alltoallv
10	LOG	MAX	0	MAX	1.0	MPI_Reduce
11	LOG	MAX	LOG	MAX	1.0	MPI_Allreduce
12	LOG	MAX	LIN	MIN	1.0	MPI_Reduce_Scatter
13	LOG	MAX	LOG	MAX	1.0	MPI_Scan

Node definition





Node Number 1

Node Architecture	Edit	
Number of Processors	2	2
Input Links per Node	1	1
Output Links per Node	0	0
Local Communication Startup [us]	No Startup	0.00
Remote Communication Startup [us]	50 us	50.00
Relative Processor Speed [%]	Automatic	1.00
Local Bandwidth [Mbyte/s]	Instantaneous	0.000000

dimemas@cepba.upc.es

Node definition

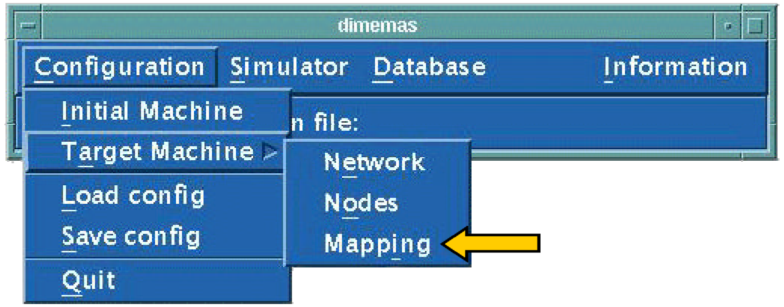
- **Node Number**
  - Node to which the parameters apply
- **Node architecture (optional)**
  - If used, loads parameters from machine database
- **Number of processors**
  - Size of SMP node
- **Input/output links**
  - Maximum number of simultaneously incoming/outgoing messages to/from the node. Simple model of node injection mechanisms.
  - To specify half duplex links, input or output must be zero
- **Local communication startup**
  - Latency used for messages within the node

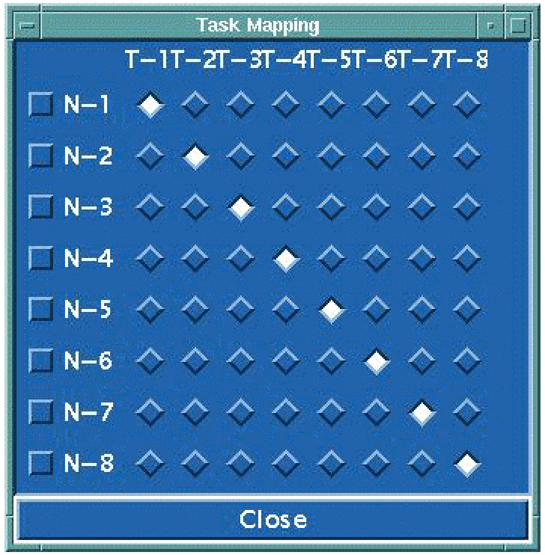
Node definition

- **Remote communication startup**
  - Latency used for messages through the network
- **Relative processor speed**
  - Processor speed ratio between the target processor and the processor where the trace was obtained
- **Local bandwidth**
  - Bandwidth for communication within SMP
- **Click the Apply button to confirm the input for the current node**
- **Click the Do all same button to copy the confirmed values of the current node to remaining nodes**

Process mapping

- Define the process– to– node mapping



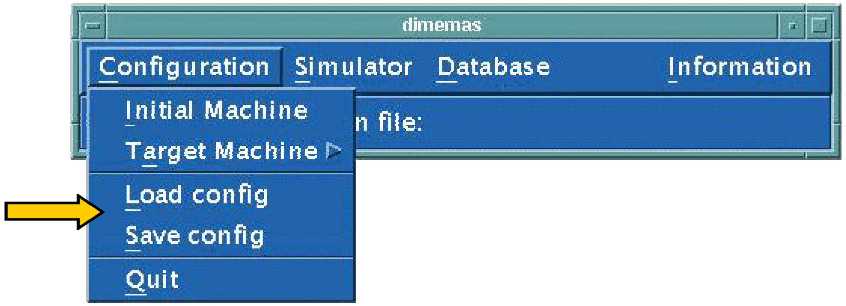

Task Mapping

	T–1	T–2	T–3	T–4	T–5	T–6	T–7	T–8
□ N–1	◆	◇	◇	◇	◇	◇	◇	◇
□ N–2	◇	◆	◇	◇	◇	◇	◇	◇
□ N–3	◇	◇	◆	◇	◇	◇	◇	◇
□ N–4	◇	◇	◇	◆	◇	◇	◇	◇
□ N–5	◇	◇	◇	◇	◆	◇	◇	◇
□ N–6	◇	◇	◇	◇	◇	◆	◇	◇
□ N–7	◇	◇	◇	◇	◇	◇	◆	◇
□ N–8	◇	◇	◇	◇	◇	◇	◇	◆

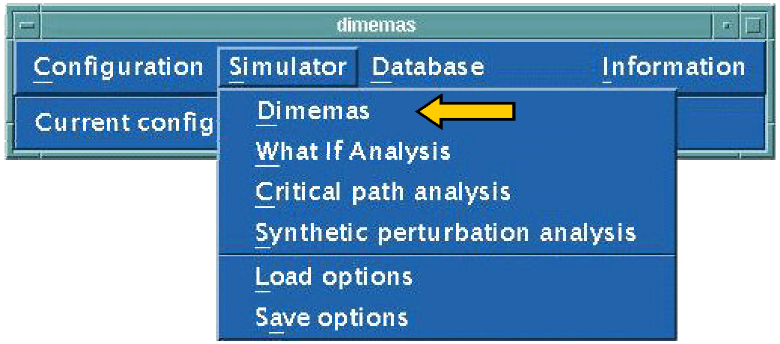
- Click the button close to Node number, to open the node definition dialog

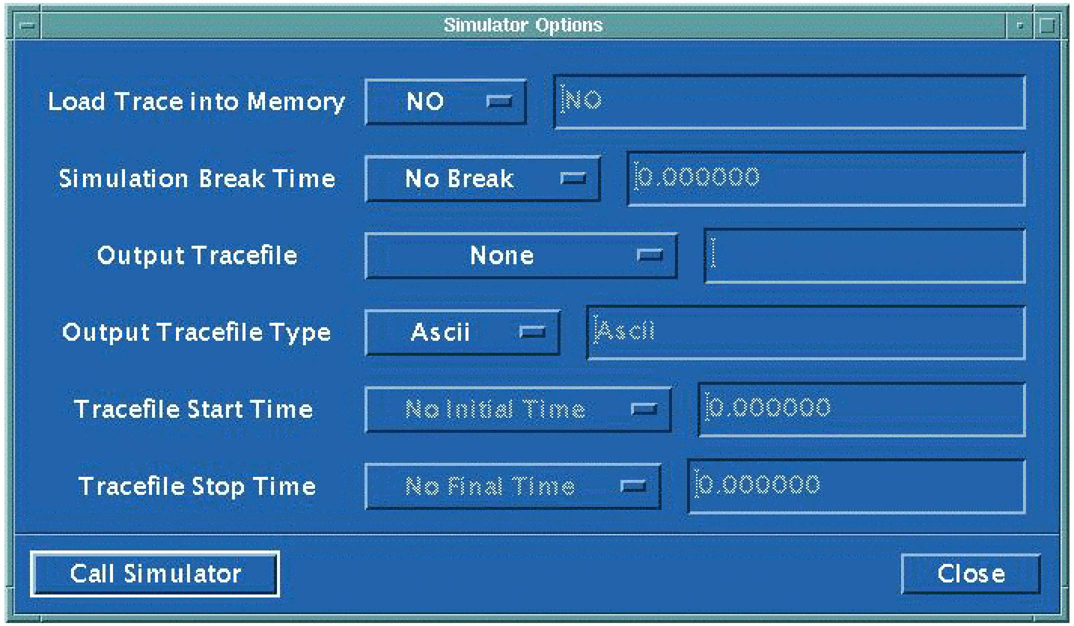
Configuration file

- **File storing the initial and target model parameters**
  - Save the initial and target machine definitions with the menu function
  - Load configuration files with the menu function



Simulator execution




Simulator Options

Load Trace into Memory: NO — NO

Simulation Break Time: No Break — 0.000000

Output Tracefile: None

Output Tracefile Type: Ascii — Ascii

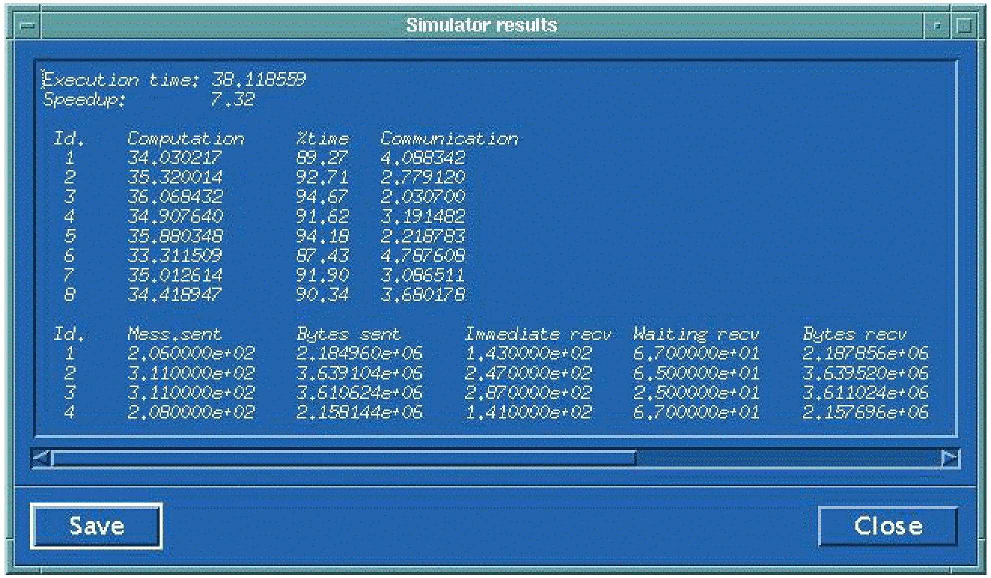
Tracefile Start Time: No Initial Time — 0.000000

Tracefile Stop Time: No Final Time — 0.000000

- Once the initial machine and the target machine have been defined, the simulator can be used
- Click on the Call Simulator button to run the simulator

Simulator results

- The simulator will display the performance results



```
Execution time: 38.118559
Speedup:        7.32

 Id.    Computation      %time   Communication
  1     34.030217        89.27   4.088342
  2     35.320014        92.71   2.779120
  3     36.068432        94.67   2.030700
  4     34.907640        91.62   3.191482
  5     35.880348        94.18   2.218783
  6     33.311509        87.43   4.787608
  7     35.012614        91.90   3.086511
  8     34.418947        90.34   3.680178

 Id.    Mess.sent        Bytes sent       Immediate recv   Waiting recv     Bytes recv
  1     2.060000e+02     2.184960e+06     1.430000e+02     6.700000e+01     2.187856e+06
  2     3.110000e+02     3.639104e+06     2.470000e+02     6.500000e+01     3.639520e+06
  3     3.110000e+02     3.610624e+06     2.870000e+02     2.500000e+01     3.611024e+06
  4     2.080000e+02     2.158144e+06     1.410000e+02     6.700000e+01     2.157696e+06
```

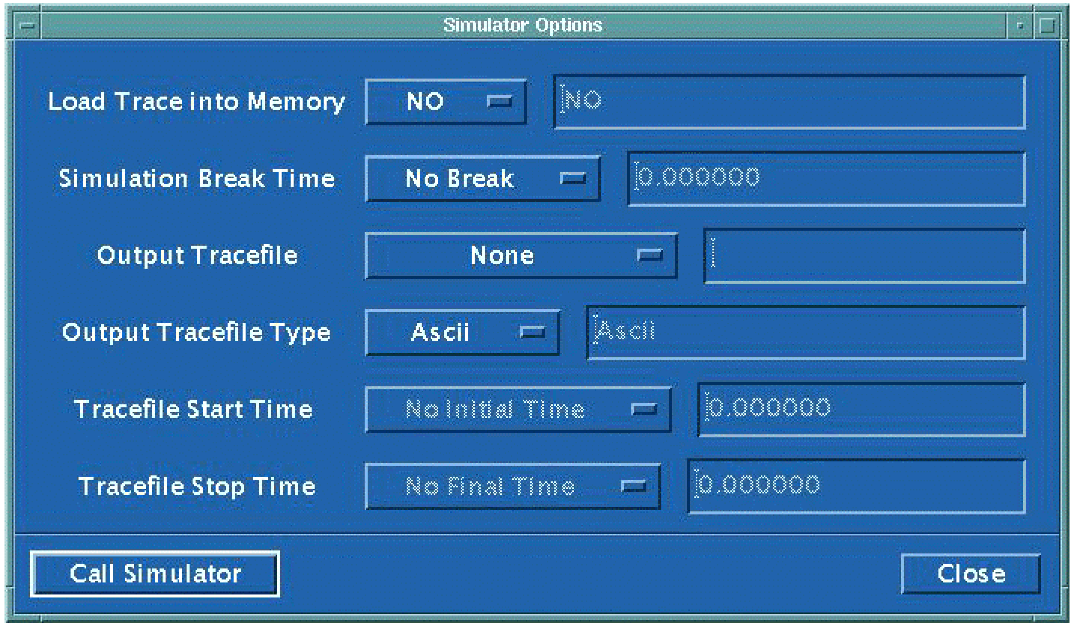
- Click on the Save button to save the simulation results to disk (the configuration file is also saved)

Simulator results

- **Global statistics:**
  - execution time: Modeled elapsed time in seconds
  - speedup: Total CPU time / execution time
- **Per– process statistics:**
  - computation time: absolute time in seconds and percentage
  - communication time: absolute time in seconds

  - Sent: number of messages and number of bytes sent
  - Received: number of messages received without blocking (the message had arrived before the reception request), number of messages received with blocking and number of bytes received
  - Group operations: number and total size

Simulation parameters



Load Trace into Memory	NO	NO
Simulation Break Time	No Break	0.000000
Output Tracefile	None	
Output Tracefile Type	Ascii	Ascii
Tracefile Start Time	No Initial Time	0.000000
Tracefile Stop Time	No Final Time	0.000000

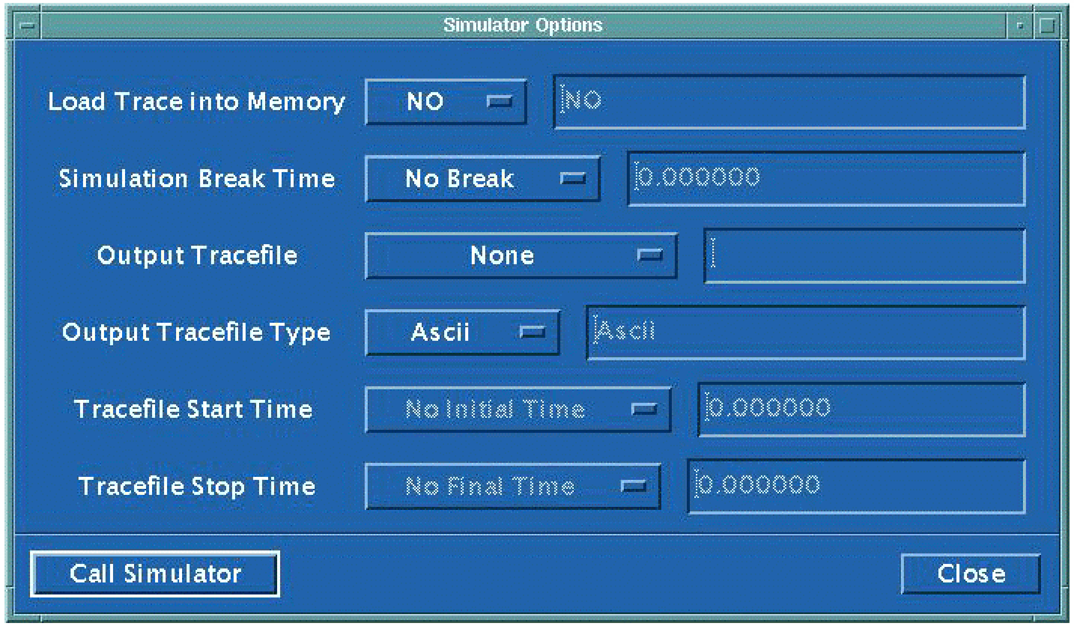
Load trace into memory

- If set, speeds up the simulation. May cause swapping with large trace files.

Break time

- Define the final simulation time

Simulation parameters



Simulator Options

Load Trace into Memory NO NO

Simulation Break Time No Break 0.000000

Output Tracefile None

Output Tracefile Type Ascii Ascii

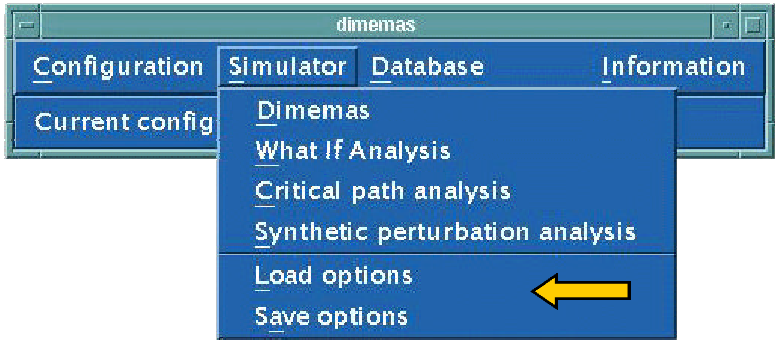
Tracefile Start Time No Initial Time 0.000000

Tracefile Stop Time No Final Time 0.000000

Output tracefile (visualization tracefile)

- None, Paraver and Vampir (if license available)
- Tracefile type: ASCII or Binary
- Start time and stop time
  - ✓ Specify the initial time and final time for the visualization tracefile
  - ✓ Messages starting before “start time” are delivered at time 0
  - ✓ Messages ending after “stop time” are delivered at stop time

Simulator options



- **Save execution options for Dimemas (command line options to the simulator)**
- **Load previously saved execution options**

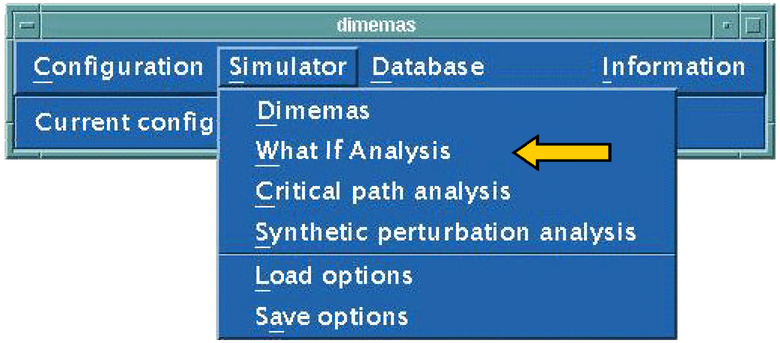
Methodology

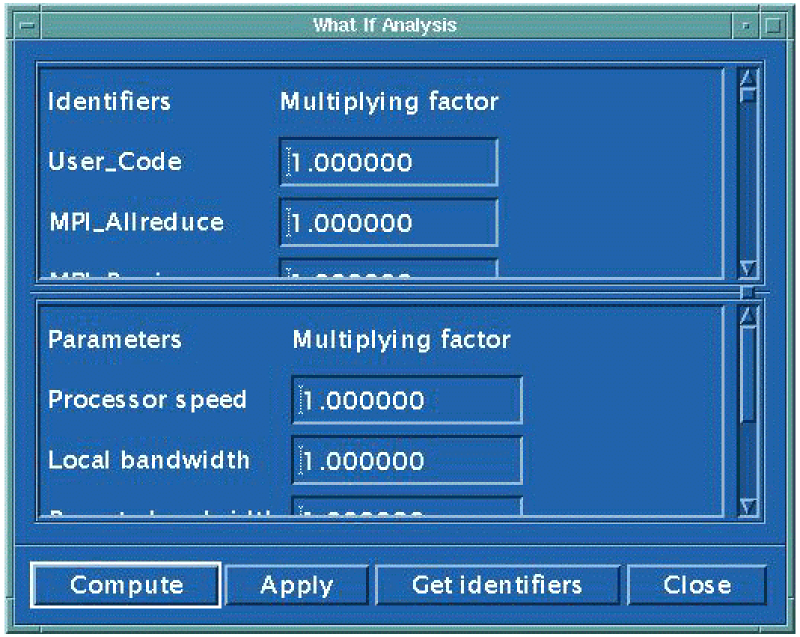
- **Initial studies that may help understanding the application characteristics**
  - Does the application have load balanced and dependence problems?
    - ✓ **BW = ∞, L = 0**
  - Would we benefit from grouping messages?
    - ✓ **L = … , BW = ∞**
  - Is bandwidth the problem?
    - ✓ **BW = … , L = 0**
  - Is network contention the problem?
    - ✓ **BW = target , L = target , Buses = 1, 2, …**

Methodology

- Perform some of the above extreme simulations generating Paraver traces. Visualize the traces to perceive the general behavior and on which parts of the application the different parameters are more relevant.
- Perform parametric studies (many simulations without generating traces, just reporting the execution time) to quantify the influence of the different parameters.
- Iterate the process going from very large dynamic range of the parameters to points closer to the estimated operation range of the target machine. If the visualization identifies that problems are only on a specific trace section, you can generate with Paraver a tracefile of only that section. In this way the simulation will be faster and the parametric studies will reflect the effect of the parameters on the section of interest.
- **The following tools within Dimemas may be helpful in this analysis**

What if analysis





Identifiers	Multiplying factor
User_Code	1.000000
MPI_Allreduce	1.000000

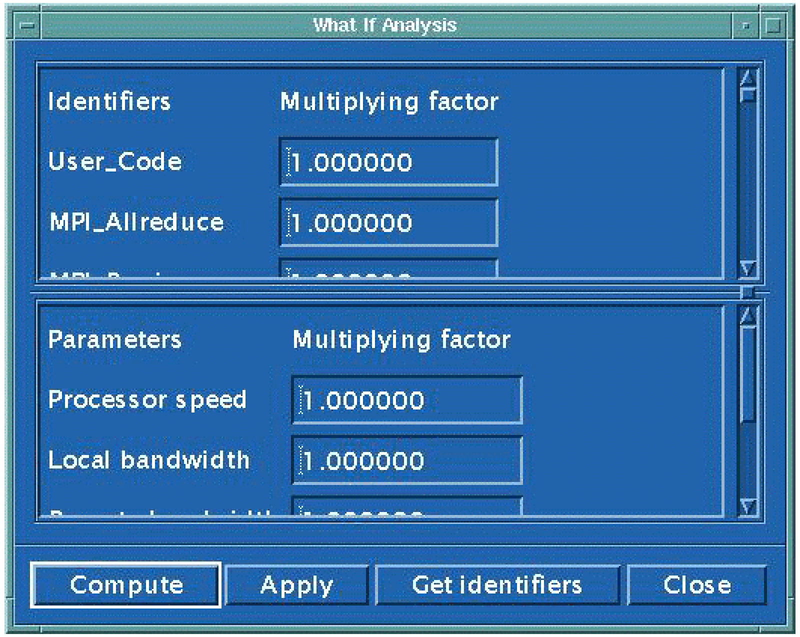
Parameters	Multiplying factor
Processor speed	1.000000
Local bandwidth	1.000000

- Analyze the behavior of the application under relative parameter modifications

What if analysis

- **Upper area: subroutines/block names**
  - The multiplying factor applies to de CPU demand within the subroutine or block.
  - Examples
    - ✓ a value of 0.75 for a given routine could be used to estimate the effect of sequential code optimizations that speed up the computation within that routine by 25%.
    - ✓ A value of 0 for a given routine can be used to estimate the effect on the global performance of totally eliminating that routine.
- **Lower area: architectural parameters**
  - The multiplying factor applies to the parameter as defined in the target mode definition section.
  - Examples:
    - ✓ a value of 2 in relative processor speed could be used to estimate the effect on the execution time of a CPU twice as fast as the one defined by the value specified in the target node definition section
    - ✓ a value of 0 in the network latency could be used to estimate the effect of an ideal (null) network latency

What if analysis

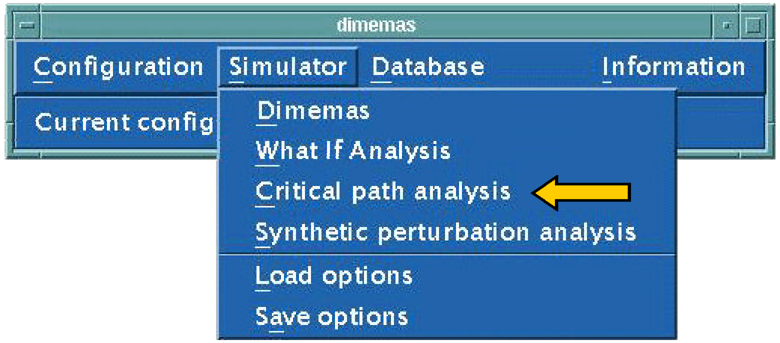


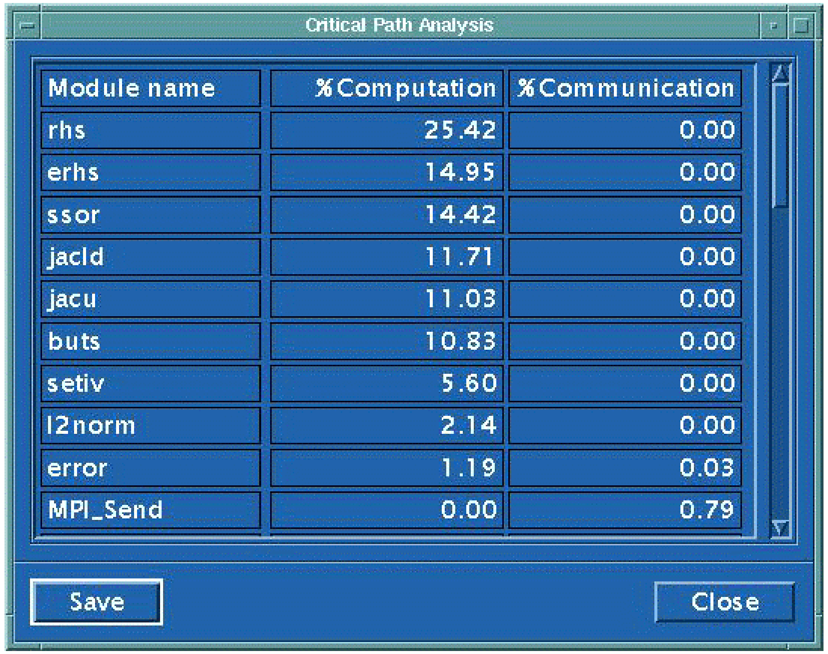
Identifiers	Multiplying factor
User_Code	1.000000
MPI_Allreduce	1.000000

Parameters	Multiplying factor
Processor speed	1.000000
Local bandwidth	1.000000

- Click on the Compute button to predict effects
- Click on the Apply button to save this changes to the current configuration
- Click on the Get Identifiers button to analyze the tracefile to get the application identifiers

Critical Path analysis



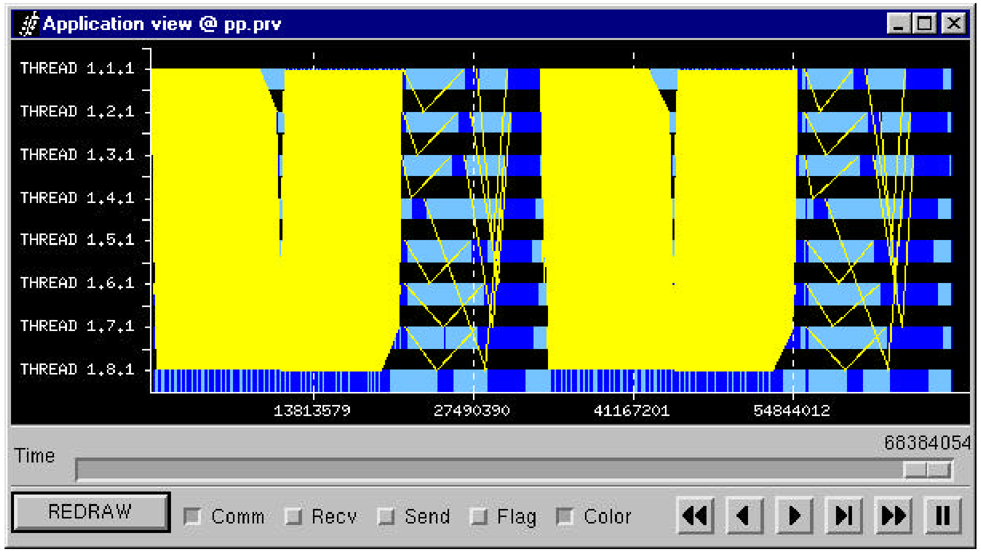


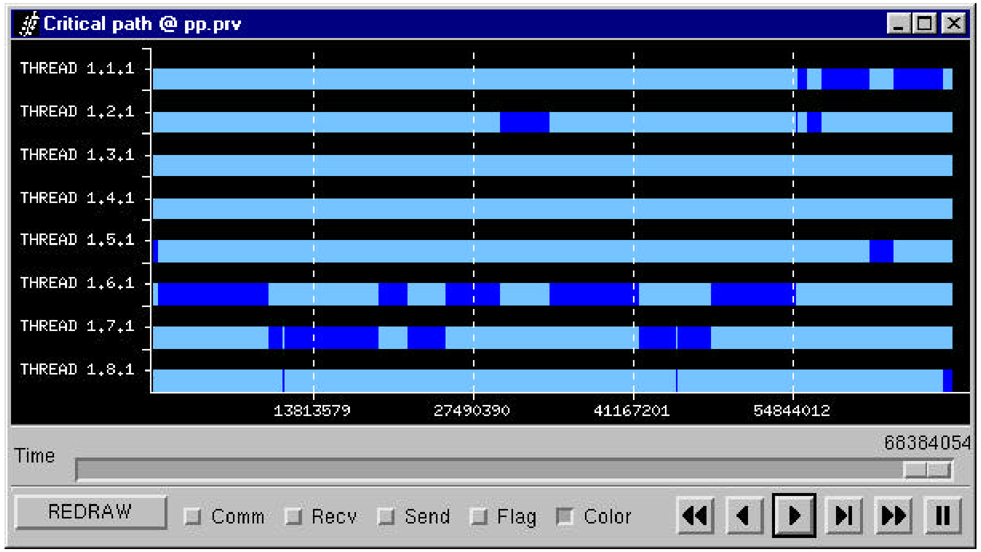
Module name	%Computation	%Communication
rhs	25.42	0.00
erhs	14.95	0.00
ssor	14.42	0.00
jacld	11.71	0.00
jacu	11.03	0.00
buts	10.83	0.00
setiv	5.60	0.00
l2norm	2.14	0.00
error	1.19	0.03
MPI_Send	0.00	0.79

- Percent computing and computation time on critical path
- Click on the **Save** button to save the critical path to disk
- **Paraver** is able to show you the Critical Path

Critical Path analysis

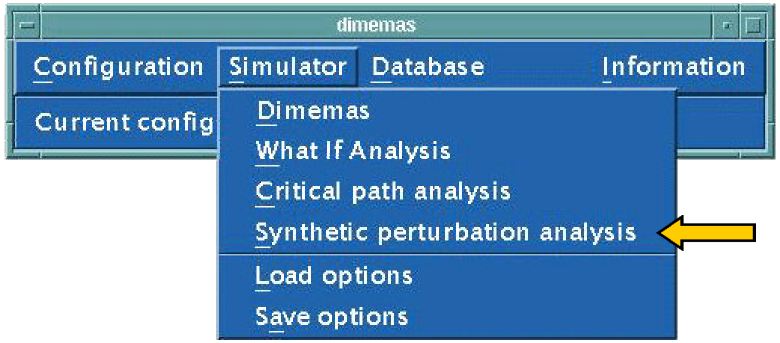
- **Visualization of CP using Paraver for LU, class B, 2 iterations, 8 tasks, contention network**

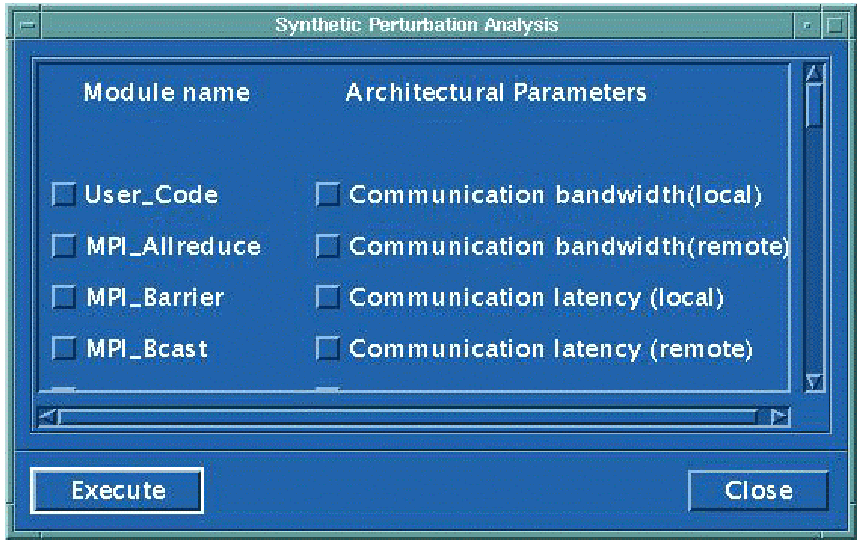




Synthetic Perturbation Analysis

- **Full factorial analysis**





Module name	Architectural Parameters
□ User_Code	□ Communication bandwidth(local)
□ MPI_Allreduce	□ Communication bandwidth(remote)
□ MPI_Barrier	□ Communication latency (local)
□ MPI_Bcast	□ Communication latency (remote)

- **Select “interesting” factors**
  - routine sequential computation demands
  - architectural parameters
- **Click on Execute button to compute model**

Synthetic Perturbation Analysis

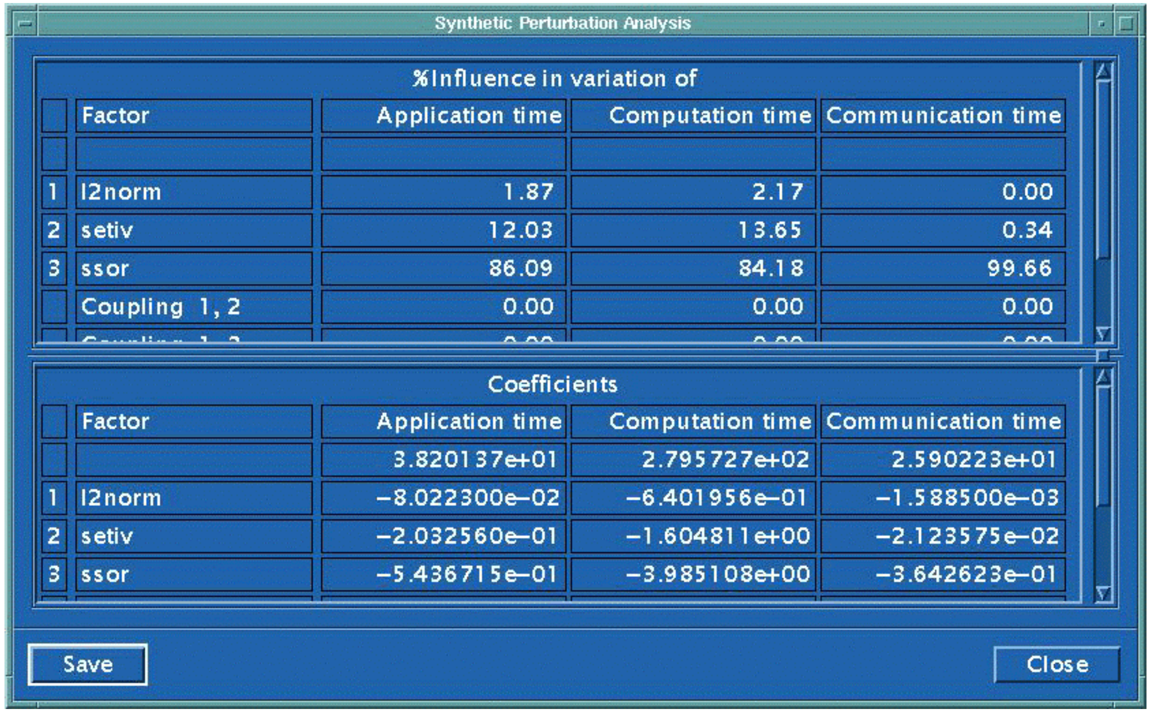
- **Model**

$$Time = T_0 + T_1 * F_1 + T_2 * F_2 + T_{12} * F_1 * F_2$$

- **T_0 is the model execution time for the nominal parameter set up.**
- **F_i represents the percentage variation of the ith parameter value.**
  - F=1 represents a +10% variation of the parameter
  - F=-1 represents a -10%
- **V_i is the coefficient for factor i**
- **V_{ij} are coupled coefficient for factors i and j**
- **SPA displays**
  - Influence of each factor: Percentage of total variability explained by each factor
  - Model coefficients

Synthetic Perturbation Analysis

- Example output



Synthetic Perturbation Analysis

%Influence in variation of

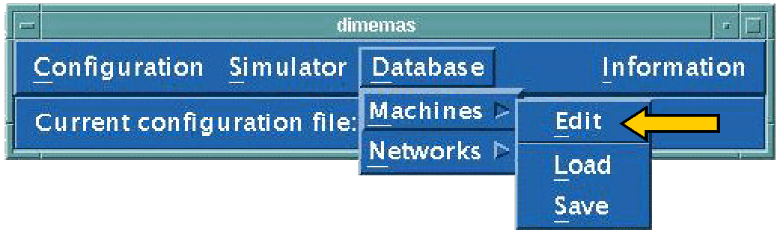
	Factor	Application time	Computation time	Communication time
1	l2norm	1.87	2.17	0.00
2	setiv	12.03	13.65	0.34
3	ssor	86.09	84.18	99.66
	Coupling 1, 2	0.00	0.00	0.00

Coefficients

	Factor	Application time	Computation time	Communication time
		3.820137e+01	2.795727e+02	2.590223e+01
1	l2norm	−8.022300e−02	−6.401956e−01	−1.588500e−03
2	setiv	−2.032560e−01	−1.604811e+00	−2.123575e−02
3	ssor	−5.436715e−01	−3.985108e+00	−3.642623e−01

- ssor is the most influent routine (more than 85%)
- Comparing the coefficients with T_0 exposes the actual magnitude of the variation (2 orders of magnitude difference in this example)

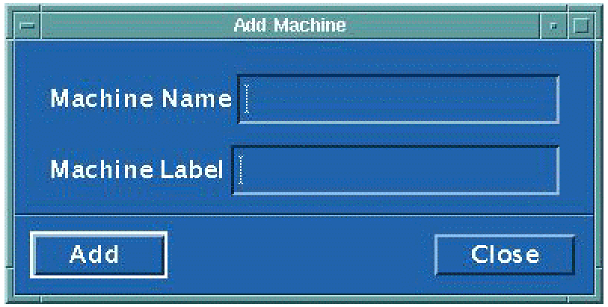
Machine database



- To maintain the information of the machines
- To define the ratio speed between to machines in DB
- Information to automatically complete fields in the GUI
- Load and Save database to disk

Machine database

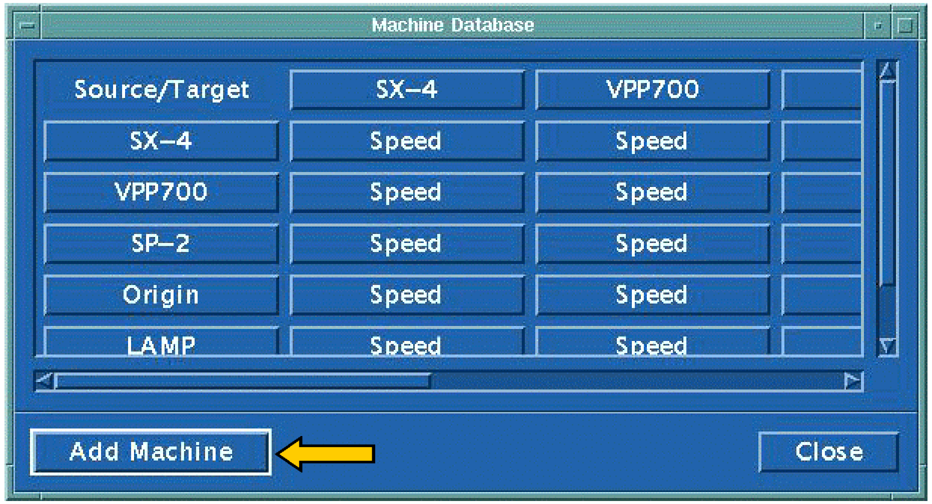
- **New machine**



Add Machine

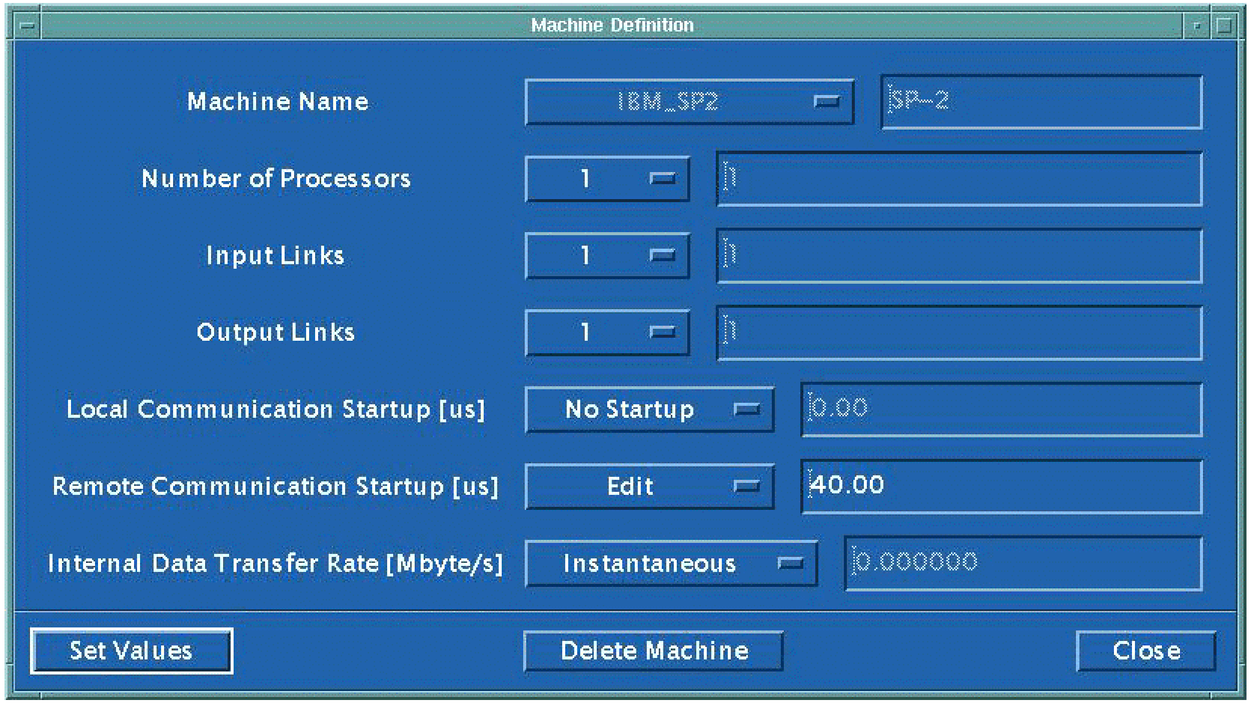
Machine Name

Machine Label



Machine Database

Source/Target	SX–4	VPP700	
SX–4	Speed	Speed	
VPP700	Speed	Speed	
SP–2	Speed	Speed	
Origin	Speed	Speed	
LAMP	Speed	Speed	

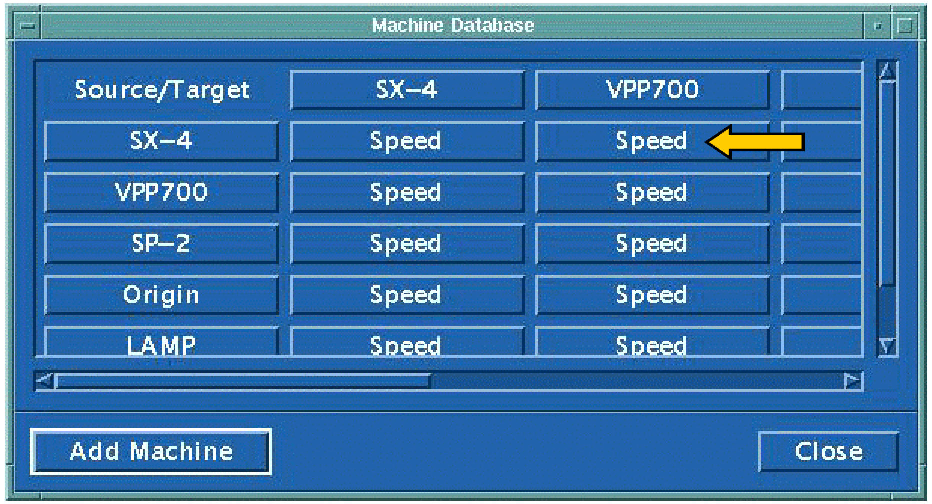


Machine Definition

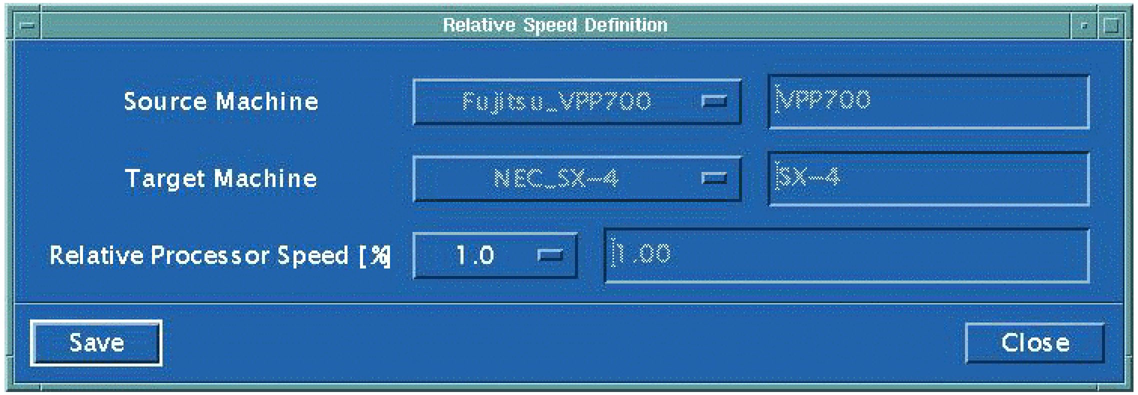
Machine Name	IBM_SP2	SP–2
Number of Processors	1	1
Input Links	1	1
Output Links	1	1
Local Communication Startup [us]	No Startup	0.00
Remote Communication Startup [us]	Edit	40.00
Internal Data Transfer Rate [Mbyte/s]	Instantaneous	0.000000

Machine database

- **Modify relative processor speed**



Source/Target	SX-4	VPP700	
SX-4	Speed	Speed	
VPP700	Speed	Speed	
SP-2	Speed	Speed	
Origin	Speed	Speed	
LAMP	Speed	Speed	

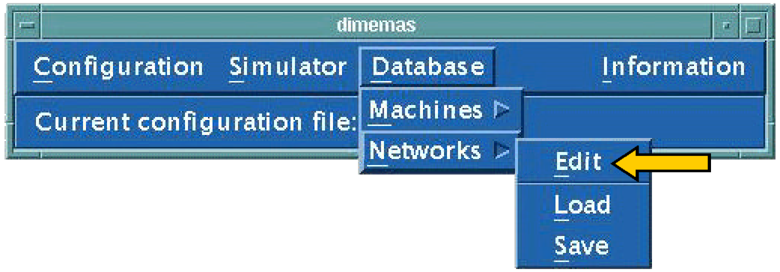


Source Machine: Fujitsu_VPP700 — VPP700

Target Machine: NEC_SX-4 — SX-4

Relative Processor Speed [%]: 1.0 — 1.00

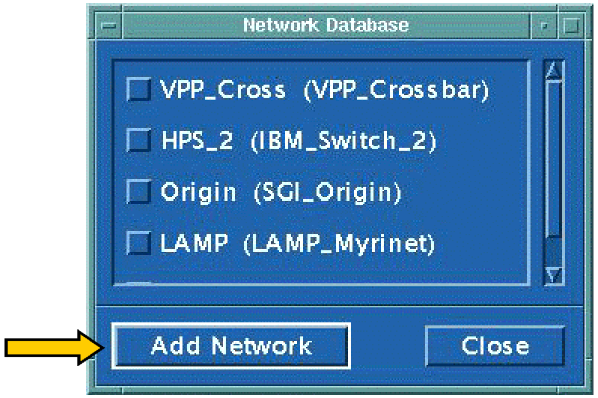
Network database



- To maintain the information of the networks
- Information to automatically complete fields in the GUI
- Load and Save database to disk

Network database

- New network

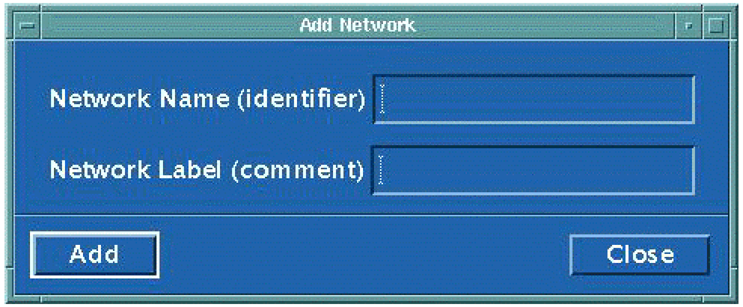


□ VPP_Cross (VPP_Crossbar)

□ HPS_2 (IBM_Switch_2)

□ Origin (SGI_Origin)

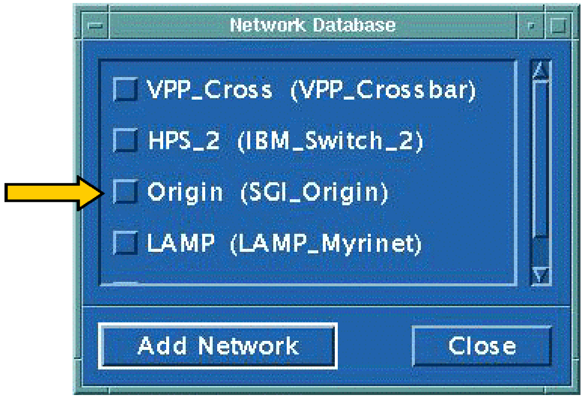
□ LAMP (LAMP_Myrinet)



Network Name (identifier)	
Network Label (comment)	

Network database

- **Network parameters modification**



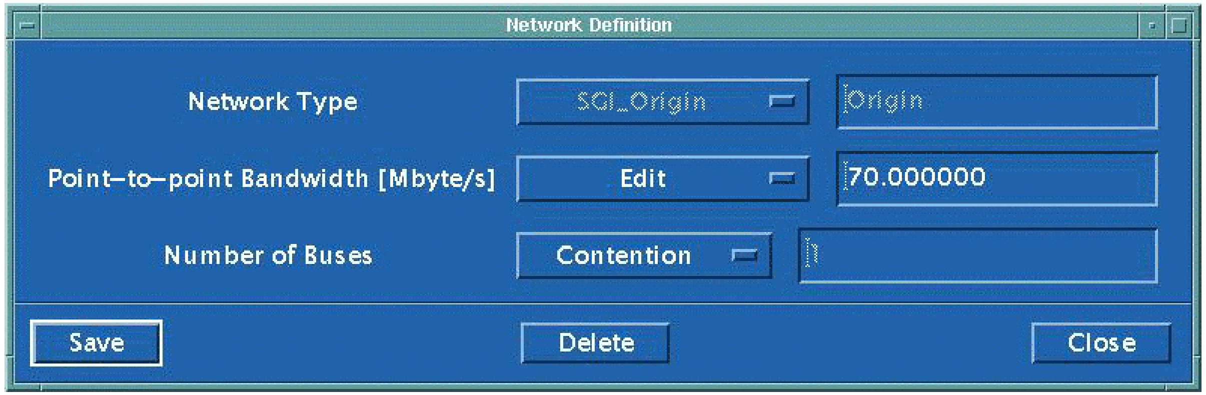
Network Database

□ VPP_Cross (VPP_Crossbar)

□ HPS_2 (IBM_Switch_2)

□ Origin (SGI_Origin)

□ LAMP (LAMP_Myrinet)



Network Definition

Network Type	SGI_Origin	Origin
Point–to–point Bandwidth [Mbyte/s]	Edit	70.000000
Number of Buses	Contention	1

More information

http://www.cepba.upc.es/dimemas

dimemas@cepba.upc.es