

Distributed Real-Time Computing with Harness

C. Di Biagio - G. Pennella – MBDA SW AR&T
E. Di Saverio – M. Cesati – University of Rome “Tor Vergata”
C. Engelmann - Oak Ridge National Laboratory

14th EUROPEAN PVM MPI Users’ Group Meeting
Paris (France)



University of Rome “Tor Vergata”
Information Engineering Dept.



Oak Ridge
National
Laboratory

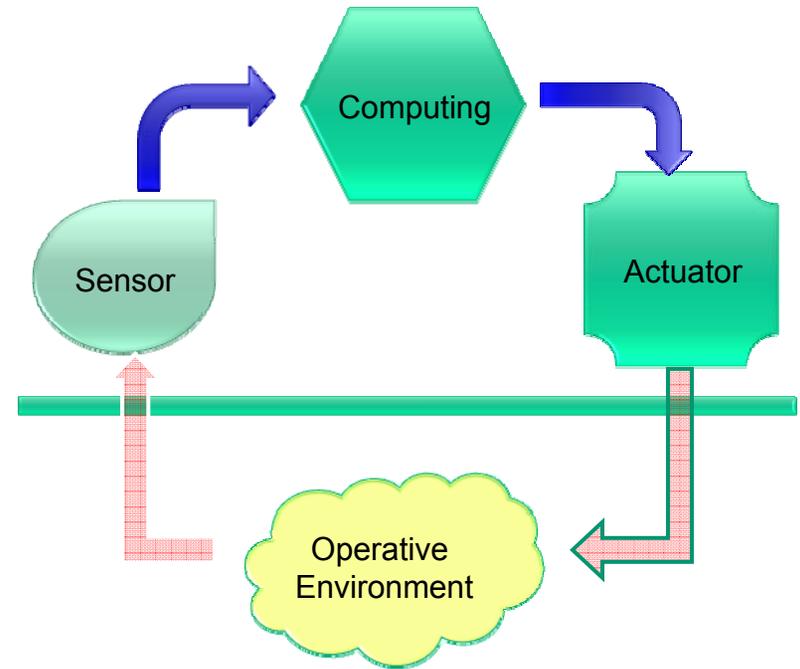
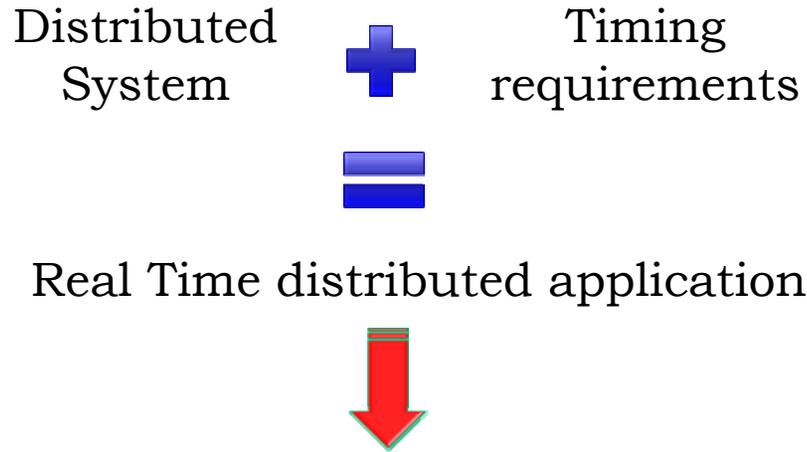
MBDA
Software Applied Research &
Technology Department

- Context
- Goal
- Harness overview
- Harness and the RT issue
- Proposed solutions
- Conclusions & Future Work



- Within industrial applications, the requirements that needs to be addressed are:
 - *Safety Criticalness*
 - *Real Time – usually Hard Real Time*
 - *Embeddability*
 - *Heterogeneity – in the case of a distributed system*
 - *High Availability*
- In the case of a heterogeneous, distributed system, the main communication paradigm is message passing.

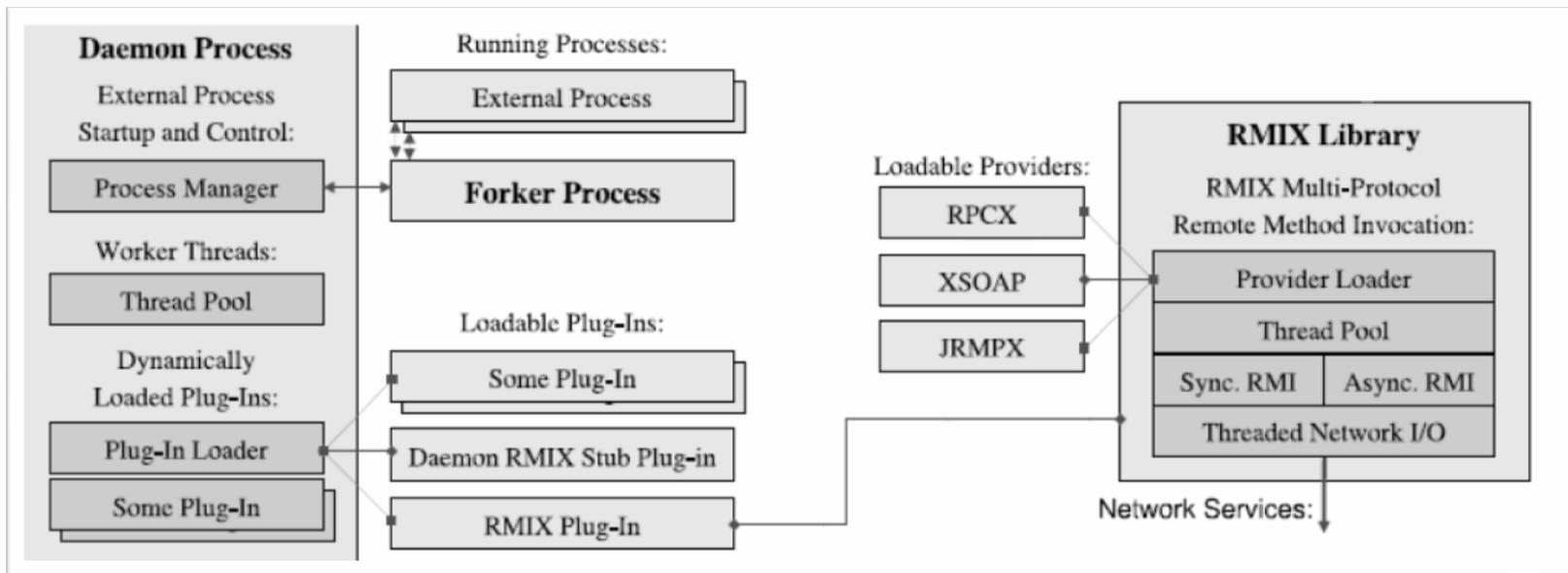




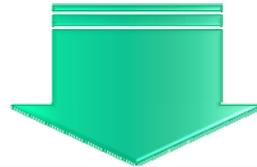
Key element: a middleware able to support RT and efficient communications



- In the past, excellent results were achieved using PVM.
- We decided to move towards HARNESS.
 - *“The heterogeneous adaptable reconfigurable networked systems (Harness) research project focused on the design and development of a pluggable lightweight heterogeneous Distributed Virtual Machine (DVM) environment.”*



- The version we take into account is the “C” one (no JAVA for such systems !).
- Very promising elements ... but lack of specific RT features:
 - *Priorities management of the DVM jobs*
 - *RMIX Provider Plugin based on TCP*
 - *Lack of event logging facility*



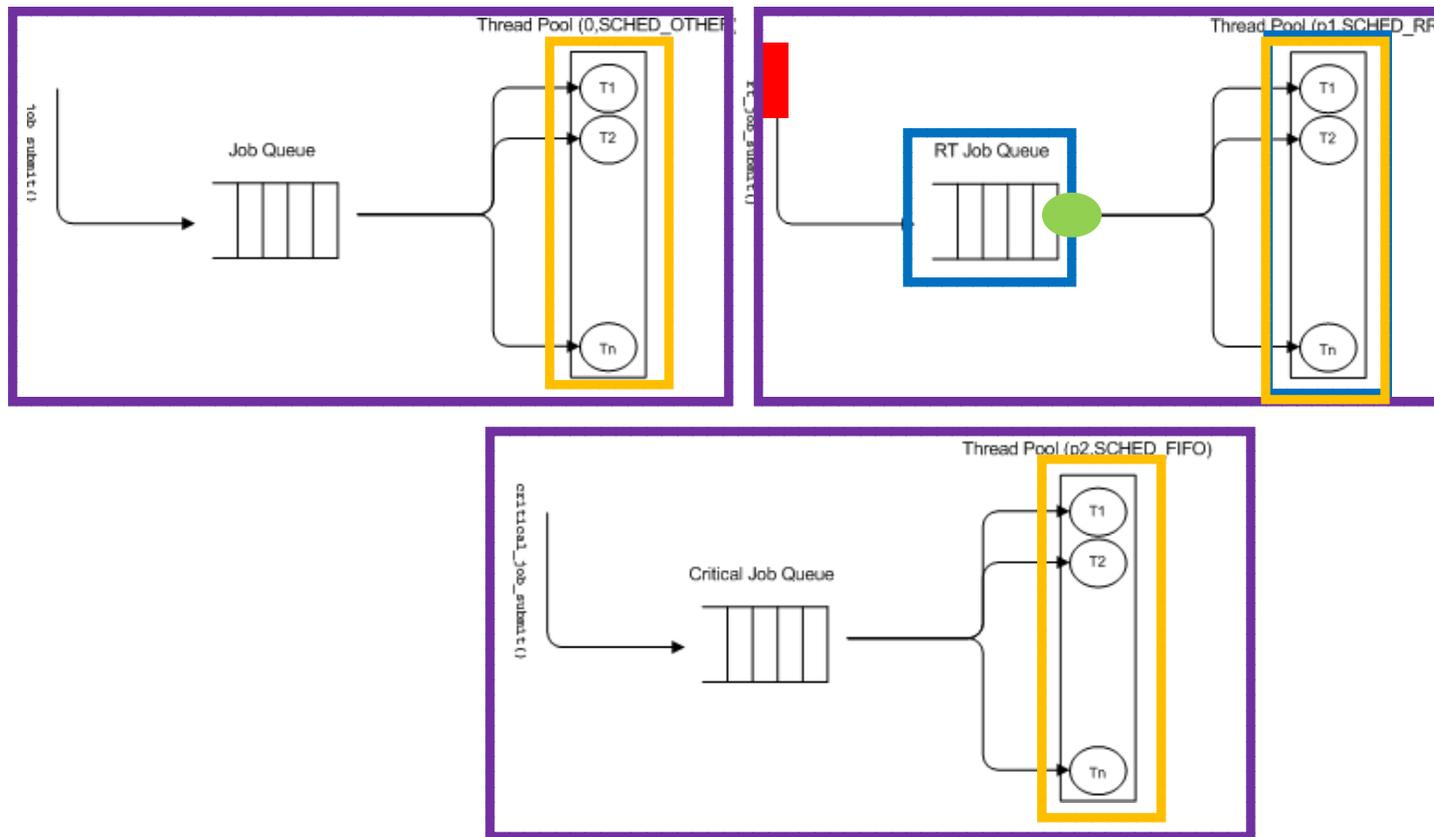
Harness RT improvements

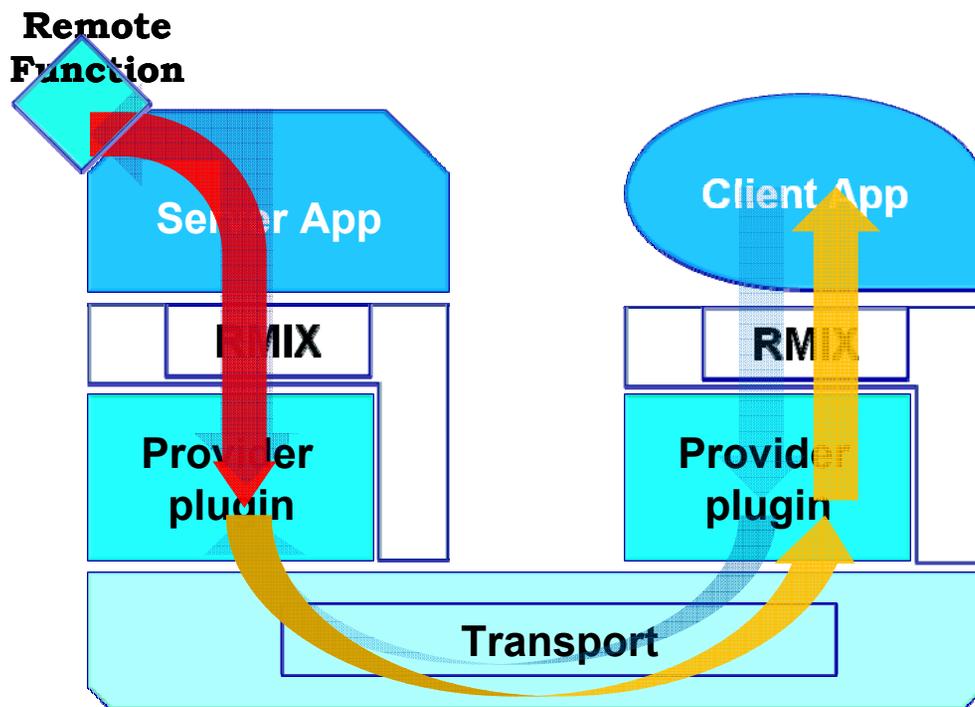
Development of a set of plugins to overcome the seen elements to make Harness *best fit for Real Time industrial applications*



- The first plugin developed support job management in RT environment.
- The approach followed is a QoS multi job queue:
 - *Is now possible to define different “class-of-services” for the execution of user jobs.*
 - *For every class there is a job queue and for every queue there is a pre-allocated thread pool – thus no thread creation/detach/destroy time wasting activities are done.*
 - *Moreover, each class has its own priority (i.e.: the threads of the class have a specific priority) and scheduling policy (like FIFO or RR).*
- In order to exploit the new RT features, a proper RTOS shall be used.
 - *“Our” Linux Hard Real Time kernel fully support this approach*







Functions

- ❖ Export Function
- ❖ Send - Async
- ❖ Invoke - Sync
- ❖ Oneway - No return

rmix-rtrpcx new provider plugin

Low latency UDP Transport Layer

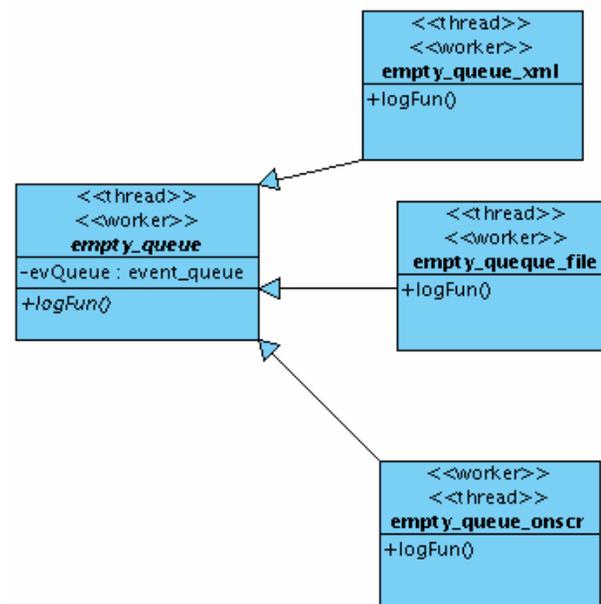
Support rthreads

Preallocated Memory Buffers



Event Log Service

- ❖ Timestamp: 10 usec
- ❖ Event creation task → High prio
- ❖ Writing log → Low prio
- ❖ 3 Output formats

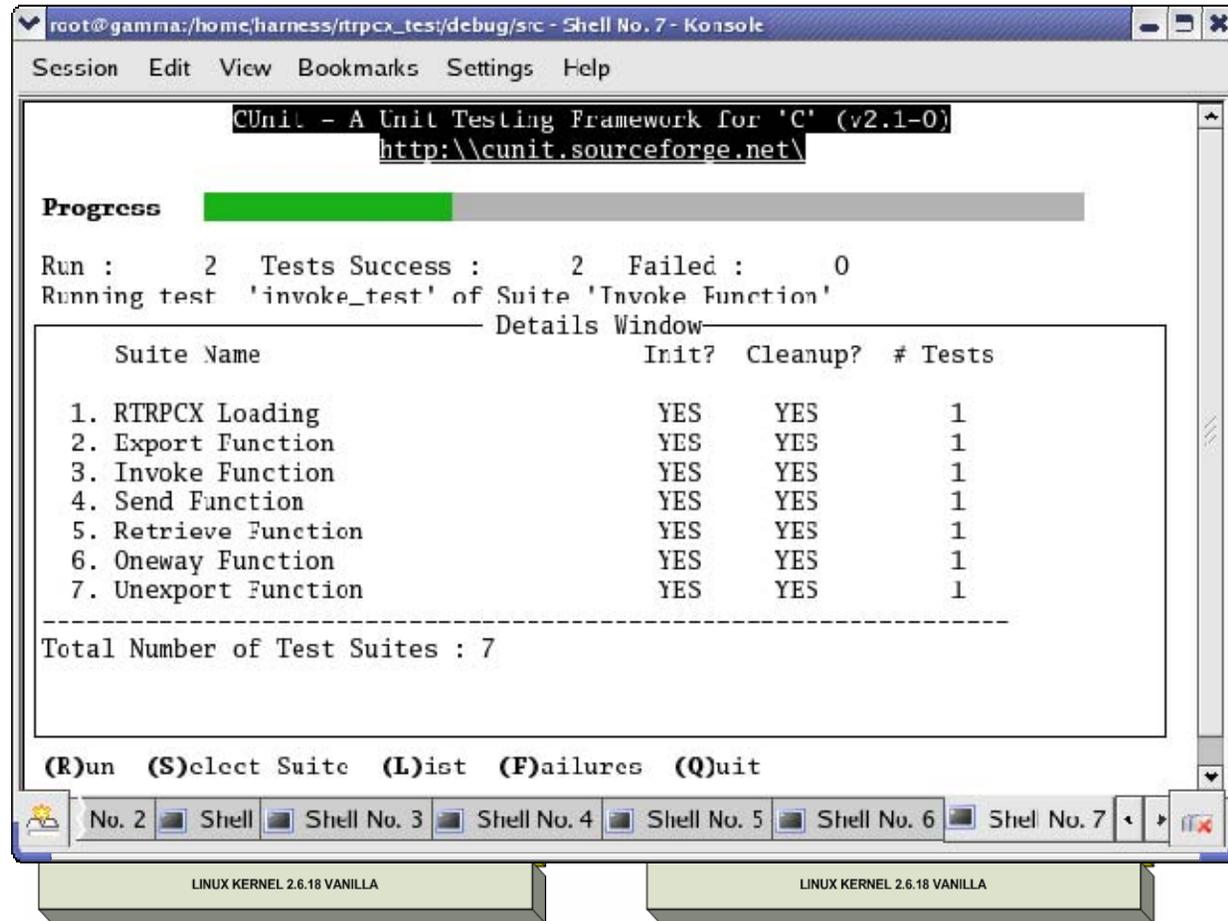


Event Log - jerome @ ercole , started at 07/04/10 - 14:37:04

Time	Source	Description
[GMT 07/04/10 - 14:37:01] 752466000 nsec	Ev_Gen_by-pilot.c:246 system_init	System Started and Running
[GMT 07/04/10 - 14:37:04] 643995000 nsec	Ev_Gen_by-pilot.c:250 handle_signal	Signal Received
[GMT 07/04/10 - 14:37:04] 768972000 nsec	Ev_Gen_by-pilot.c:255 signal_feedback	Signal Processed
[GMT 07/04/10 - 14:37:10] 893045000 nsec	Ev_Gen_by-pilot.c:259 wait	Enter Waiting State
[GMT 07/04/10 - 14:37:17] 152092000 nsec	Ev_Gen_by-pilot.c:267 process_alert	ALERT Signal For: Thermal Gauge
[GMT 07/04/10 - 14:37:17] 17113000 nsec	Ev_Gen_by-pilot.c:263 handle_signal	Signal Received
[GMT 07/04/10 - 14:37:17] 187082000 nsec	Ev_Gen_by-pilot.c:271 alert_feedback	Raise ALARM: Flashing Light
[GMT 07/04/10 - 14:37:31] 621894000 nsec	Ev_Gen_by-pilot.c:275 system_fini	System Finalizing



- All the seen plugins have been tested
- Unit tests
- Functional tests
- Benchmarks



The screenshot shows a terminal window titled "root@gamma:/home/harness/rtrpcx_test/debug/src - Shell No. 7 - Konsole". The window displays the output of a CUnit test run. At the top, it says "CUnit - A Unit Testing Framework for 'C' (v2.1-0)" and provides the URL "http://cunit.sourceforge.net/". Below this, there is a progress bar and the following statistics: "Run : 2 Tests Success : 2 Failed : 0". The test being run is "invoke_test" of Suite "Invoke Function". A "Details Window" is open, showing a table of test suites:

Suite Name	Init?	Cleanup?	# Tests
1. RTRPCX Loading	YES	YES	1
2. Export Function	YES	YES	1
3. Invoke Function	YES	YES	1
4. Send Function	YES	YES	1
5. Retrieve Function	YES	YES	1
6. Oneway Function	YES	YES	1
7. Unexport Function	YES	YES	1

Below the table, it says "Total Number of Test Suites : 7". At the bottom of the terminal, there are keyboard shortcuts: "(R)un (S)elect Suite (L)ist (F)ailures (Q)uit". The terminal window is running on a Linux kernel 2.6.18 vanilla.

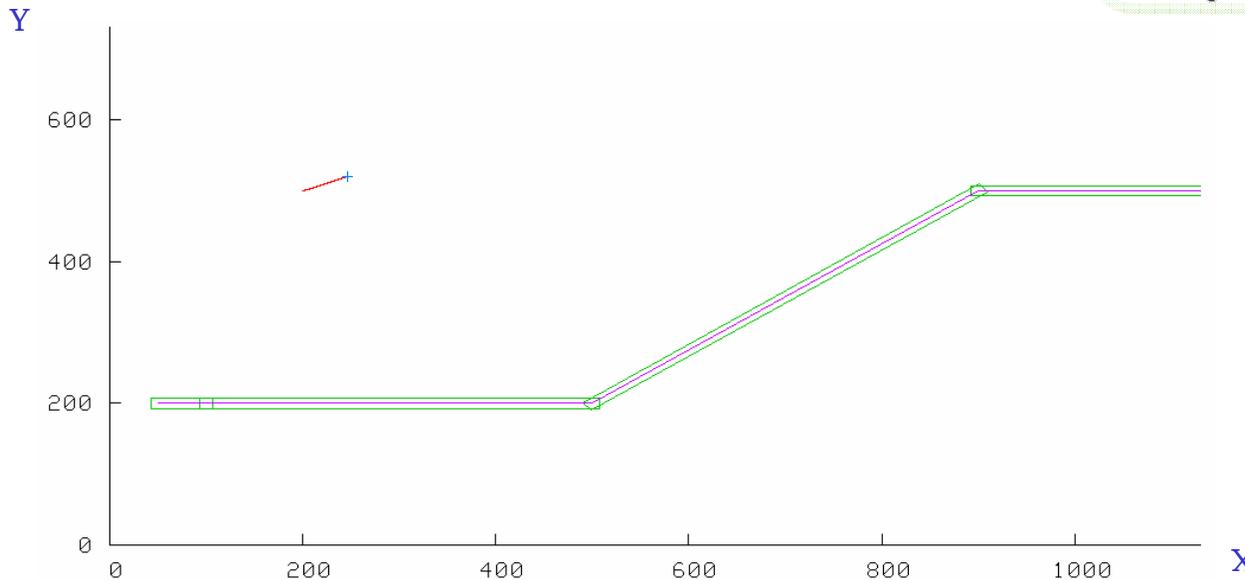
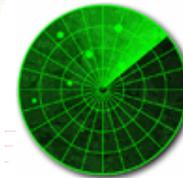


**Application: 2D
control of a vector in
space to manouver
to to a specific route**



T1

T2

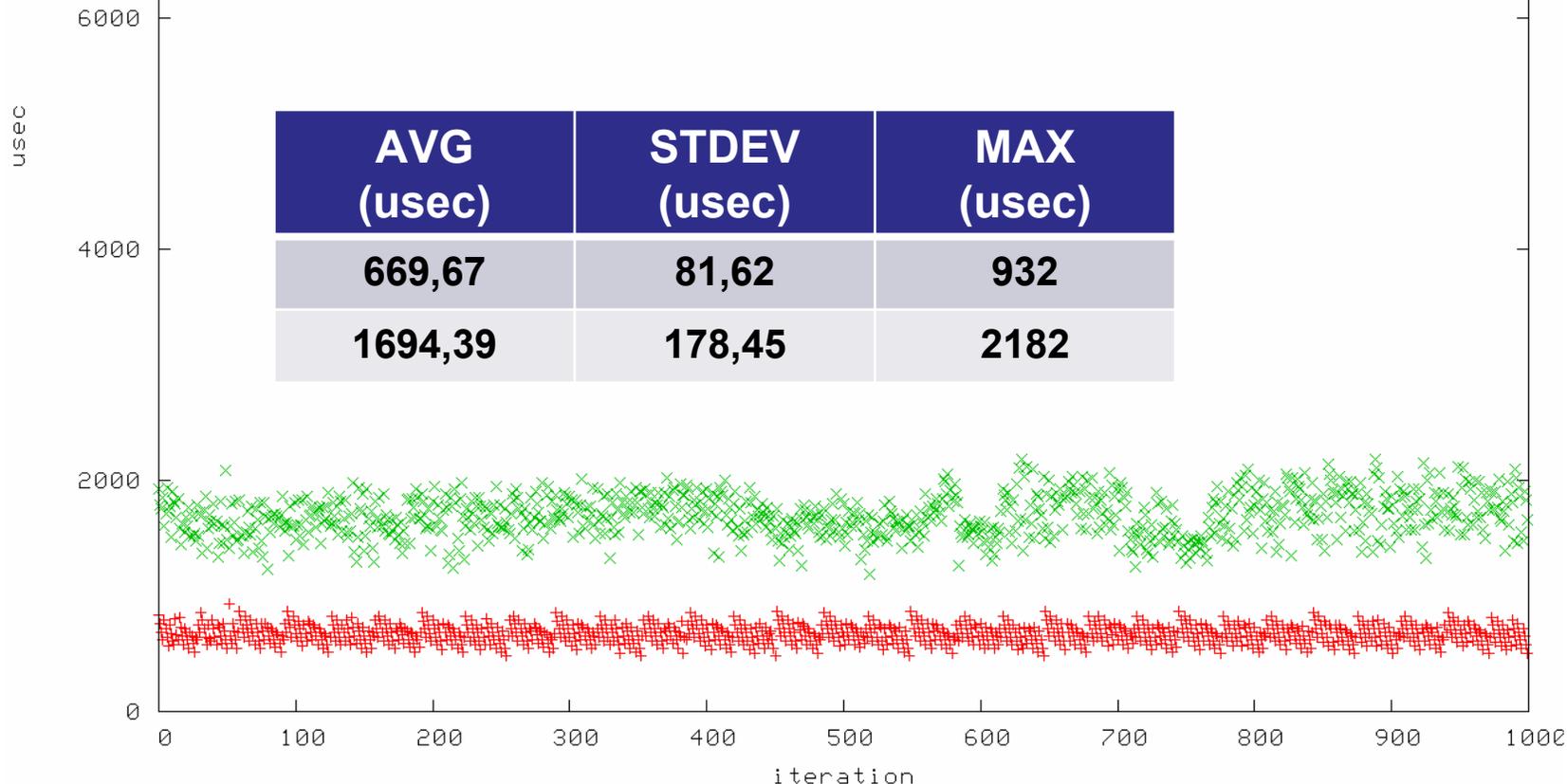


**Round Trip
Time
 $RTT = T2 - T1$**

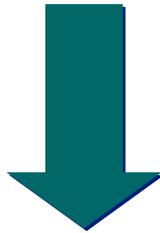


Carico: Unix Benchmark

'scarico/NEWSpd' +
'ubench-c-m/NEWSpd' x



- A new set of Harness plugins has been developed and tested.
- With these plugins, Harness (C-based version) is now able to fully support Industrial Distributed Real Time environment.



- Next improvements: studies on Infiniband support.



END OF PRESENTATION

Questions welcome !



University of Rome "Tor Vergata"
Information Engineering Dept.



Oak Ridge
National
Laboratory

MBDA
Software Applied Research &
Technology Department