Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Assignments

Prof. Jesús Labarta

BSC & UPC

ACM Summer school

Barcelona, July 22nd 2019

Agenda

- **Introduction**: before start working with the system
  - Software requirements, login information, configuration, system overview, etc.
- **Paraver tutorial**: getting familiar with paraver
  - A Paraver tutorial to introduce the fundamentals of this tool
- **The matrix multiply example**: playing with this code
  - The matrix multiplication code has appeared during the seminar's slides. A good opportunity to try it!
- **Tutorial exercises**: applying learned lessons (guided examples)
  - Heat and N-Body kernels using TAMPI
  - LULESH using DLB
- **The IFSKer Practice**: let's do it!
  - Submit your report about your experiences applying TAMPI or DLB

Documents & slides (available on-line)

https://pm.bsc.es/ftp/training/acm-ess-2019

Table Of Contents

Introduction
01-Paraver
02-Matrix multiplication
03-TAMPI
04-DLB
05-IFSKER
Software Liability

File Downloads

HybridProgramming.pdf
Assignments.pdf

Quick search

Go

Enter search terms or a module, class or function name.

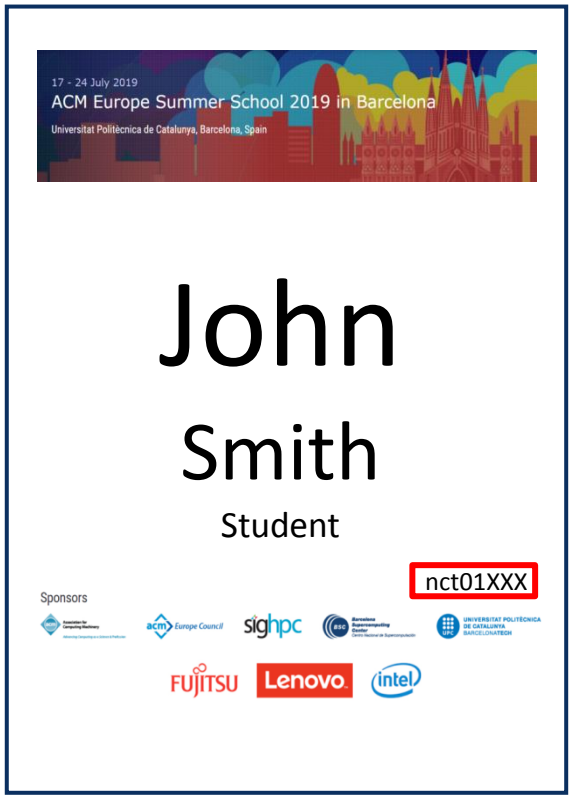
Software requirements

- SSH: Secure SHell (to connect the HPC system)
  - Linux: has native support of secure shell “ssh user@host”
  - Windows: need to install a ssh program
    - PuTTY http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
    - MobaXterm http://mobaxterm.mobatek.net/download.html

- X Server / X forwarding (for wxparaver or .pdf readers)
  - Linux: has native support (remember to connect with **“ssh -X user@host”**
  - Windows: need to install a X server program
    - Xming https://sourceforge.net/projects/xming/
    - MobaXterm already includes a X server within the package

Login information

- Connecting to Nord III (login)
  - nord1.bsc.es
  - nord2.bsc.es
- User provided in your **badge**



Practical information for the ACM Summer School

Xxxxx xxxx xx xxxxx xxxx x xxxxxxxx xx xxxxx xxxx x xxxx xxxxx xxxx xxxxxx xxxx xxxxxx x xx xxxxxx xxxx xx xxxxx xxxxxxxx xx xxxxx xxxx x xxxx

xxxxx xxxx xxxxxx x xx xxxxxx x xxxx xxxxx xxxx xxxxxx x xx xxxxxx xxxx x xxxx xxxxx x xxxx xxxxx xxxx xxxxxx x xx xxxxxx xxxx xxxxxx xxxxxxxxx x xx xxxxxx

xx xxxxx xxxx xx xxxxx xxxx x xxxx xxxxx xxxx xxxxxx x xx xxxxxx xxxx x xxxx xxxxx xxxx xxxxxx x xx xxxxxx

WIFI in Sala Àgora

Xxxxxx
Xxxxxx xxxxxxxx

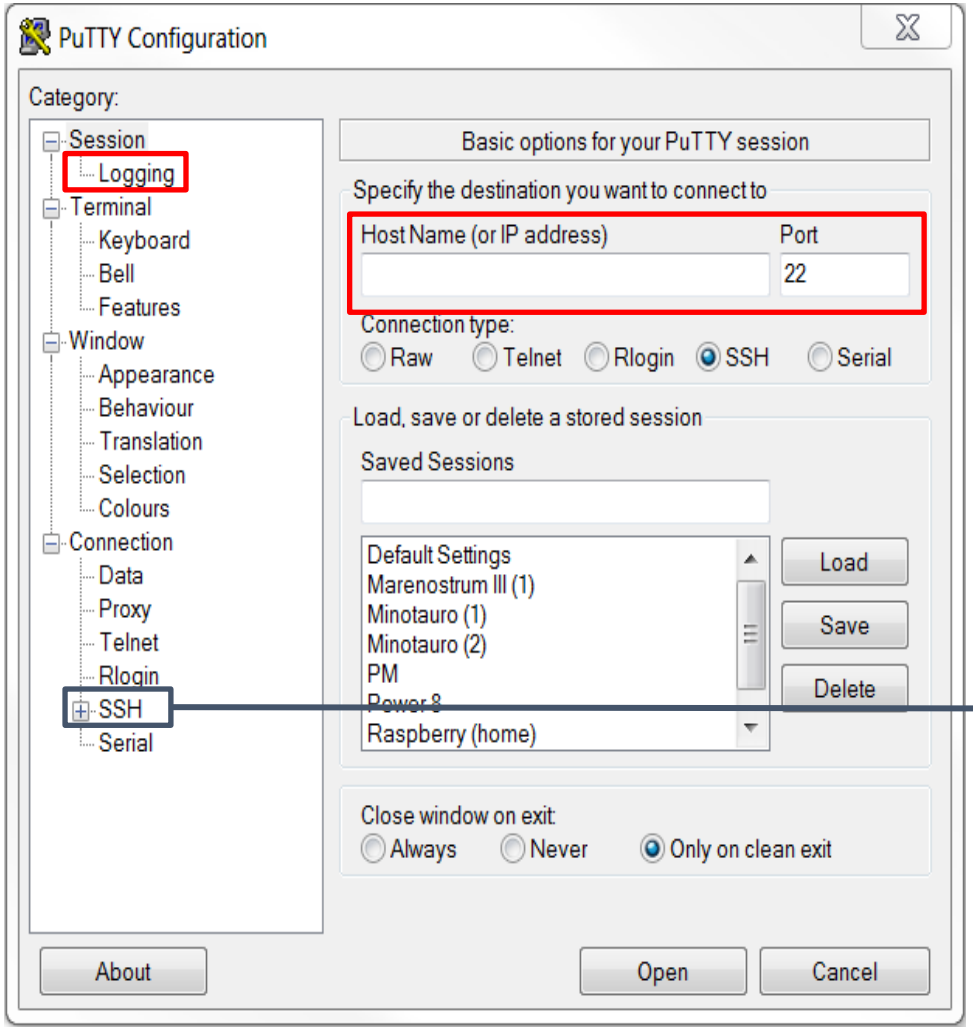
Access to FTP Server for lecture material

xxxxx xxxx xxxxxx x xx xxxxxx xxxx
Xxx xxxxxx x xxxx
Xxx xx xxxxx x xxxx

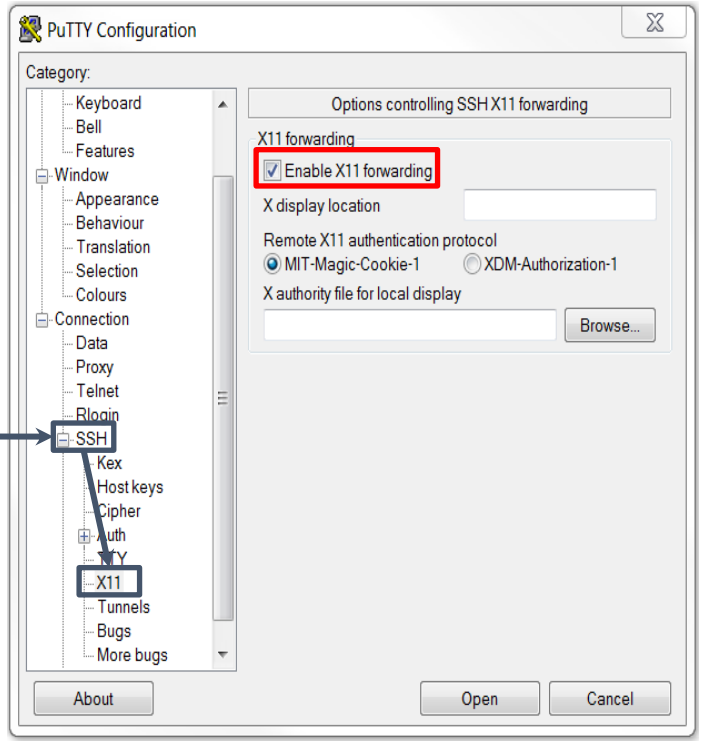
Access to Nord cluster

ssh nct01YYY@nord1.bsc.es
Username: nct01YYY (specific value for YYY included in badge)
Password: ●●●●●●●●●●

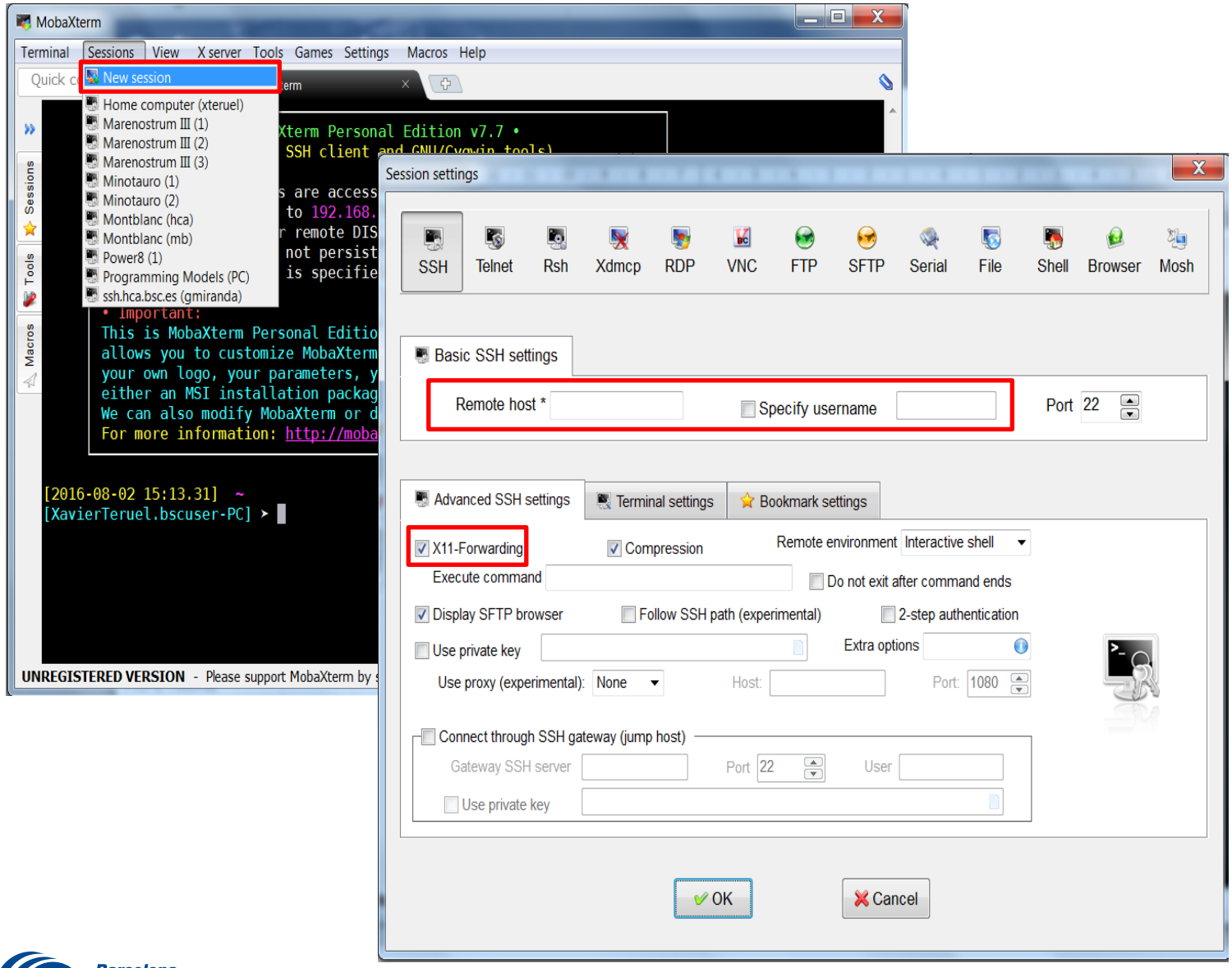
PuTTY: SSH configuration



Before open session make sure…



MobaXterm: SSH configuration



System Overview: Nord III

- 9 iDataPlex compute racks. Each one composed of:
  - 84 IBM dx360 M4 compute nodes
  - 4 Mellanox 36-port Managed FDR10 IB Switches
  - 2 BNT RackSwitch G8052F (Management Network)
  - 2 BNT RackSwitch G8052F (GPFS Network)
  - 4 Power Distribution Units
- All IBM dx360 M4 node contain:
  - 2x E5–2670 SandyBridge-EP 2.6GHz cache 20MB 8-core
  - 500GB 7200 rpm SATA II local HDD
  - 8x 16G DDR3–1600 DIMMs (8GB/core) Total RAM: 128GB/node
- 1.9 PB of GPFS disk storage
- Interconnection Networks
  - Infiniband Mellanox FDR10: High bandwidth network used by parallel applications communications (MPI)
  - Gigabit Ethernet: 10GbitEthernet network used by the GPFS Filesystem.
- Operating System: Linux - SuSe Distribution 11 SP3

Assignment description

https://pm.bsc.es/ftp/training/acm-ess-2019

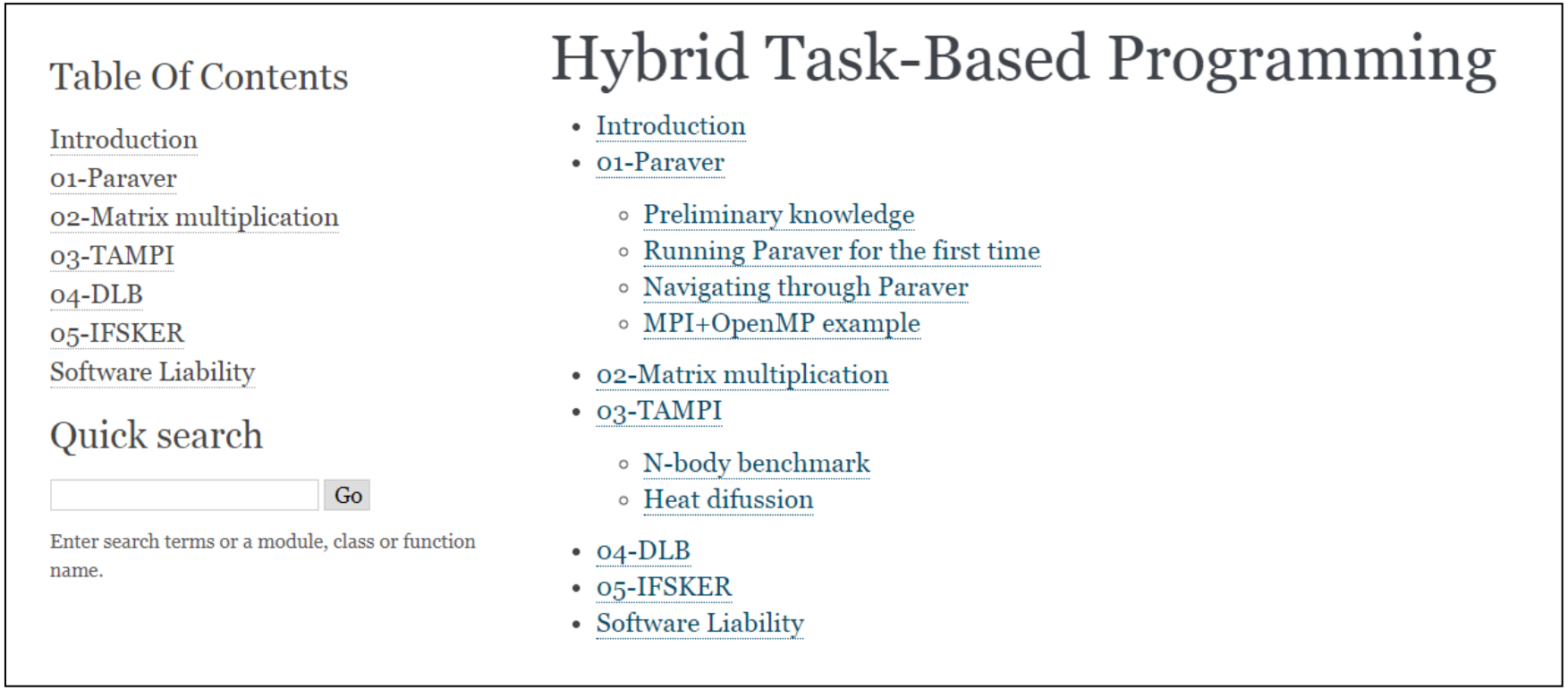


Table Of Contents

Introduction
01-Paraver
02-Matrix multiplication
03-TAMPI
04-DLB
05-IFSKER
Software Liability

Quick search

Go

Enter search terms or a module, class or function name.

Hybrid Task-Based Programming

- Introduction
- 01-Paraver
  - Preliminary knowledge
  - Running Paraver for the first time
  - Navigating through Paraver
  - MPI+OpenMP example
- 02-Matrix multiplication
- 03-TAMPI
  - N-body benchmark
  - Heat difussion
- 04-DLB
- 05-IFSKER
- Software Liability

01 – Paraver Tutorial

- Getting familiar with Paraver
  - You can overlap this exercise with the next matrix multiplication example
  - Get traces, analyze them, get the corresponding histograms, etc.
  - Complete this sections with the Extrae/Paraver documentation [see more]: https://tools.bsc.es/tools_manuals

01-Paraver

Preliminary knowledge

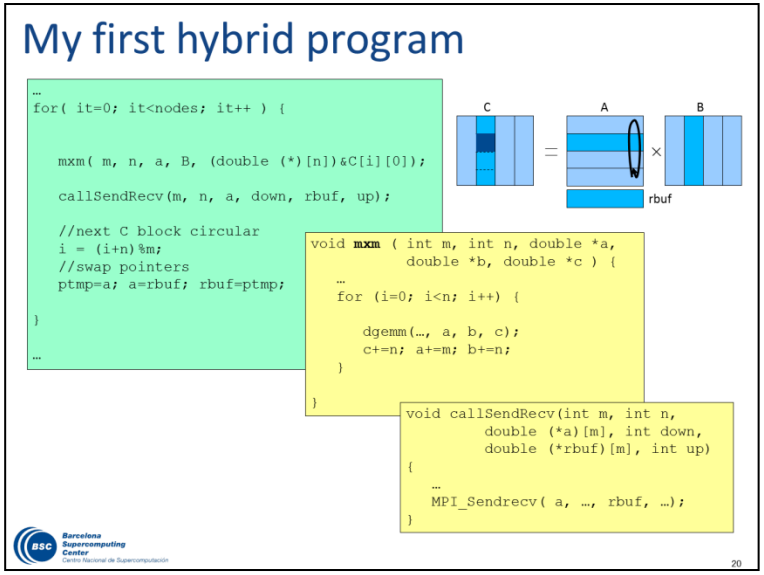
What is Paraver?

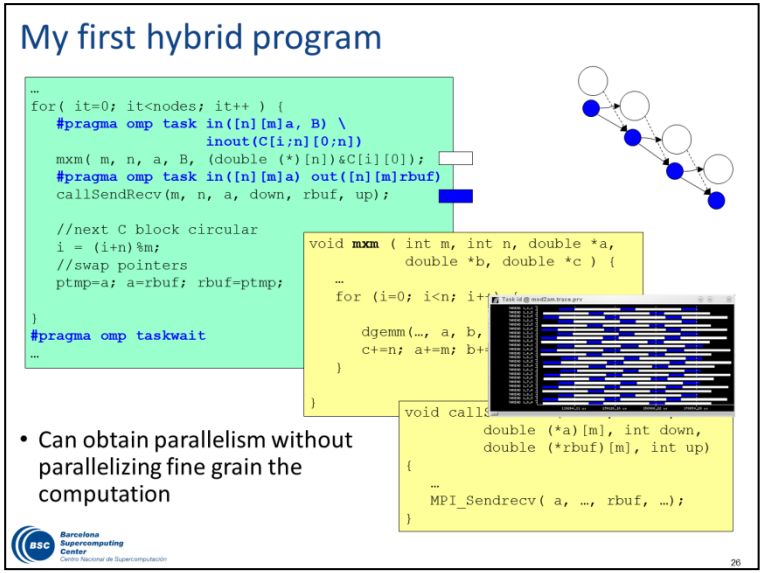
Paraver was developed to respond to the need to have a qualitative global perception of the application behavior by visual inspection and then to be able to focus on the detailed quantitative analysis of the problems. Expressive power, flexibility and the capability of efficiently handling large traces are key features addressed in the design of Paraver. The clear and modular structure of Paraver plays a significant role towards achieving these targets.

02 – Matrix Multiplication example

- This code has been used through this tutorial
- Several version/techniques are included in the slides
  - Overlap computation in tasks and communication in master
  - Overlap between computation and communication in tasks (+ nested)
  - Can start computation of next block of C as soon as communication terminates
  - Can obtain parallelism without parallelizing fine grain the computation
- Now, you can try it, change it, test it

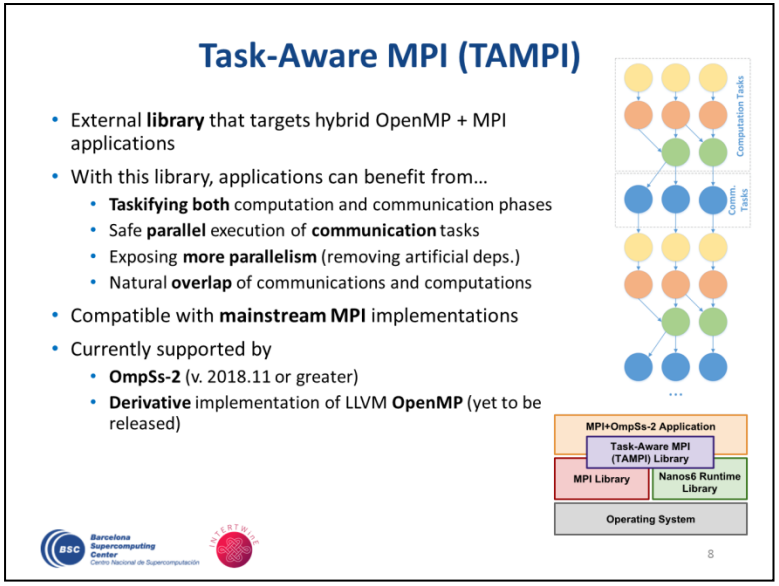
Play with it

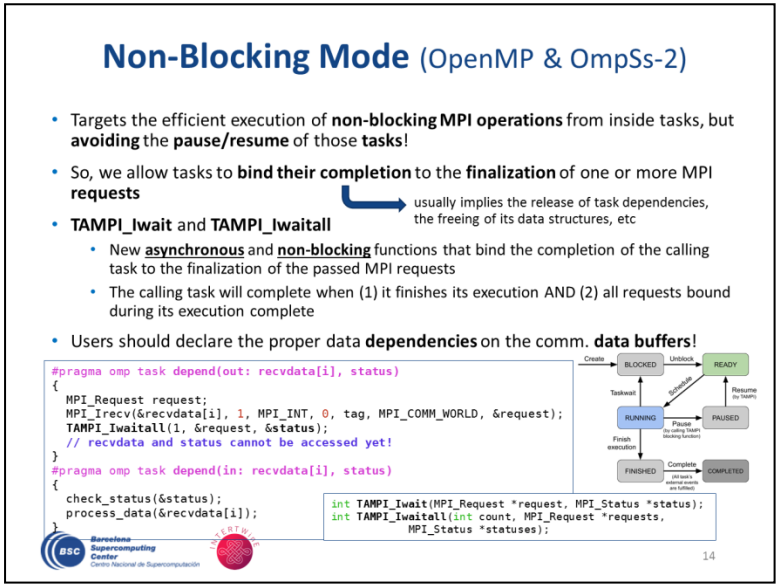




03 – TAMPI Exercises

- **N-Body simulation** kernel: system of bodies interacting one each other
- **Heat diffusion** kernel: Gauss-Seidel method solving the heat equation
- Goals of this exercise
  - Analyze execution, is there comm. serialization?
  - Adapt the code to exploit TAMPI
  - Compare performance results with the initial version
  - Compare traces before/after changes



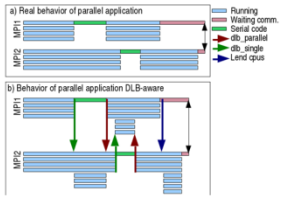


04 – DLB Exercises

- **LULESH**: The Shock Hydrodynamics Challenge Problem
- Goals of this exercise
  - Analyze execution, is there imbalance?
  - Adapt program in order to use DLB
  - Compare performance results with the initial version
  - Compare traces before/after changes

Exploiting malleability

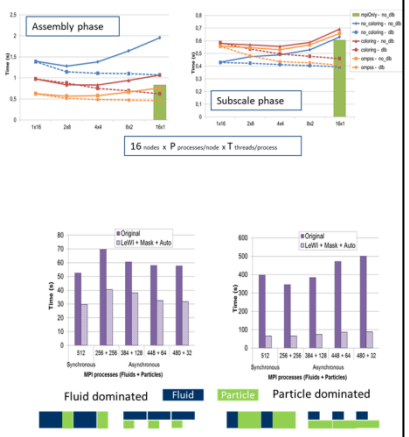
- Dynamic Load Balance & Resource management
  - Intra/inter process/application
- Library (DLB)
  - Runtime interception (MPIP, OMPT, …)
  - API to hint resource demands
  - Core reallocation policy
- Opportunity to fight Amdalh's law
  - Productive / Easy !!!
    - Nx1
    - Hybridize imbalanced regions



"LeWI: A Runtime Balancing Algorithm for Nested Parallelism". M.Garcia et al. ICPP09
"Hints to improve automatic load balancing with LeWI for hybrid applications" JPDC2014

Towards the throughput age

- Dynamic resource sharing/management
- Configuration independence
  - Amount of resources is what really matters
- Side effects
  - Nx1 can be better than pure MPI !!!
  - hope for lazy programmers



05 – IFSKER Practice

- **IFSKernel** code: a kernel of the OpenIFS application, used in global forecast and climate reanalyzes (from ECMWF)
- Consider using other interesting codes:
  - NTCHEM, Your own application! Send us an email[1] to confirm “interest”
- Goals of this exercise
  - Analysis Phase: check program’s arguments and/or runtime’s options impacting on the behavior (e.g., number of threads, schedulers, etc.). Scalability/parallel efficiency. Traces. Histograms.
  - Argument: Is there imbalance? Is there comm. serialization? Others?
  - Determine (and justify) what is your approach:
    - Consider to adapt the code to exploit DLB or TAMPI
    - Is there any other technique you consider it can improve the code?
  - Compare performance results and traces before/after your contribution
  - Highlights. Conclusions. Extensions/next-steps.

[1] Send an email to **htbp-submit [at] bsc.es**

05 – IFSKER Practice (submit)

- Write a 5-10 pages document report. It should include:
  - Initial performance analysis (how the application behaves)
    - Scaling/efficiency factors according to…
    - Paraver traces/histograms
    - Others,…
  - Proof of concept description (your plan)
  - Implementation description (what you have done)
    - Parallelization, taskyfication, synchronization, etc.
  - Experiment results (how the application behaves now)
    - Version comparison, discussion, compare them with initial analysis
  - Experiences and conclusions (what you have learnt)
- Submit a single **.tar.gz** file (< 20 MB) including the document, source codes, traces, etc. to **htbp-submit [at] bsc.es** before **September 15th 2019**

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Thanks!