StarPU: Programming for Heterogeneous MultiGPU Systems

JOAO GAZOLLA – PHD CANDIDATE / UFF MEDIALAB RESEARCHER

ESTEBAN CLUA – ASSOCIATE PROFESSOR / UFF MEDIALAB DIRECTOR

GPU TECHNOLOGY CONFERENCE

Instituto de Computação

GTC 2015, San Jose, CA, USA, March 18th, 2015.

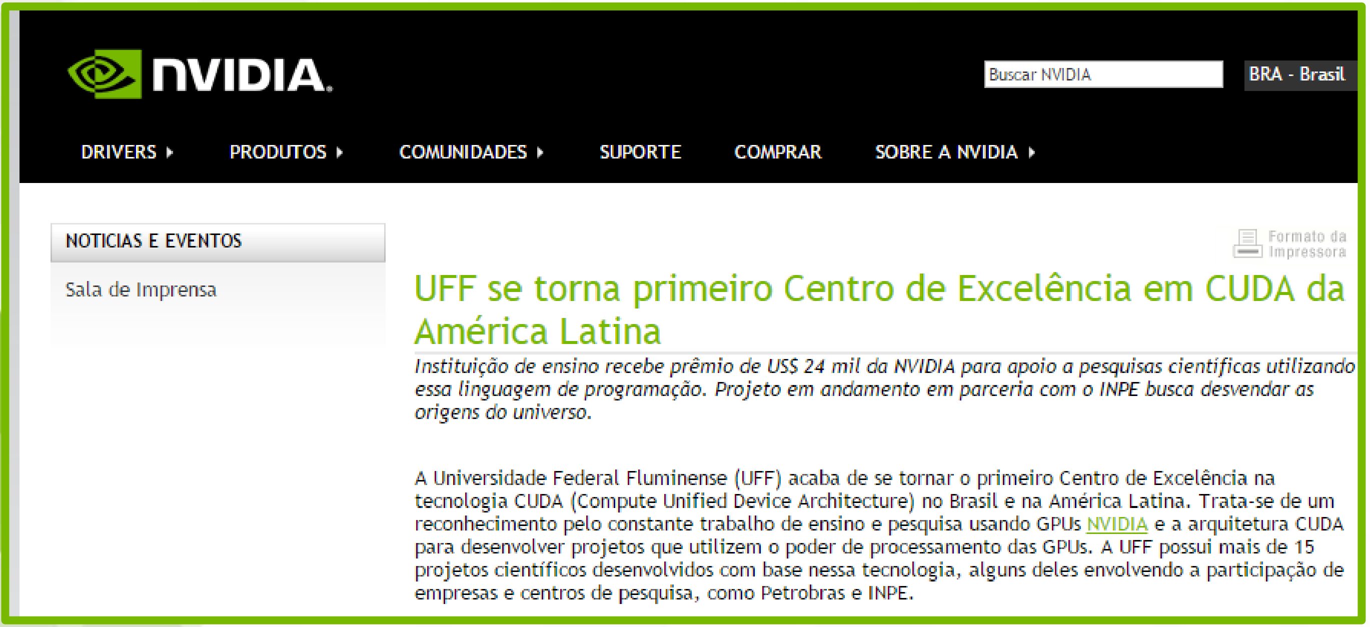
About Us

- Joao Gazolla is a PhD candidate a UFF/Rio de Janeiro, Brazil and Researcher at MediaLab/UFF.

- Esteban Clua is an associate professor at UFF (Federal Fluminense University) and Director of MediaLab at UFF.

Medialab UFF

- MediaLab UFF was the First Cuda Research Center in Latin America.



UFF se torna primeiro Centro de Excelência em CUDA da América Latina

Instituição de ensino recebe prêmio de US$ 24 mil da NVIDIA para apoio a pesquisas científicas utilizando essa linguagem de programação. Projeto em andamento em parceria com o INPE busca desvendar as origens do universo.

A Universidade Federal Fluminense (UFF) acaba de se tornar o primeiro Centro de Excelência na tecnologia CUDA (Compute Unified Device Architecture) no Brasil e na América Latina. Trata-se de um reconhecimento pelo constante trabalho de ensino e pesquisa usando GPUs NVIDIA e a arquitetura CUDA para desenvolver projetos que utilizem o poder de processamento das GPUs. A UFF possui mais de 15 projetos científicos desenvolvidos com base nessa tecnologia, alguns deles envolvendo a participação de empresas e centros de pesquisa, como Petrobras e INPE.

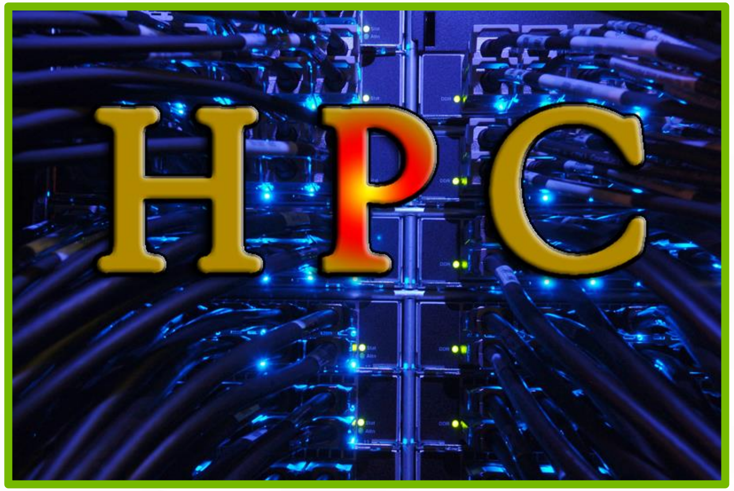
2012

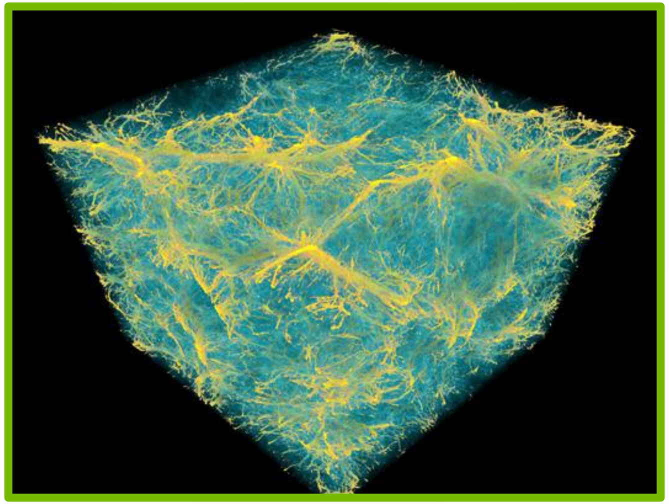
GPU TECHNOLOGY CONFERENCE

Medialab UFF

NVIDIA

CUDA™
TEACHING
CENTER





Before we really start...

- We are not part of the StarPU Team.
- We are researchers using their framework.

- StarPU was developed by INRIA Bordeaux, France.

Outline

- Objective
- Motivation
- StarPU Overview
- StarPU "Hello World"
- Conclusion
- Questions and Answers

What you will learn in this session...

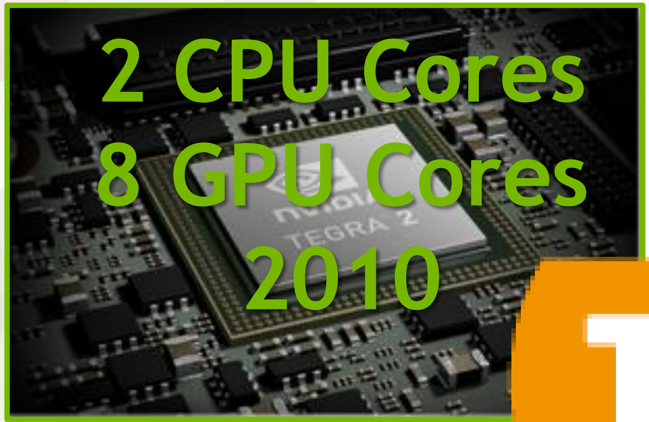
2 Main Objectives:

- Understand what is StarPU.
- Understand How It Works.

Outline

- Objectives
- Motivation
- StarPU Overview
- StarPU “Hello World”
- Conclusion
- Questions and Answers

Motivation



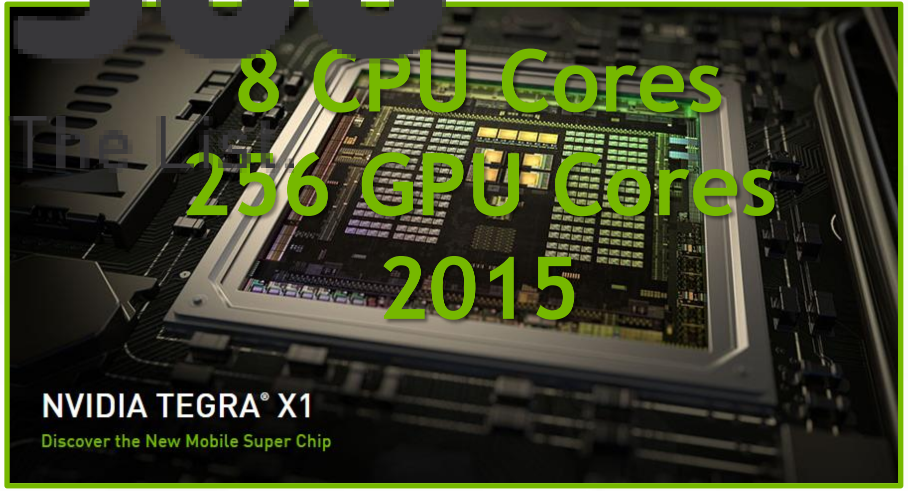
2 CPU Cores
8 GPU Cores
2010

4 CPU Cores
12 GPU Cores
11/2012

5 CPU Cores
72 GPU Cores
2013

4 CPU Cores
192 GPU Cores
2014

NVIDIA® TEGRA® K1
IMPOSSIBLY ADVANCED



8 CPU Cores
256 GPU Cores
2015

NVIDIA TEGRA® X1
Discover the New Mobile Super Chip

TOP 500
The List.

Motivation

“…very little attention has been paid to…the possibility of having heterogeneous accelerators and processors to interact…” **

**StarPU Team at http://starpu.gforge.inria.fr/doc/html/

Imagine if...

Geforce 9800 GT

112 cores

Titan X

~3000 cores/each

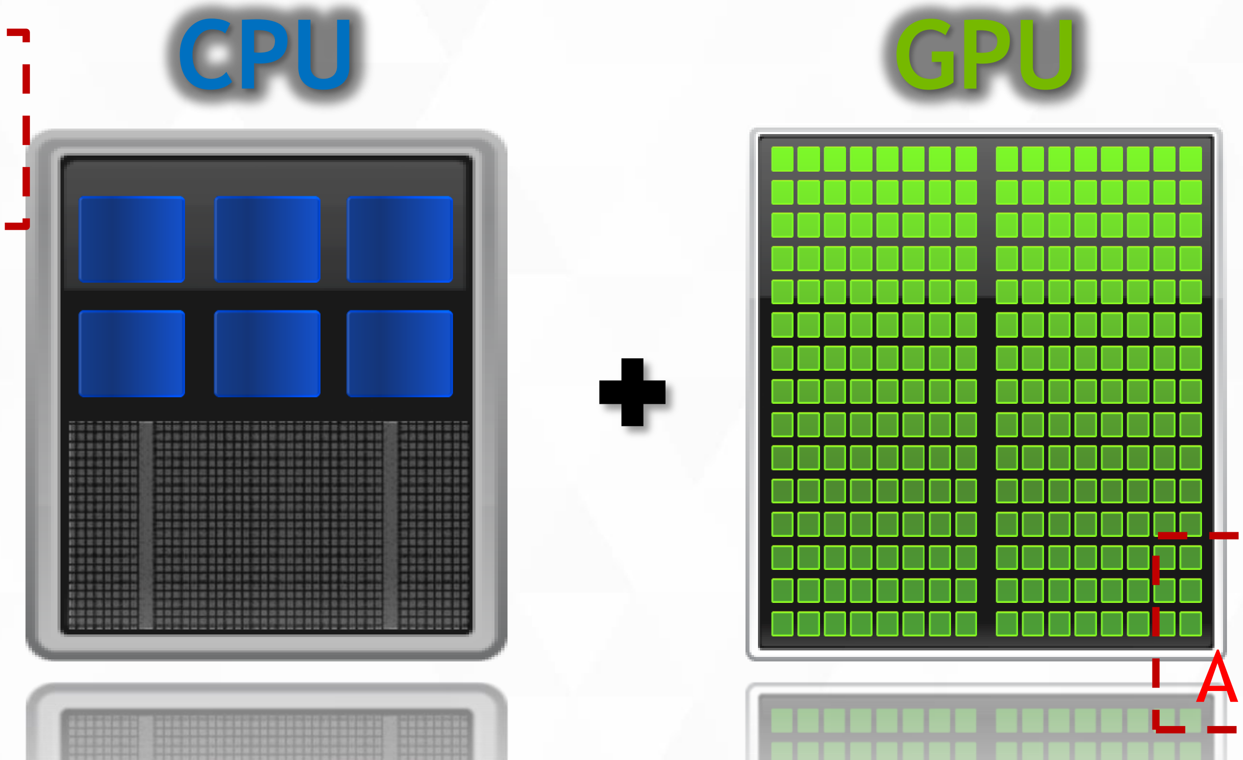
GTX 660

960 cores

What is StarPU?

StarPU is a task programming library for hybrid architectures.

Can’t Ignore CPUs



High End Architecture

You could ask…

"…But using CUDA, you can already program to many GPUs and CPUS simultaneously, right?"

YES! You are right! But…

You could ask…

NVIDIA CUDA

NVIDIA CUDA

+

StarPU

Outline

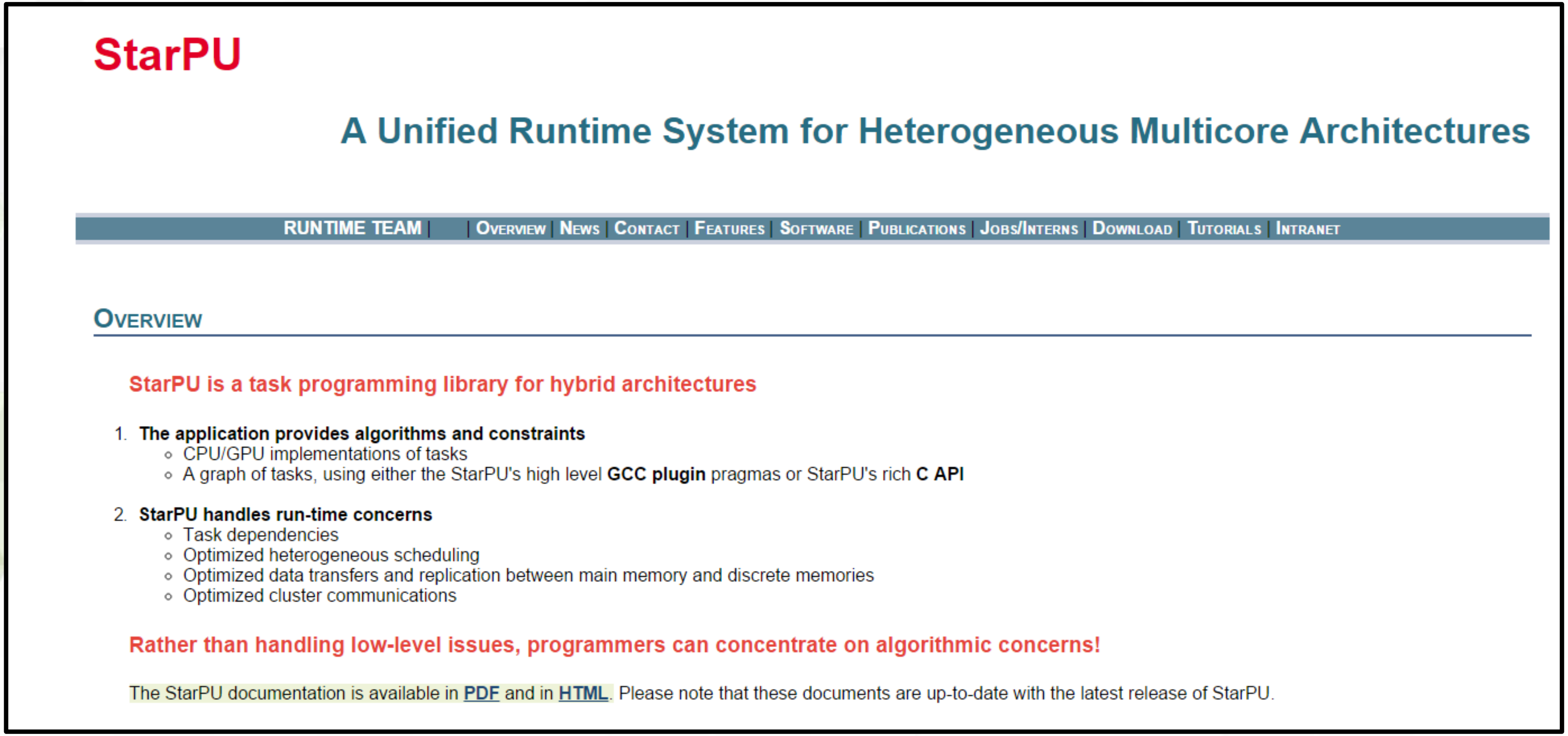
- Objectives
- Motivation
- StarPU Overview
- StarPU “Hello World”
- Conclusion
- Questions and Answers

StarPU Overview

GPU TECHNOLOGY CONFERENCE

About StarPU

- Developed by INRIA Bordeaux, France.



StarPU

A Unified Runtime System for Heterogeneous Multicore Architectures

RUNTIME TEAM | | OVERVIEW | NEWS | CONTACT | FEATURES | SOFTWARE | PUBLICATIONS | JOBS/INTERNS | DOWNLOAD | TUTORIALS | INTRANET

OVERVIEW

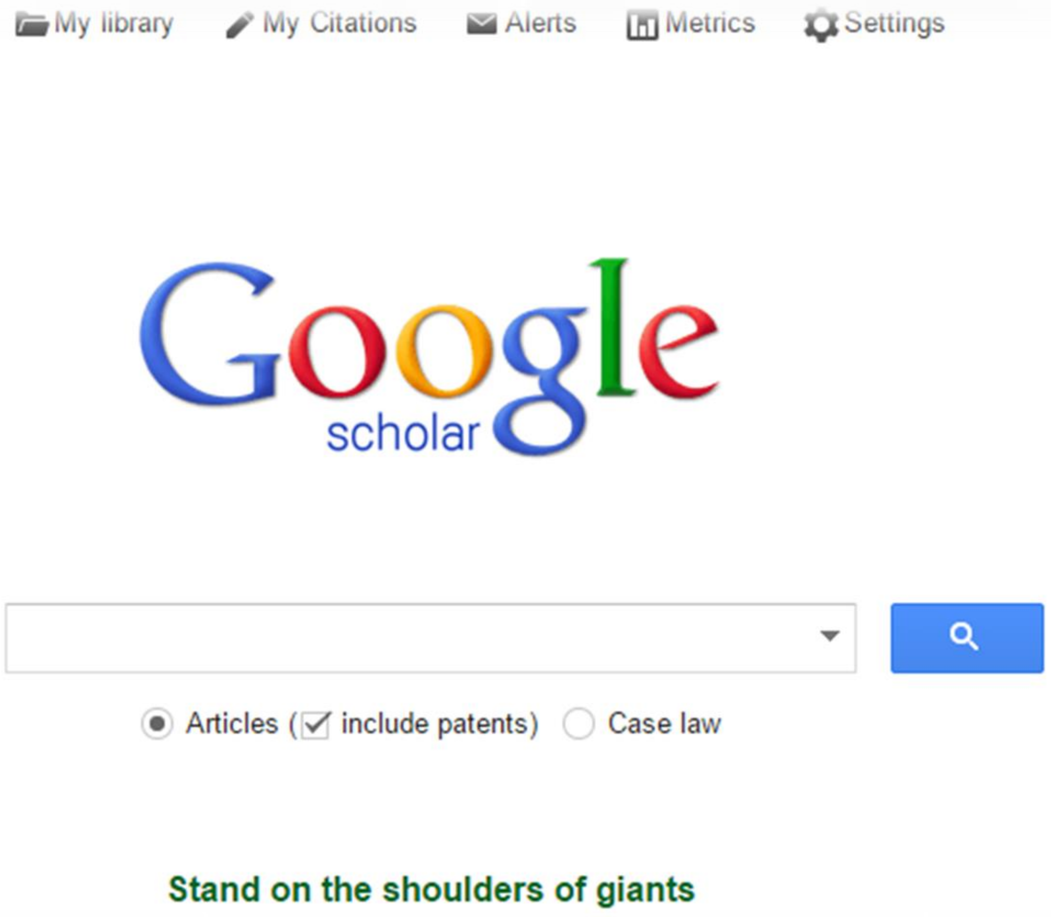
StarPU is a task programming library for hybrid architectures

1. **The application provides algorithms and constraints**
   - CPU/GPU implementations of tasks
   - A graph of tasks, using either the StarPU's high level **GCC plugin** pragmas or StarPU's rich **C API**
2. **StarPU handles run-time concerns**
   - Task dependencies
   - Optimized heterogeneous scheduling
   - Optimized data transfers and replication between main memory and discrete memories
   - Optimized cluster communications

Rather than handling low-level issues, programmers can concentrate on algorithmic concerns!

The StarPU documentation is available in PDF and in HTML. Please note that these documents are up-to-date with the latest release of StarPU.

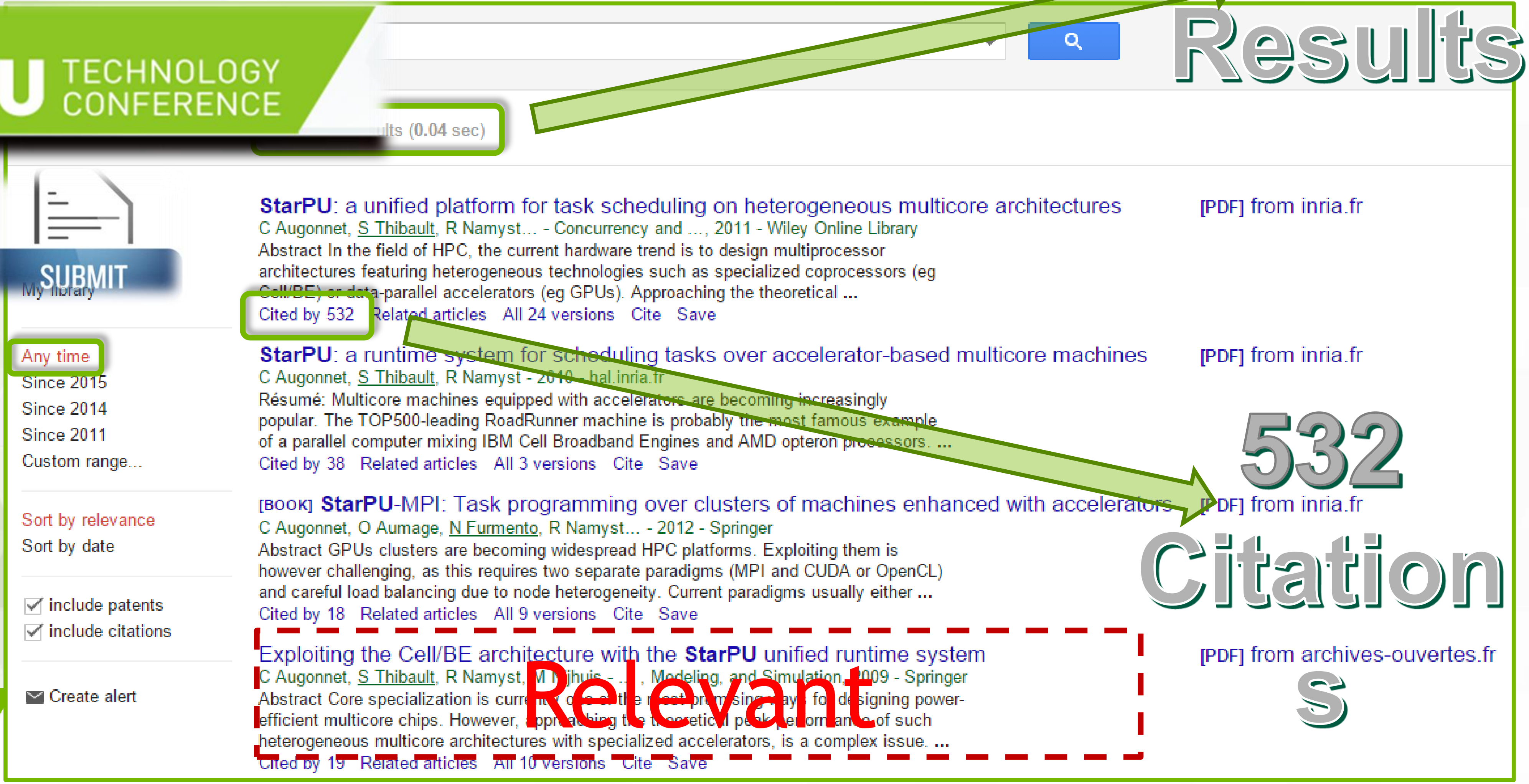
Why we are using StarPU?



● Articles (☑ include patents) ○ Case law

Stand on the shoulders of giants

StarPU on Google Scholar



ılts (0.04 sec)

SUBMIT

My library

Any time
Since 2015
Since 2014
Since 2011
Custom range...

Sort by relevance
Sort by date

☑ include patents
☑ include citations

✉ Create alert

StarPU: a unified platform for task scheduling on heterogeneous multicore architectures — [PDF] from inria.fr
C Augonnet, S Thibault, R Namyst... - Concurrency and ..., 2011 - Wiley Online Library
Abstract In the field of HPC, the current hardware trend is to design multiprocessor architectures featuring heterogeneous technologies such as specialized coprocessors (eg Cell/BE) or data-parallel accelerators (eg GPUs). Approaching the theoretical ...
Cited by 532 Related articles All 24 versions Cite Save

StarPU: a runtime system for scheduling tasks over accelerator-based multicore machines — [PDF] from inria.fr
C Augonnet, S Thibault, R Namyst - 2010 - hal.inria.fr
Résumé: Multicore machines equipped with accelerators are becoming increasingly popular. The TOP500-leading RoadRunner machine is probably the most famous example of a parallel computer mixing IBM Cell Broadband Engines and AMD opteron processors. ...
Cited by 38 Related articles All 3 versions Cite Save

[BOOK] **StarPU**-MPI: Task programming over clusters of machines enhanced with accelerators — [PDF] from inria.fr
C Augonnet, O Aumage, N Furmento, R Namyst... - 2012 - Springer
Abstract GPUs clusters are becoming widespread HPC platforms. Exploiting them is however challenging, as this requires two separate paradigms (MPI and CUDA or OpenCL) and careful load balancing due to node heterogeneity. Current paradigms usually either ...
Cited by 18 Related articles All 9 versions Cite Save

Exploiting the Cell/BE architecture with the **StarPU** unified runtime system — [PDF] from archives-ouvertes.fr
C Augonnet, S Thibault, R Namyst, M Nijhuis - ..., Modeling, and Simulation, 2009 - Springer
Abstract Core specialization is currently one of the most promising ways for designing power-efficient multicore chips. However, approaching the theoretical peak performance of such heterogeneous multicore architectures with specialized accelerators, is a complex issue. ...
Cited by 19 Related articles All 10 versions Cite Save

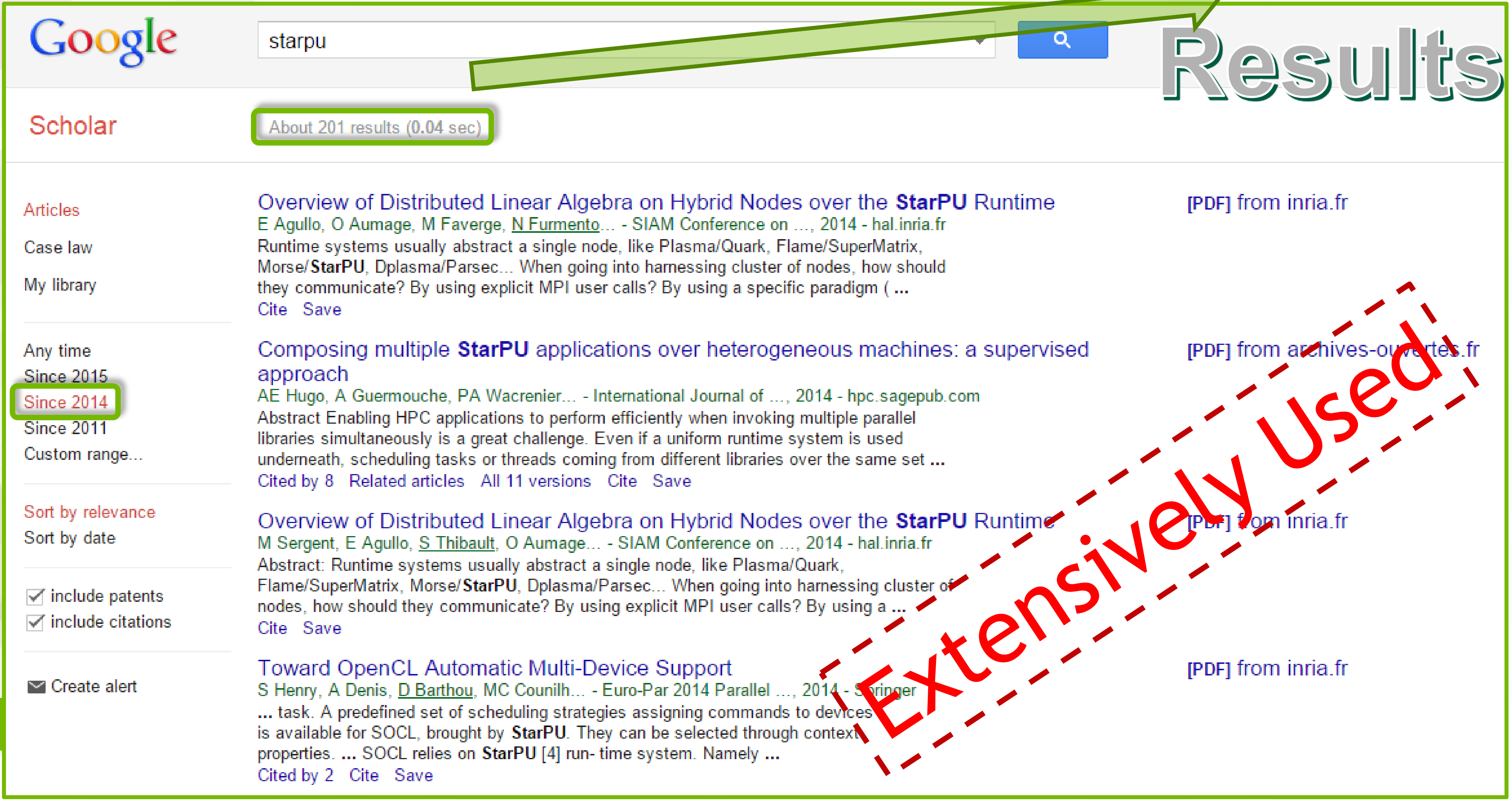
941 Results

532 Citations

Relevant

StarPU on Google Scholar

201 Results



About 201 results (0.04 sec)

Articles
Case law
My library

Any time
Since 2015
Since 2014
Since 2011
Custom range...

Sort by relevance
Sort by date

☑ include patents
☑ include citations

✉ Create alert

Overview of Distributed Linear Algebra on Hybrid Nodes over the StarPU Runtime — [PDF] from inria.fr
E Agullo, O Aumage, M Faverge, N Furmento... - SIAM Conference on ..., 2014 - hal.inria.fr
Runtime systems usually abstract a single node, like Plasma/Quark, Flame/SuperMatrix, Morse/StarPU, Dplasma/Parsec... When going into harnessing cluster of nodes, how should they communicate? By using explicit MPI user calls? By using a specific paradigm (...
Cite Save

Composing multiple StarPU applications over heterogeneous machines: a supervised approach — [PDF] from archives-ouvertes.fr
AE Hugo, A Guermouche, PA Wacrenier... - International Journal of ..., 2014 - hpc.sagepub.com
Abstract Enabling HPC applications to perform efficiently when invoking multiple parallel libraries simultaneously is a great challenge. Even if a uniform runtime system is used underneath, scheduling tasks or threads coming from different libraries over the same set ...
Cited by 8 Related articles All 11 versions Cite Save

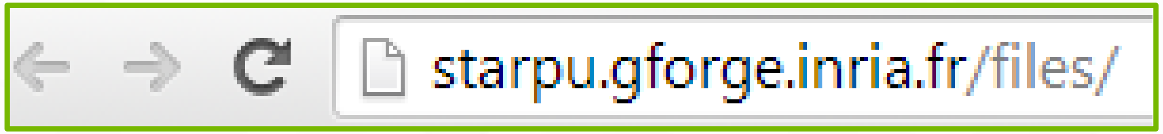
Overview of Distributed Linear Algebra on Hybrid Nodes over the StarPU Runtime — [PDF] from inria.fr
M Sergent, E Agullo, S Thibault, O Aumage... - SIAM Conference on ..., 2014 - hal.inria.fr
Abstract: Runtime systems usually abstract a single node, like Plasma/Quark, Flame/SuperMatrix, Morse/StarPU, Dplasma/Parsec... When going into harnessing cluster of nodes, how should they communicate? By using explicit MPI user calls? By using a ...
Cite Save

Toward OpenCL Automatic Multi-Device Support — [PDF] from inria.fr
S Henry, A Denis, D Barthou, MC Counilh... - Euro-Par 2014 Parallel ..., 2014 - Springer
... task. A predefined set of scheduling strategies assigning commands to devices is available for SOCL, brought by StarPU. They can be selected through context properties. ... SOCL relies on StarPU [4] run- time system. Namely ...
Cited by 2 Cite Save

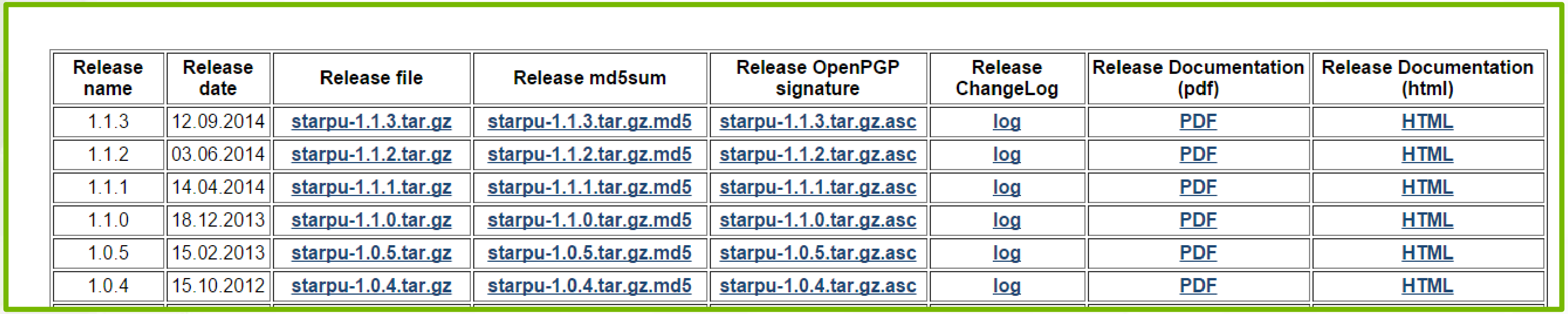
Extensively Used

GPU TECHNOLOGY CONFERENCE

StarPU - Download



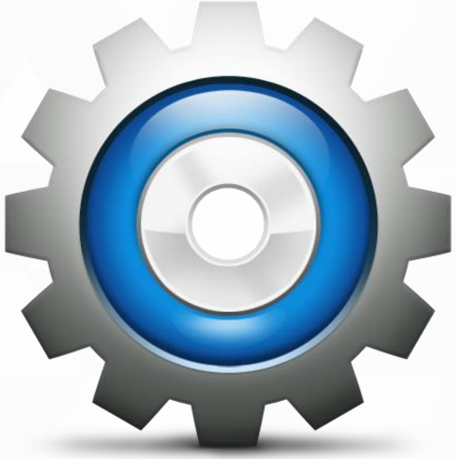
starpu.gforge.inria.fr/files/



Release name	Release date	Release file	Release md5sum	Release OpenPGP signature	Release ChangeLog	Release Documentation (pdf)	Release Documentation (html)
1.1.3	12.09.2014	starpu-1.1.3.tar.gz	starpu-1.1.3.tar.gz.md5	starpu-1.1.3.tar.gz.asc	log	PDF	HTML
1.1.2	03.06.2014	starpu-1.1.2.tar.gz	starpu-1.1.2.tar.gz.md5	starpu-1.1.2.tar.gz.asc	log	PDF	HTML
1.1.1	14.04.2014	starpu-1.1.1.tar.gz	starpu-1.1.1.tar.gz.md5	starpu-1.1.1.tar.gz.asc	log	PDF	HTML
1.1.0	18.12.2013	starpu-1.1.0.tar.gz	starpu-1.1.0.tar.gz.md5	starpu-1.1.0.tar.gz.asc	log	PDF	HTML
1.0.5	15.02.2013	starpu-1.0.5.tar.gz	starpu-1.0.5.tar.gz.md5	starpu-1.0.5.tar.gz.asc	log	PDF	HTML
1.0.4	15.10.2012	starpu-1.0.4.tar.gz	starpu-1.0.4.tar.gz.md5	starpu-1.0.4.tar.gz.asc	log	PDF	HTML

StarPU – Supported Operating Systems

Windows



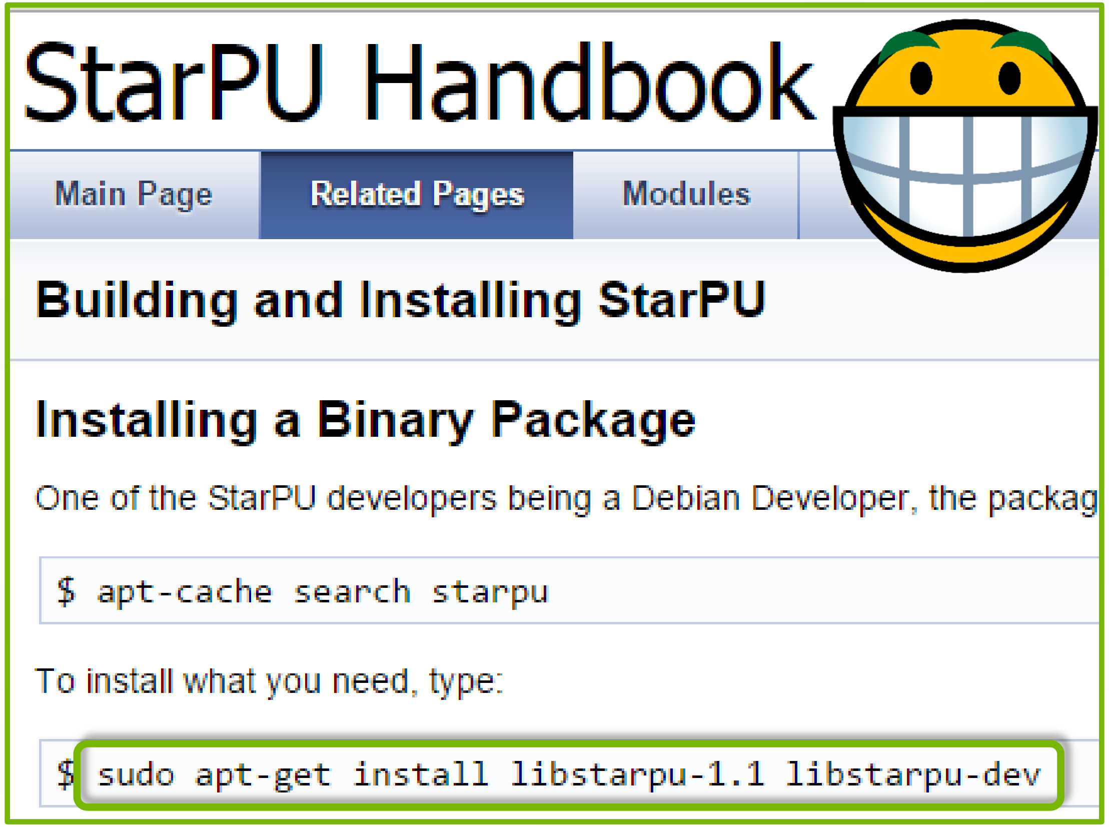
Linux™

StarPU - Installation

Windows

Easier to Install On Linux

Adjust Library Path



StarPU Handbook

Building and Installing StarPU

Installing a Binary Package

One of the StarPU developers being a Debian Developer, the packag

```
$ apt-cache search starpu
```

To install what you need, type:

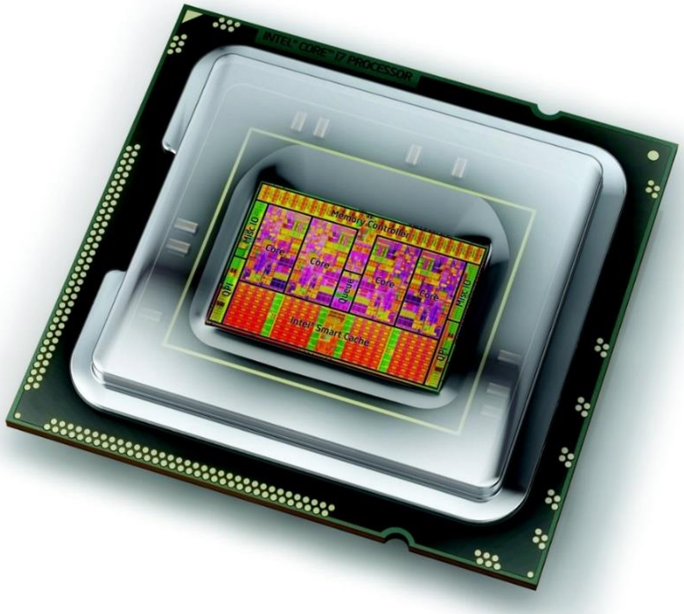
```
$ sudo apt-get install libstarpu-1.1 libstarpu-dev
```

StarPU - “System Requirements”

4 x K80

4 x 4992 CUDA Cores

~20000 Cores

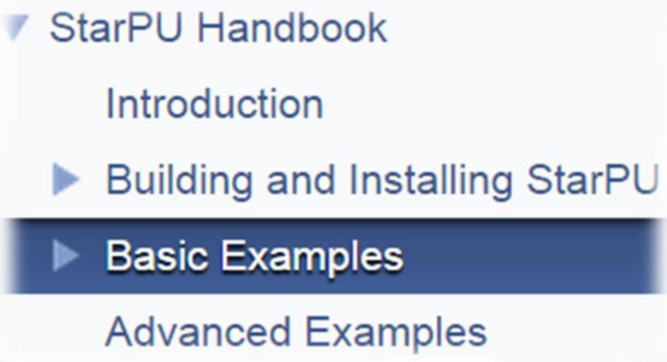


Tips and Tricks

- Install hwloc: “libhwloc-dev”

Tips and Tricks

- Sample Codes Folder.



StarPU Considerations

- Allocates and Dispatches Resources, based on tasks.

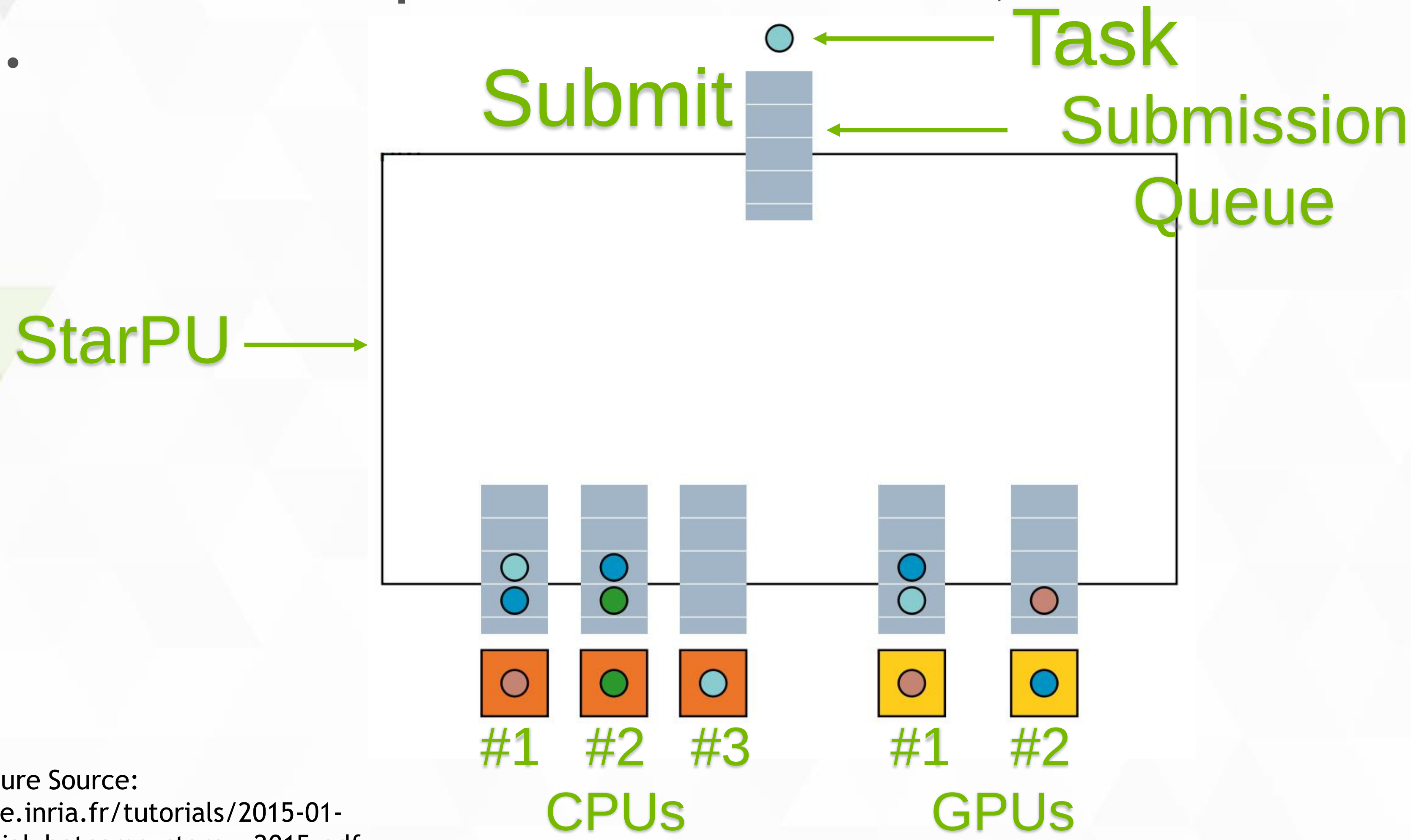


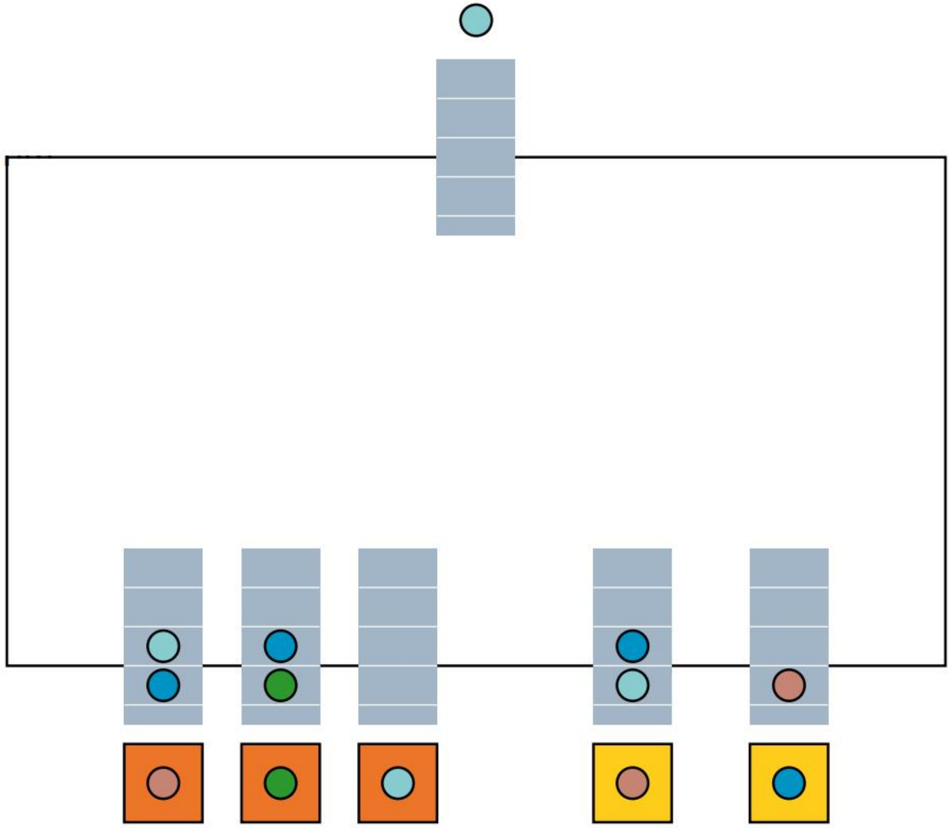
Figure Source: http://starpu.gforge.inria.fr/tutorials/2015-01-HiPEAC/hipeac_tutorial_hetcomp_starpu_2015.pdf

StarPU Considerations

- How you assign and schedule kernels to each device can make a huge difference on execution time of your program.

Greedy

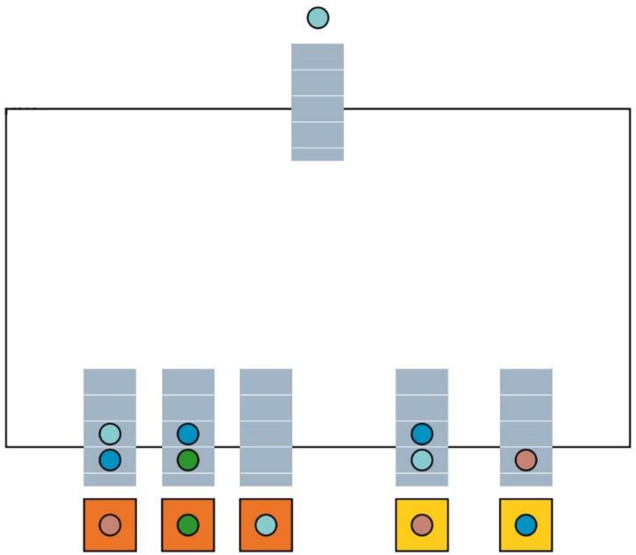
Work Stealing



StarPU Considerations

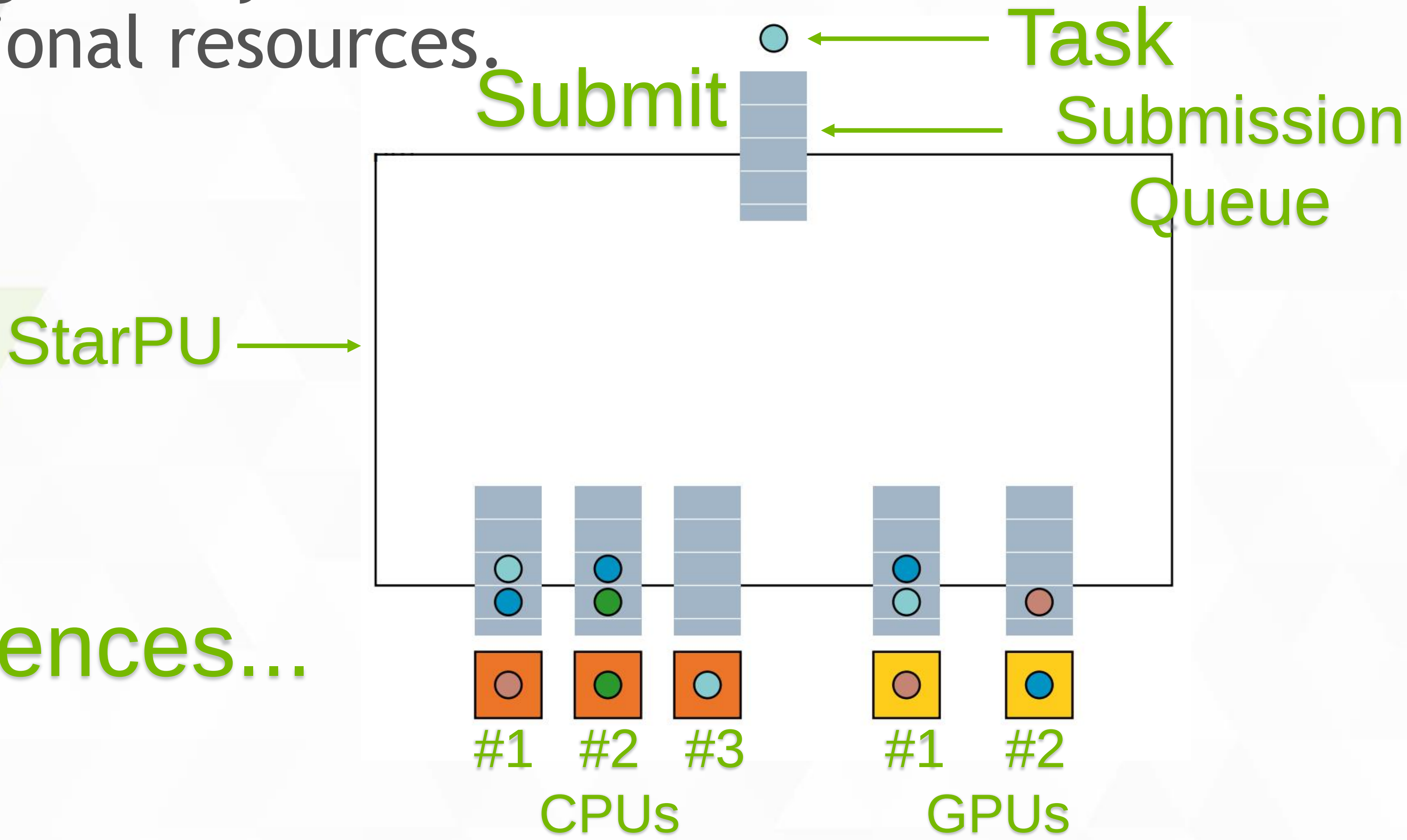
- What happens inside the scheduler is up to you!

Analogy…



StarPU Considerations

- StarPU gives you an unified view of the computational resources.



Consequences...

StarPU Considerations

Imagine if…

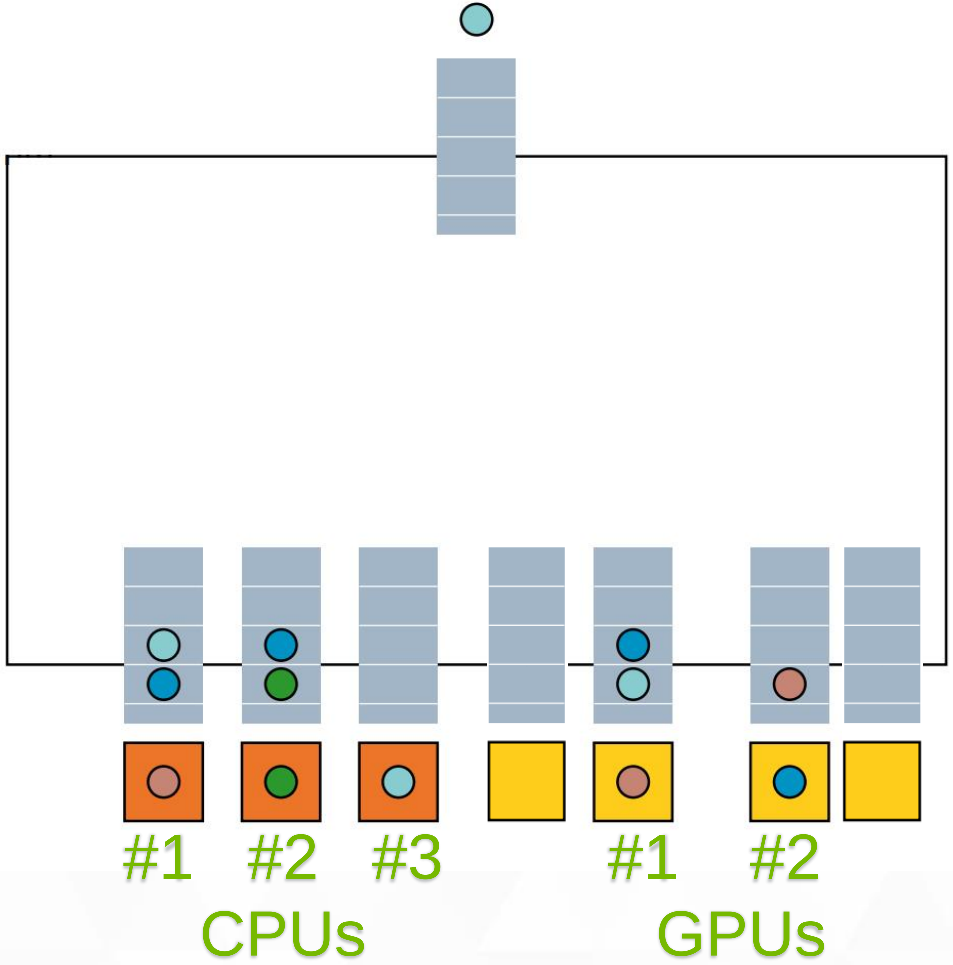
- "What are the consequences ?"

Once Implemented ➔ "Free SpeedUps"

30% ?

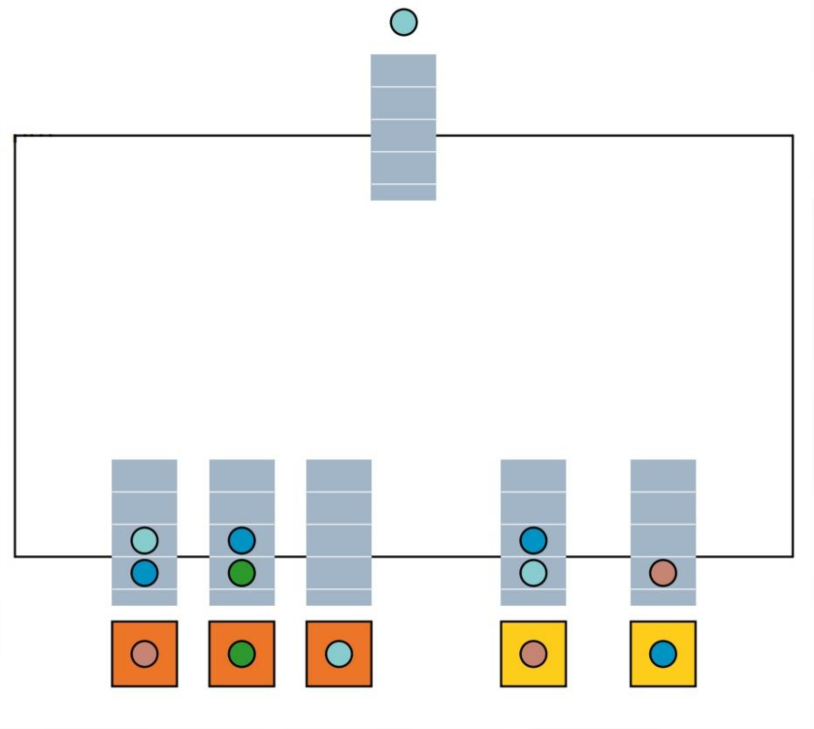
10% ?

20% ?



StarPU Considerations

- The core of StarPU is its run-time support library, which is responsible for scheduling application-provided tasks on heterogeneous machines.



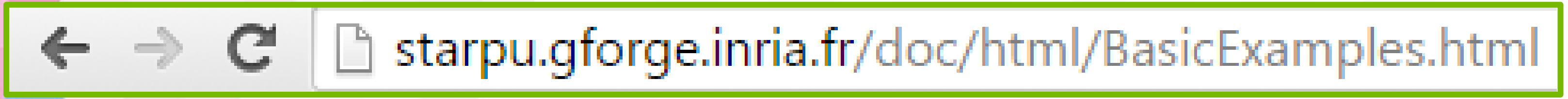
Outline

- Objectives
- Motivation
- StarPU Overview
- StarPU “Hello World”
- Conclusion
- Questions and Answers

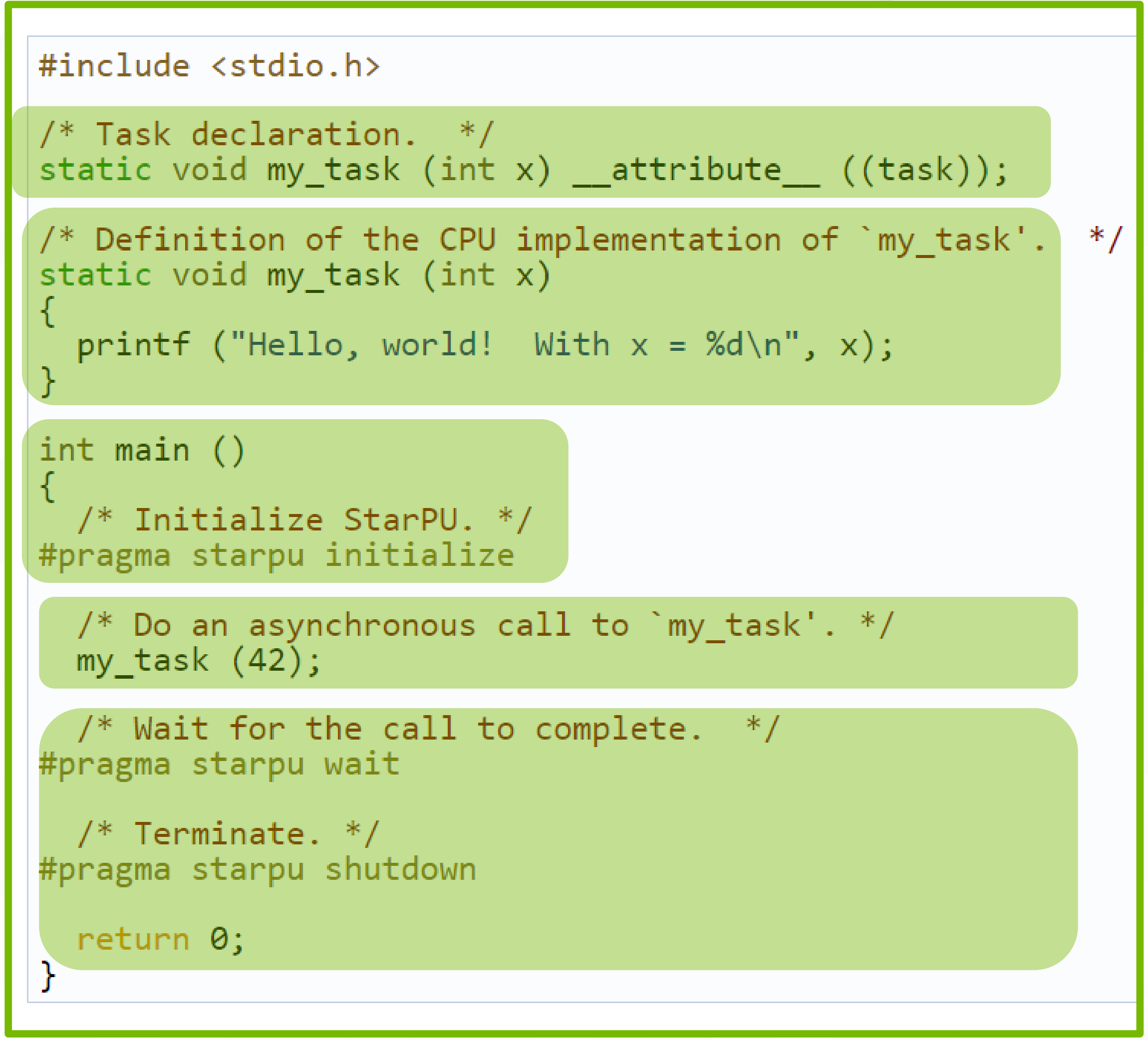
StarPU Hello World

GPU TECHNOLOGY CONFERENCE

StarPU - Hello World



starpu.gforge.inria.fr/doc/html/BasicExamples.html



```
#include <stdio.h>

/* Task declaration.  */
static void my_task (int x) __attribute__ ((task));

/* Definition of the CPU implementation of `my_task'.  */
static void my_task (int x)
{
  printf ("Hello, world!  With x = %d\n", x);
}

int main ()
{
  /* Initialize StarPU. */
#pragma starpu initialize

  /* Do an asynchronous call to `my_task'. */
  my_task (42);

  /* Wait for the call to complete.  */
#pragma starpu wait

  /* Terminate. */
#pragma starpu shutdown

  return 0;
}
```

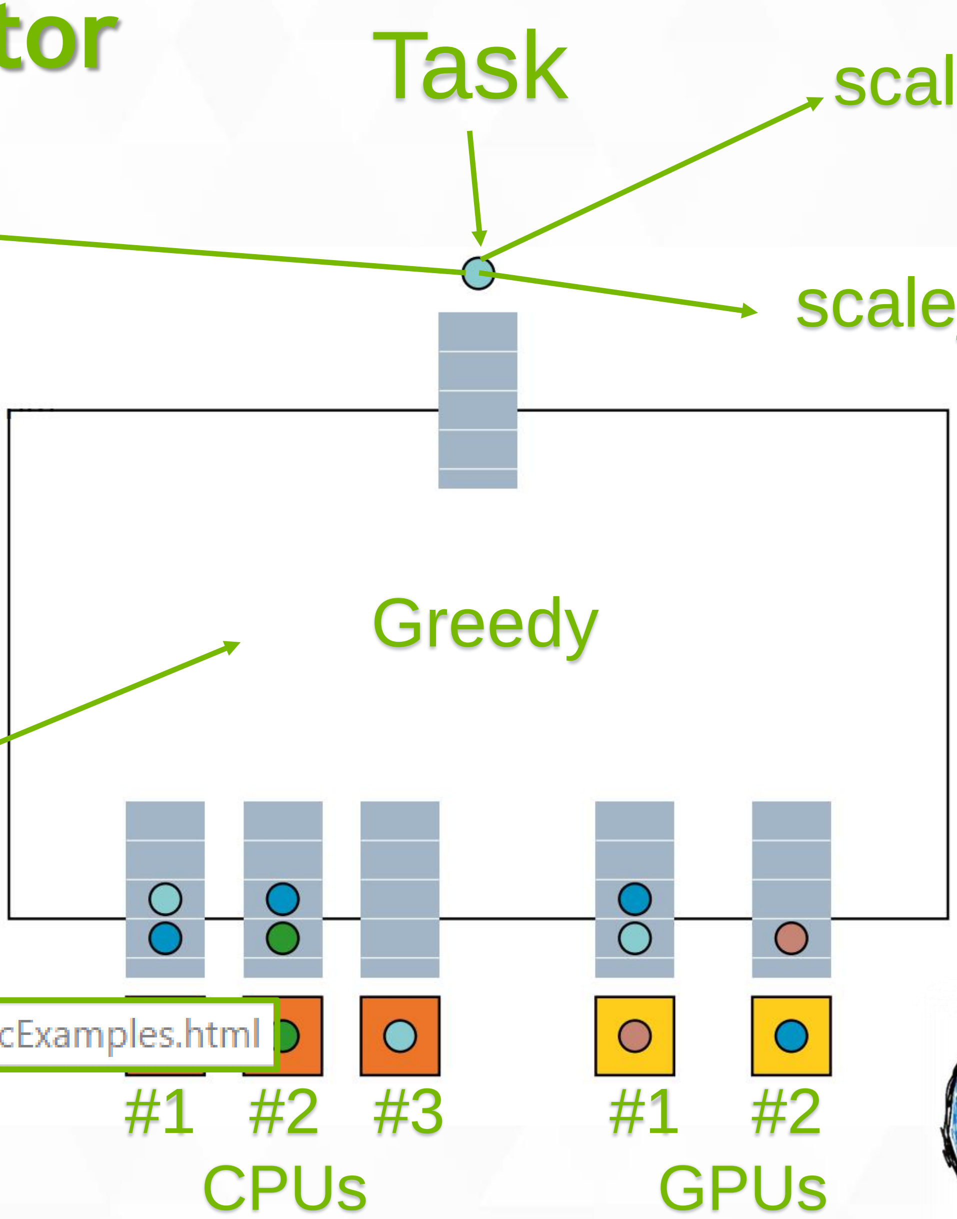
StarPU - Scale a Vector

[0,1,2,3,4]

X 2.0

[0,2,4,6,8]

Schedule Algorithms



starpu.gforge.inria.fr/doc/html/BasicExamples.html

Checking Objectives

What you have learned in this session...

- Understand what is StarPU. ✓
- Understand How It Works. ✓

Outline

- Objectives
- Motivation
- StarPU Overview
- StarPU “Hello World”
- Conclusion
- Questions and Answers

Conclusions and Recap

#1

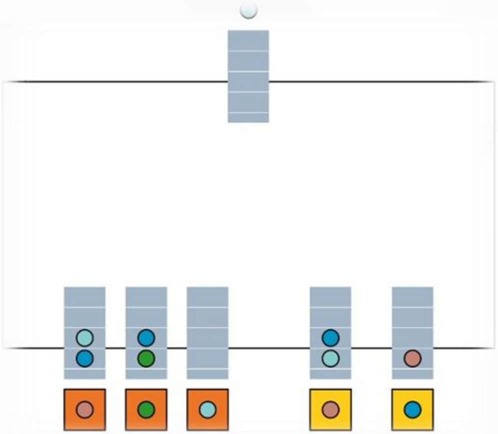
Using StarPU programmers can concentrate on algorithmic concerns.

Abstraction Layer!

Conclusions and Recap

#2

The programmer has a view of unified computation resources.



Conclusions and Recap

#3

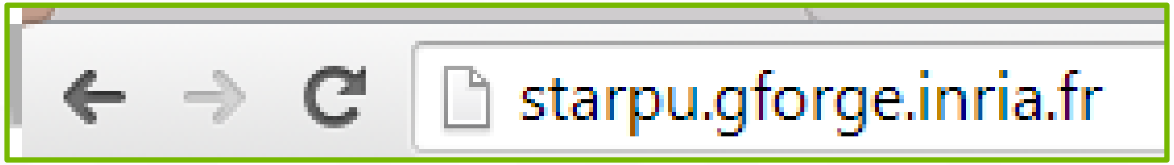
A lot of has been done to improve computing power, but many efforts still needs to be done to improve cooperation among different kind of processors.

Learning More

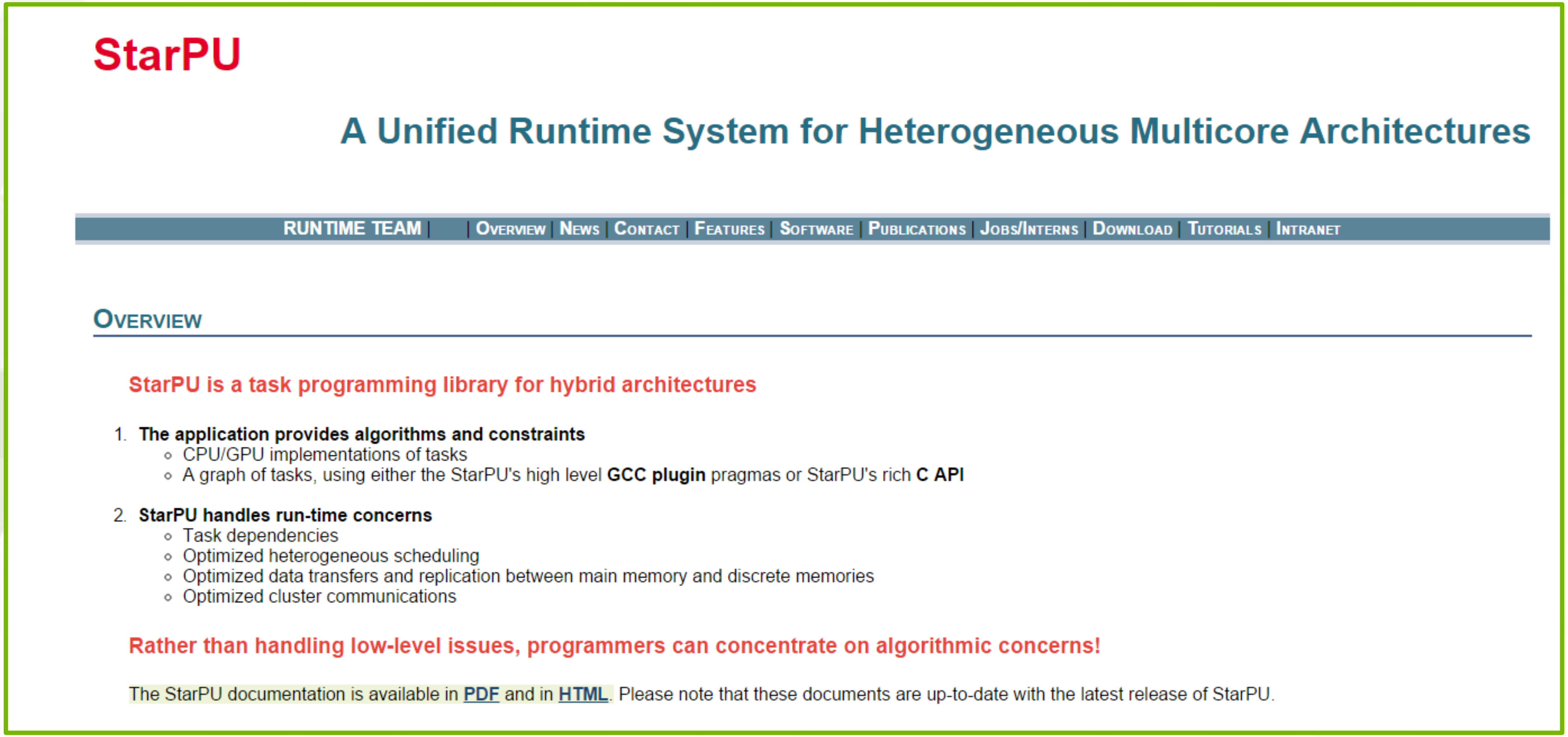
GPU TECHNOLOGY CONFERENCE

Finding more information...

- StarPU Official WebPage:



starpu.gforge.inria.fr



StarPU

A Unified Runtime System for Heterogeneous Multicore Architectures

RUNTIME TEAM | | Overview | News | Contact | Features | Software | Publications | Jobs/Interns | Download | Tutorials | Intranet

Overview

StarPU is a task programming library for hybrid architectures

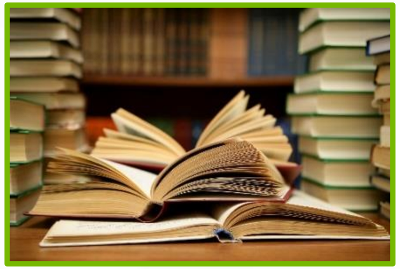
1. **The application provides algorithms and constraints**
   - CPU/GPU implementations of tasks
   - A graph of tasks, using either the StarPU's high level **GCC plugin** pragmas or StarPU's rich **C API**
2. **StarPU handles run-time concerns**
   - Task dependencies
   - Optimized heterogeneous scheduling
   - Optimized data transfers and replication between main memory and discrete memories
   - Optimized cluster communications

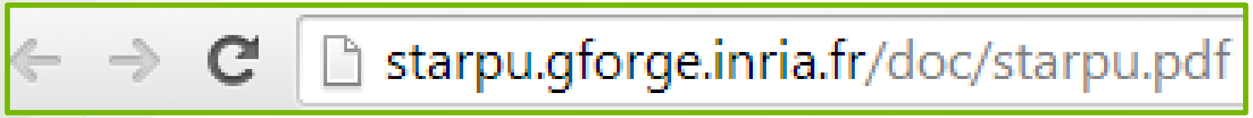
Rather than handling low-level issues, programmers can concentrate on algorithmic concerns!

The StarPU documentation is available in PDF and in HTML. Please note that these documents are up-to-date with the latest release of StarPU.

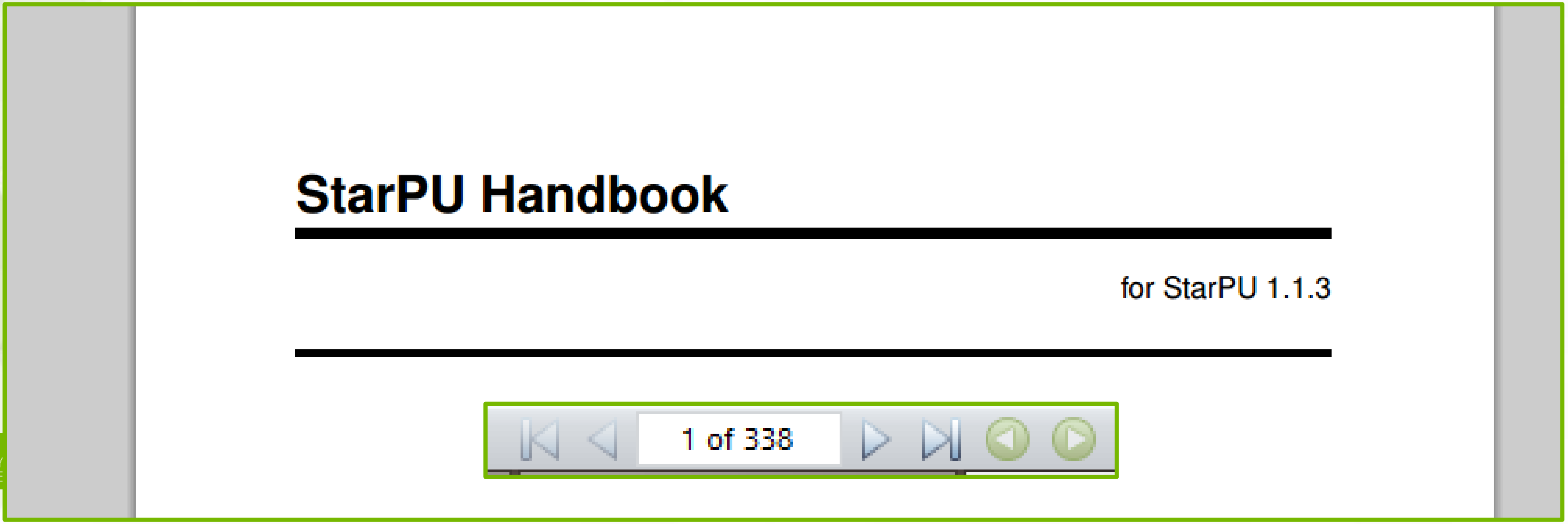
More Information...

- StarPU Handbook:





starpu.gforge.inria.fr/doc/starpu.pdf



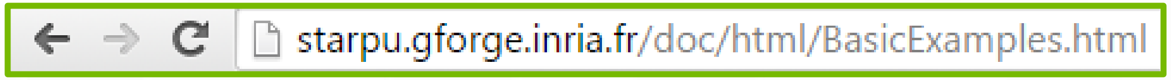
StarPU Handbook

for StarPU 1.1.3

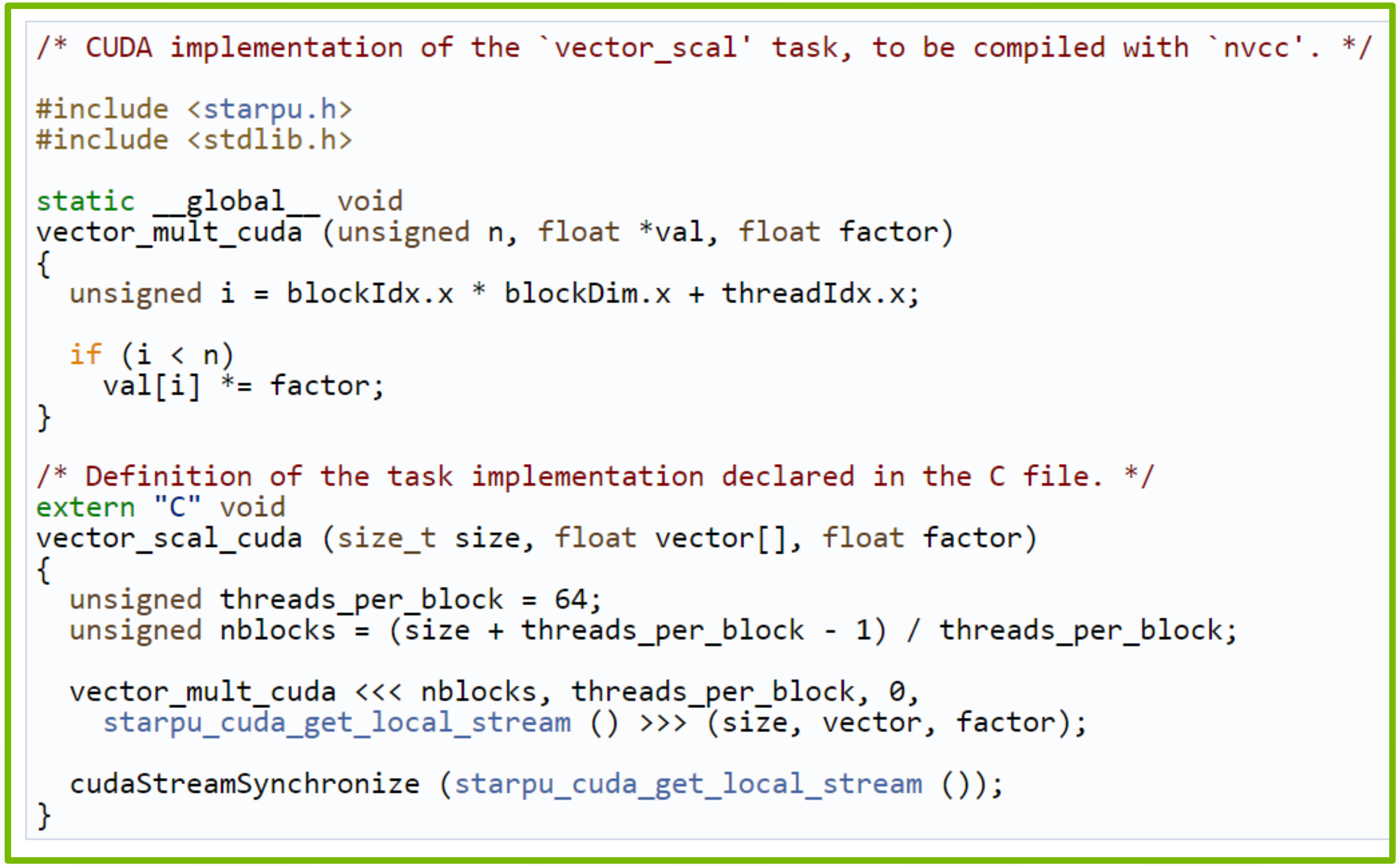
1 of 338

Finding more information...

- StarPU Basic Examples:



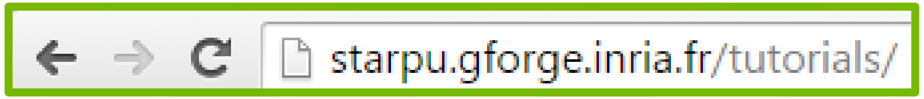
starpu.gforge.inria.fr/doc/html/BasicExamples.html



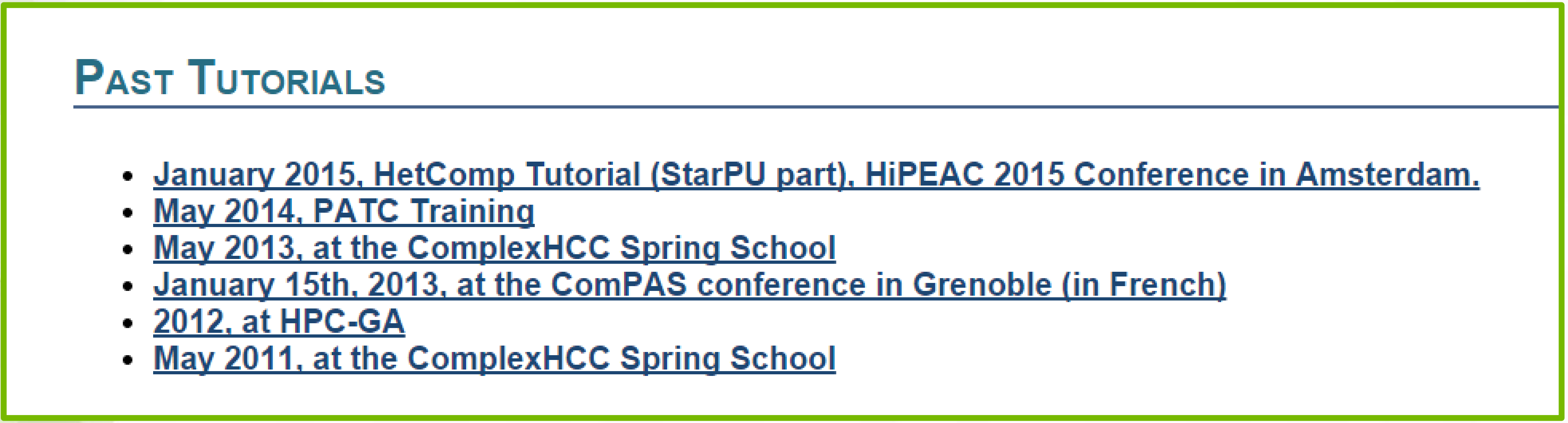
```
/* CUDA implementation of the `vector_scal' task, to be compiled with `nvcc'. */

#include <starpu.h>
#include <stdlib.h>

static __global__ void
vector_mult_cuda (unsigned n, float *val, float factor)
{
  unsigned i = blockIdx.x * blockDim.x + threadIdx.x;

  if (i < n)
    val[i] *= factor;
}

/* Definition of the task implementation declared in the C file. */
extern "C" void
vector_scal_cuda (size_t size, float vector[], float factor)
{
  unsigned threads_per_block = 64;
  unsigned nblocks = (size + threads_per_block - 1) / threads_per_block;

  vector_mult_cuda <<< nblocks, threads_per_block, 0,
    starpu_cuda_get_local_stream () >>> (size, vector, factor);

  cudaStreamSynchronize (starpu_cuda_get_local_stream ());
}
```

Finding more information...

- StarPU Tutorials:



starpu.gforge.inria.fr/tutorials/

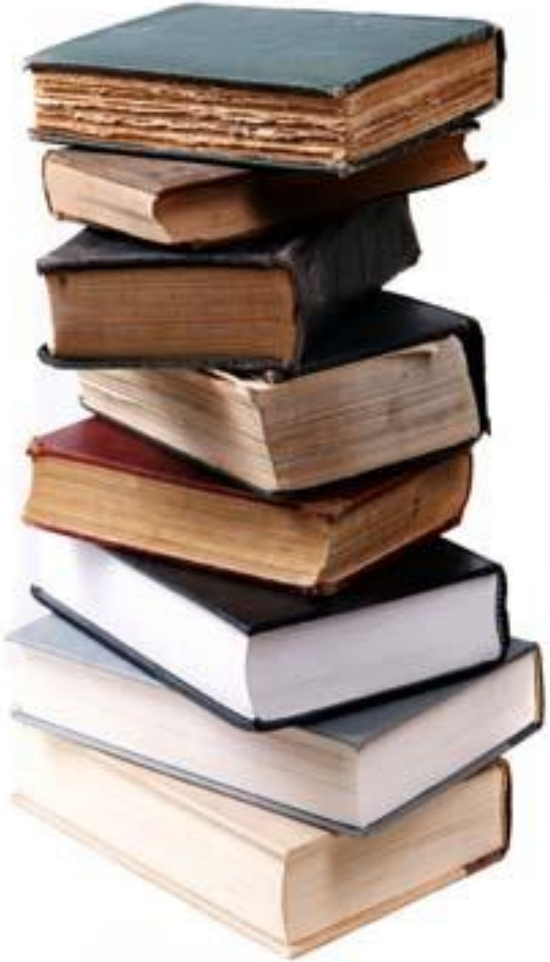


Past Tutorials

- January 2015, HetComp Tutorial (StarPU part), HiPEAC 2015 Conference in Amsterdam.
- May 2014, PATC Training
- May 2013, at the ComplexHCC Spring School
- January 15th, 2013, at the ComPAS conference in Grenoble (in French)
- 2012, at HPC-GA
- May 2011, at the ComplexHCC Spring School

References

1. Augonnet, Cédric, et al. "StarPU: a unified platform for task scheduling on heterogeneous multicore architectures." *Concurrency and Computation: Practice and Experience* 23.2 (2011): 187-198.
2. StarPU Web Site. Available: <http://starpu.gforge.inria.fr/ Access in: 15 jan. 2015.
3. NVIDIA. Available: <http://www.nvidia.com/>. Access in: 15 jan. 2015.
4. StarPU Tutorials. Available: < http://starpu.gforge.inria.fr/tutorials/2015-01-HiPEAC/hipeac_tutorial_hetcomp_starpu_2015.pdf>. Access in: 15 jan. 2015.



Acknowledgements

Thanks

StarPU Team

GPU TECHNOLOGY CONFERENCE

Instituto de Computação uff

FAPERJ
Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

uff mEdiaLab

CAPES

NVIDIA®

THANK YOU

JOIN THE CONVERSATION

#GTC15

gputechconf.com

Contact us at

gazolla@ic.uff.br
esteban@ic.uff.br

GPU TECHNOLOGY CONFERENCE

Please complete the Presenter Evaluation sent to you by email or through the GTC Mobile App. Your feedback is important!

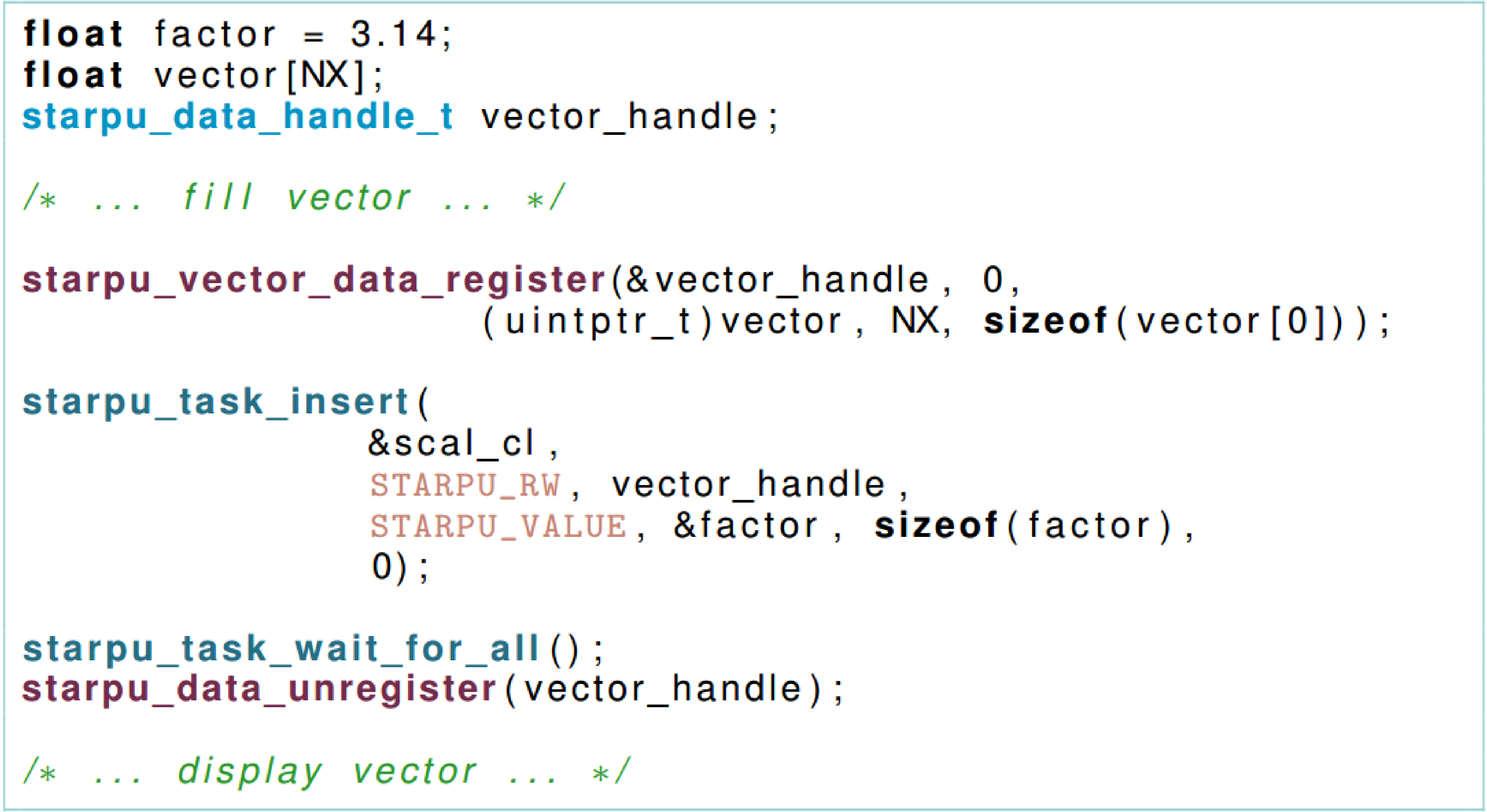
Questions and Answers

Backup Slides Area

StarPU Hello World

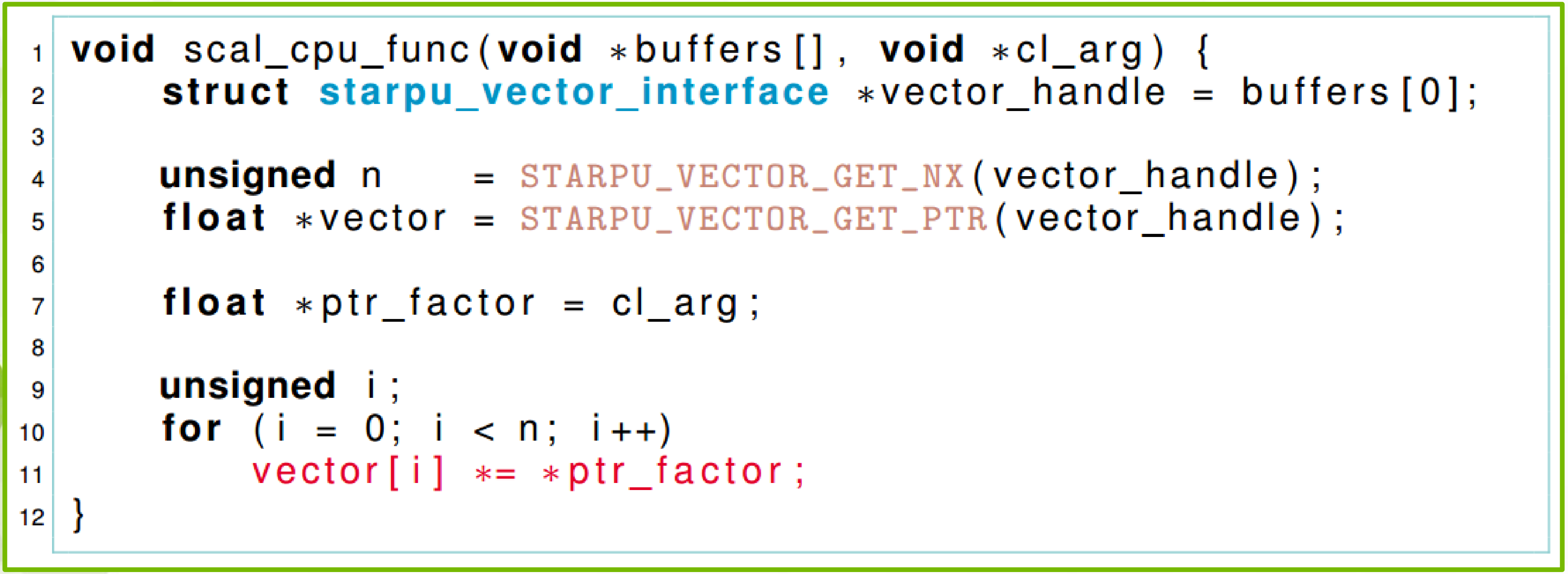
GPU TECHNOLOGY CONFERENCE

Scaling a vector...



```
float factor = 3.14;
float vector[NX];
starpu_data_handle_t vector_handle;

/* ... fill vector ... */

starpu_vector_data_register(&vector_handle, 0,
                     (uintptr_t)vector, NX, sizeof(vector[0]));

starpu_task_insert(
                &scal_cl,
                STARPU_RW, vector_handle,
                STARPU_VALUE, &factor, sizeof(factor),
                0);

starpu_task_wait_for_all();
starpu_data_unregister(vector_handle);

/* ... display vector ... */
```

Scaling a vector...



```c
void scal_cpu_func(void *buffers[], void *cl_arg) {
    struct starpu_vector_interface *vector_handle = buffers[0];

    unsigned n      = STARPU_VECTOR_GET_NX(vector_handle);
    float *vector = STARPU_VECTOR_GET_PTR(vector_handle);

    float *ptr_factor = cl_arg;

    unsigned i;
    for (i = 0; i < n; i++)
        vector[i] *= *ptr_factor;
}
```

MPI Support**

**StarPU Page at http://starpu.gforge.inria.fr/doc/html/

The integration of MPI transfers within task parallelism is done in a very natural way by the means of asynchronous interactions between the application and StarPU. This is implemented in a separate libstarpumpi library which basically provides "StarPU" equivalents of MPI functions.**

StarPU Glossary**

**StarPU Page at http://starpu.gforge.inria.fr/doc/html/

- A **codelet** records pointers to various implementations of the same theoretical function.
- A **memory node** can be either the main RAM or GPU-embedded memory.
- A **bus** is a link between memory nodes.
- A **data handle** keeps track of replicates of the same data (**registered** by the application) over various memory nodes. The data management library manages keeping them coherent.
- The **home** memory node of a data handle is the memory node from which the data was registered (usually the main memory node).
- A **task** represents a scheduled execution of a codelet on some data handles.
- A **tag** is a rendez-vous point. Tasks typically have their own tag, and can depend on other tags. The value is chosen by the application.
- A **worker** execute tasks. There is typically one per CPU computation core and one per accelerator (for which a whole CPU core is dedicated).
- A **driver** drives a given kind of workers. There are currently CPU, CUDA, and OpenCL drivers. They usually start several workers to actually drive them.
- A **performance model** is a (dynamic or static) model of the performance of a given codelet. Codelets can have execution time performance model as well as power consumption performance models.
- A data **interface** describes the layout of the data: for a vector, a pointer for the start, the number of elements and the size of elements ; for a matrix, a pointer for the start, the number of elements per row, the offset between rows, and the size of each element ; etc. To access their data, codelet functions are given interfaces for the local memory node replicates of the data handles of the scheduled task.