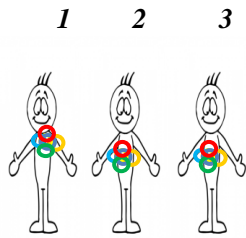


# ***R MEETING/ JUNE 25 2014***

## ***OPTIMIZATION IN R***

***Ndeye Aram GAYE /Jean-Michel BATTO***  
***Metagenopolis***

# MGP pipeline for metagenomic analysis



*individuals*

item	Individuals						
	Ind 1	Ind 2	Ind 3	Ind 4	Ind 5	Ind 6	Ind 7
1	0	36	2	0	43	106	1250
2	0	27	193	0	44	103	8
3	0	31	0	0	0	0	0
4	152	59	282	1	0	0	0
5	115	0	0	1	0	29	2
6	90	783	26	0	2	0	0
7	104	1616	0	0	0	0	5
8	0	82	0	0	0	0	0
9	2	0	0	0	0	0	0
10	23	239	1302	10	0	190	0
11	30	183	900	13	0	172	0
12	27	228	1120	6	0	324	0
13	103	0	0	0	0	0	0
14	0	30	269	0	0	0	0
15	0	0	0	0	0	0	95
16	1250	6002	468	607	492	141	8023
17	0	0	0	0	0	0	0
18	0	9	108	0	0	55	0
19	0	0	0	3	0	0	0
3300000	0	36	2	0	43	106	1250

*genes*

10 M  
↓  
40 M

200 ind  
200000 ind



BiG dAtA  
BiG dAtA



## Language and scientific computing environment

R code: Interpreted language

```
hello<-function()  
  .C("main.so")  
}
```

C code: Compiled language

```
#include <stdio.h>  
hello()  
{  
  printf("hello, world\n");  
}
```

*Binary:*  
01001000, 01100101,  
01101100,  
01101100, 01101111,..

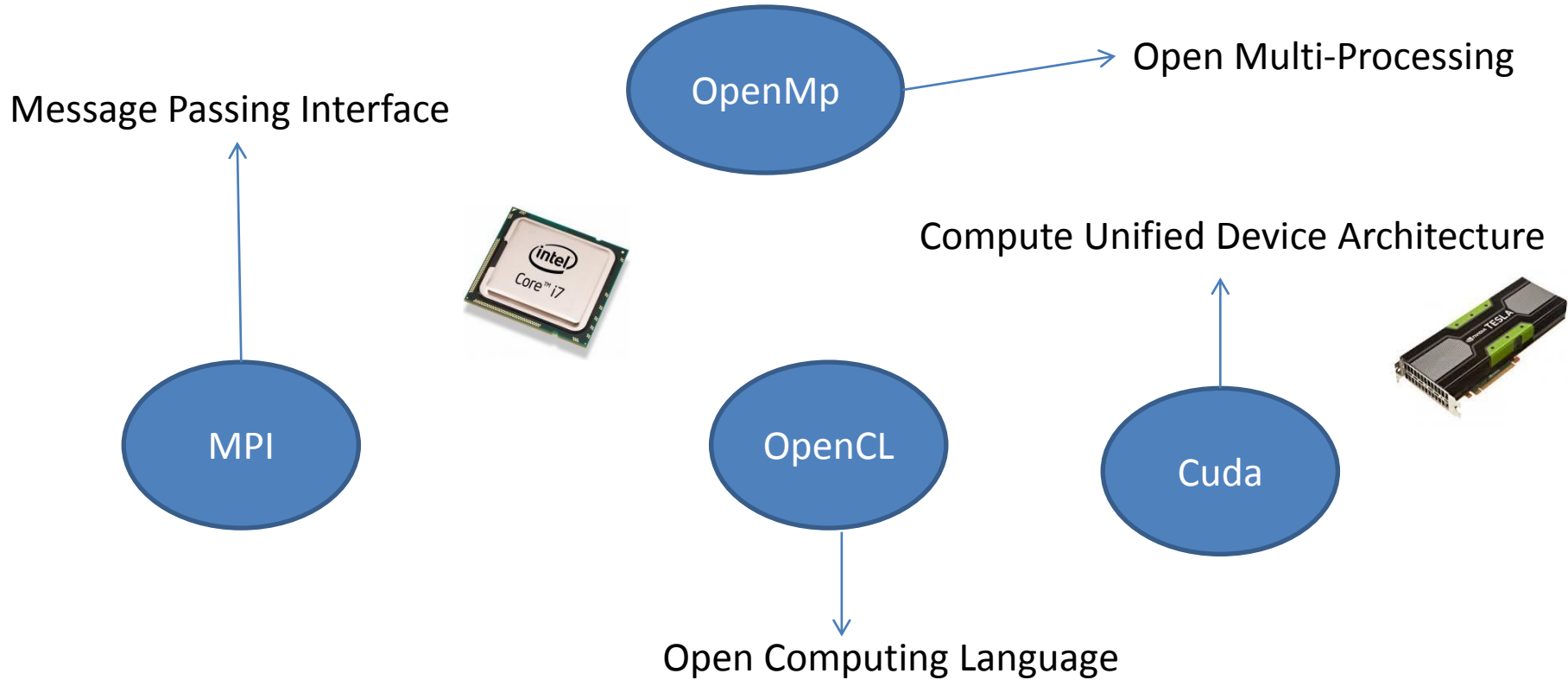
### Interpreted language VS compiled language

Compiled language : source files are **converted** to binary

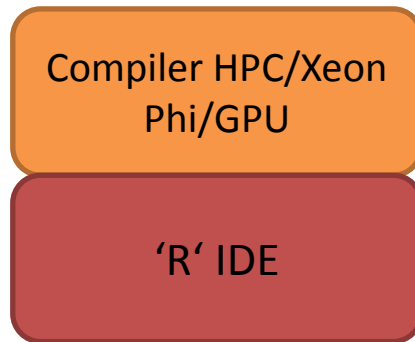
Interpreted language : read but is not compiled.

Compiled language faster than Interpreted language

# User Point of View



✓ ***Complicated for bioanalyst***



- ITEA2 European project
- Duration 3 years

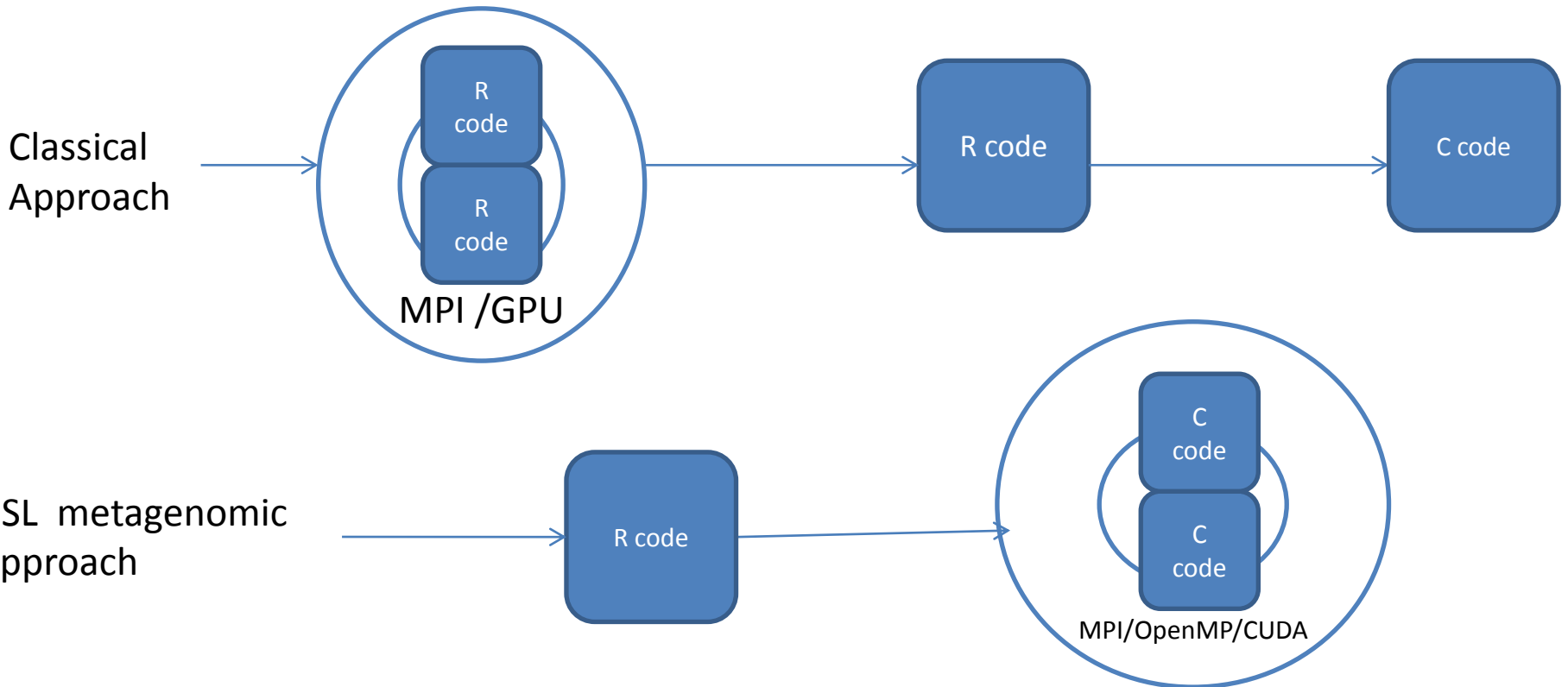
# *Domain Specific Language Applied to Metagenomic*

- DSL: is a programming language whose specifications are dedicated to a specific application domain
- Old fashion optimization : smart library
- New fashion : DSL -> smart compiler

# Strategy

Current situation:

- Need to improve calculation in R (Users)
- Acquiring experience with metagenomic dataset (Me)



# *MACH : New Accelerator*



*Xeon Phi*



- ✓ 61 cores
- ✓ 244 threads
- ✓ Performance up to 1.2 teraflops



✓ MegaPack: Map/Reduce Package (Management of a matrix of 10M)

1 R extension (ready to deliver)

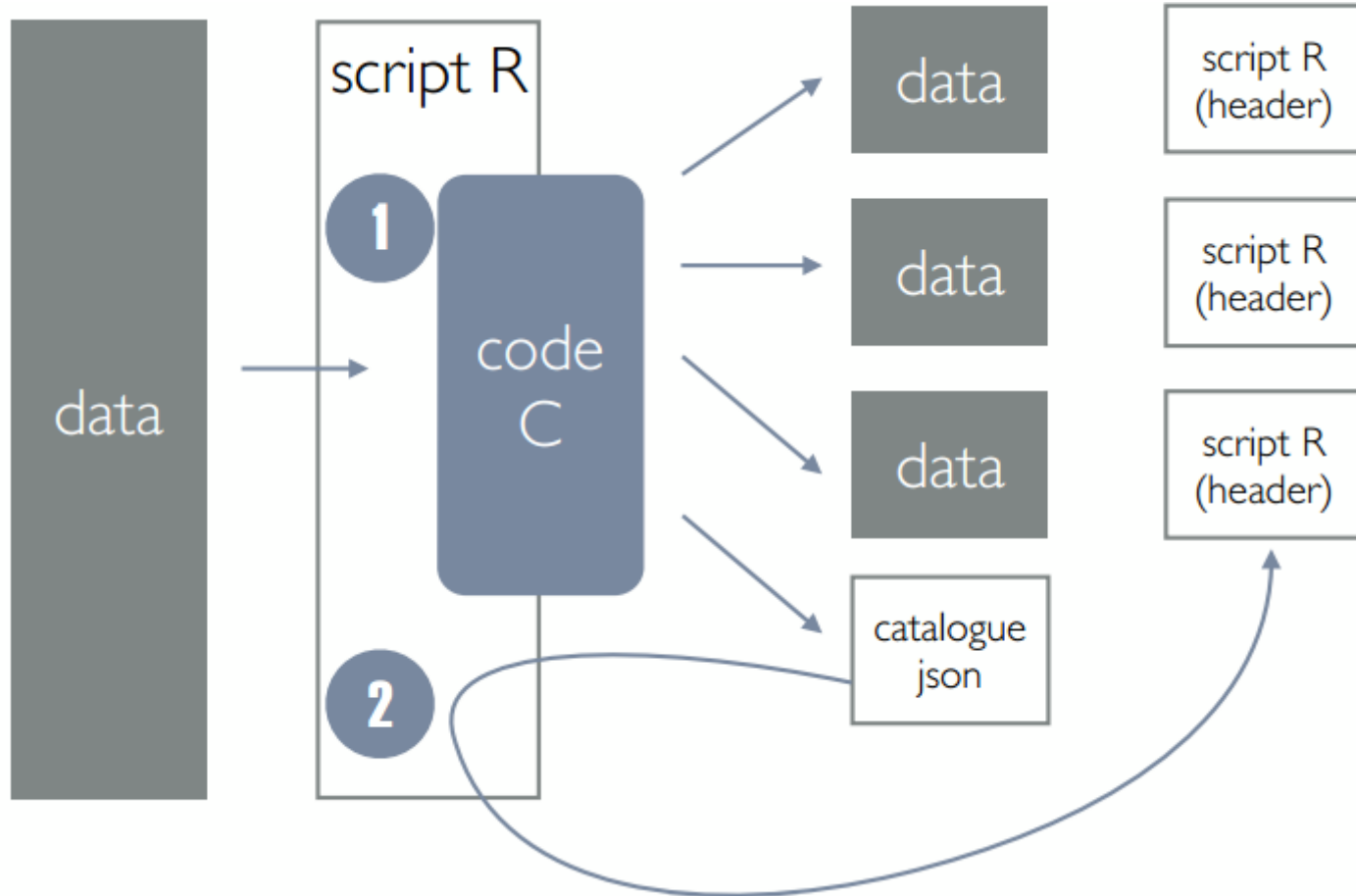
✓ Parconnector: ProActive R API (Computing on Cluster)

12 R extensions (ready to deliver)

✓ GpuStat: calculation of Pearson correlations in gpu (GPU utilisation)

1 R extension (ready to deliver)

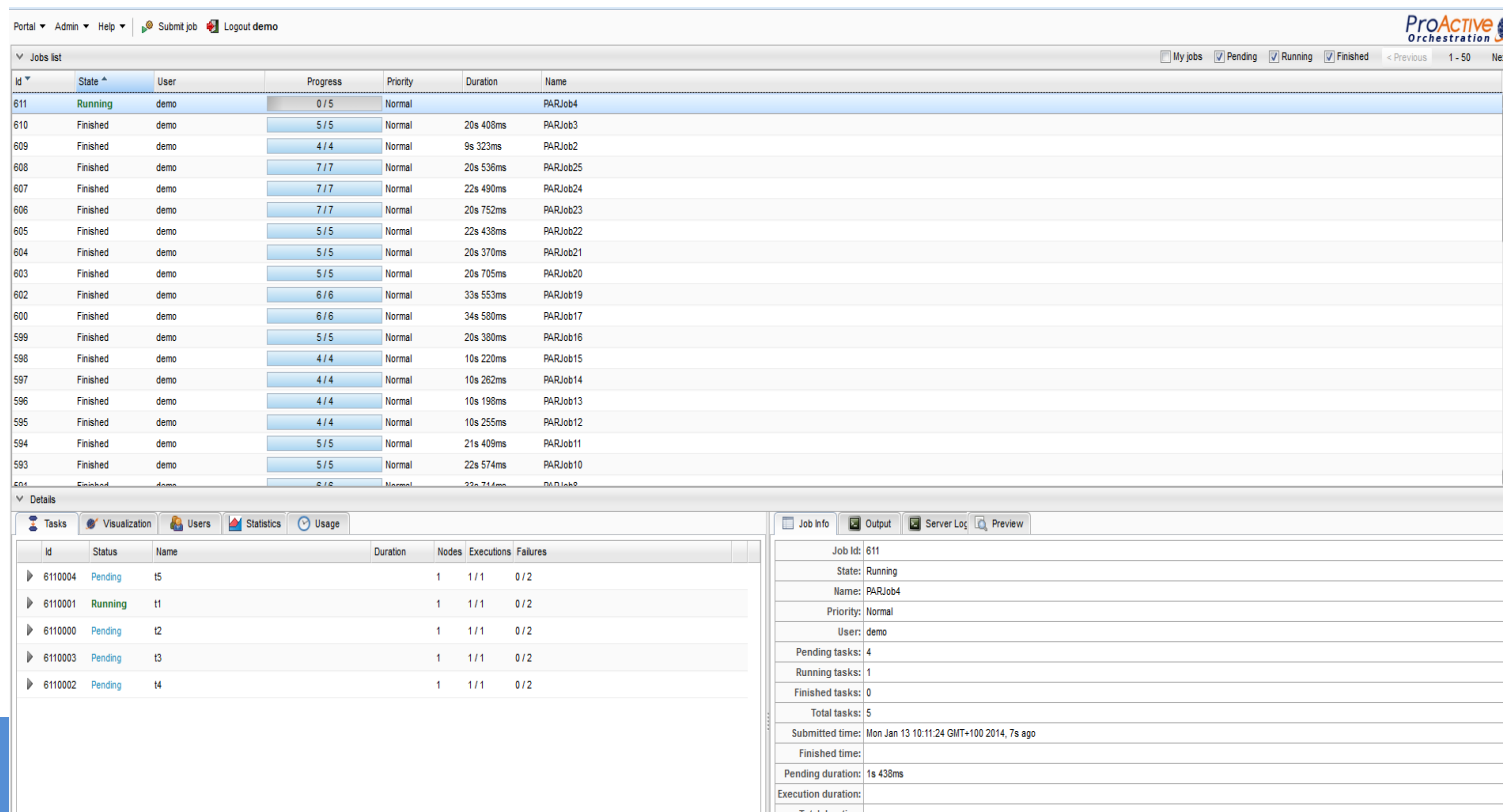
# Megapack R Extension API



# ProActive R Extension API



- Package PARConnector (PAConnect, PASolve, PAWaitFor, ...)  
Connection with ProActive Scheduler
- To hide scheduler complexity



The screenshot displays the ProActive Scheduler web interface. At the top, there is a navigation bar with 'Portal', 'Admin', 'Help', 'Submit job', and 'Logout demo'. The main area is titled 'Jobs list' and contains a table of jobs. The table has columns for 'Id', 'State', 'User', 'Progress', 'Priority', 'Duration', and 'Name'. The current job, PARJob4 (Id: 611), is in a 'Running' state with a progress of 0/5. Below the jobs list, there is a 'Details' section for the selected job. This section is divided into two panes: 'Tasks' and 'Job Info'. The 'Tasks' pane shows a list of tasks with columns for 'Id', 'Status', 'Name', 'Duration', 'Nodes', 'Executions', and 'Failures'. The 'Job Info' pane provides summary statistics for the job, including the number of pending, running, and finished tasks, and the total number of tasks.

Id	State	User	Progress	Priority	Duration	Name
611	Running	demo	0 / 5	Normal		PARJob4
610	Finished	demo	5 / 5	Normal	20s 408ms	PARJob3
609	Finished	demo	4 / 4	Normal	9s 323ms	PARJob2
608	Finished	demo	7 / 7	Normal	20s 536ms	PARJob25
607	Finished	demo	7 / 7	Normal	22s 490ms	PARJob24
606	Finished	demo	7 / 7	Normal	20s 752ms	PARJob23
605	Finished	demo	5 / 5	Normal	22s 438ms	PARJob22
604	Finished	demo	5 / 5	Normal	20s 370ms	PARJob21
603	Finished	demo	5 / 5	Normal	20s 705ms	PARJob20
602	Finished	demo	6 / 6	Normal	33s 553ms	PARJob19
600	Finished	demo	6 / 6	Normal	34s 580ms	PARJob17
599	Finished	demo	5 / 5	Normal	20s 380ms	PARJob16
598	Finished	demo	4 / 4	Normal	10s 220ms	PARJob15
597	Finished	demo	4 / 4	Normal	10s 262ms	PARJob14
596	Finished	demo	4 / 4	Normal	10s 196ms	PARJob13
595	Finished	demo	4 / 4	Normal	10s 255ms	PARJob12
594	Finished	demo	5 / 5	Normal	21s 409ms	PARJob11
593	Finished	demo	5 / 5	Normal	22s 574ms	PARJob10
601	Finished	demo	6 / 6	Normal	22s 714ms	PARJob8

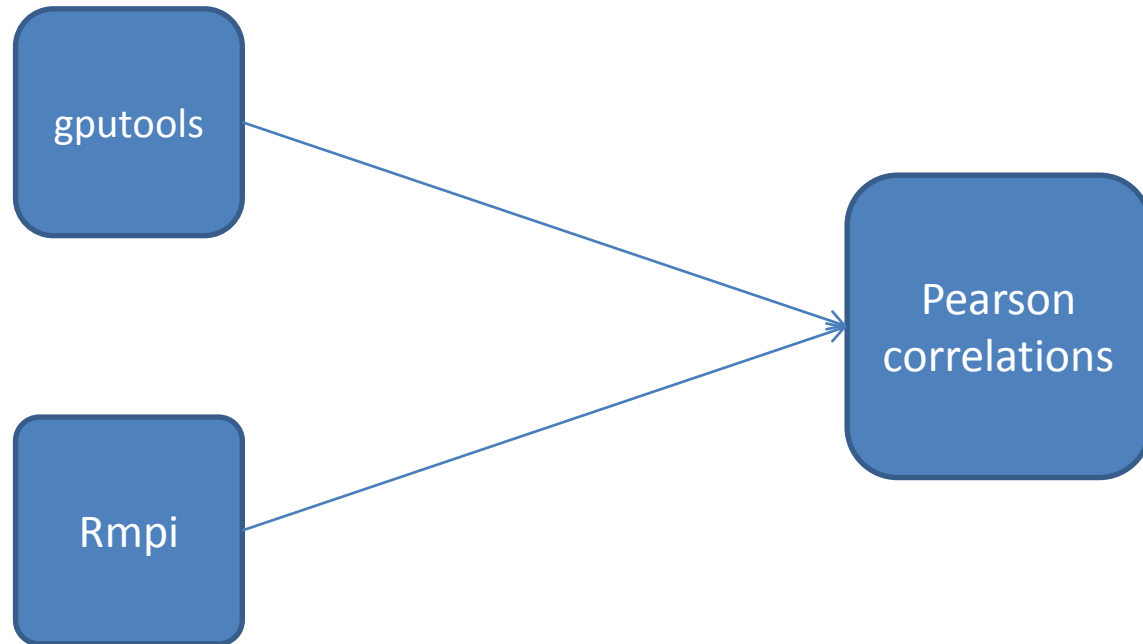
Id	Status	Name	Duration	Nodes	Executions	Failures
6110004	Pending	t5		1	1 / 1	0 / 2
6110001	Running	t1		1	1 / 1	0 / 2
6110000	Pending	t2		1	1 / 1	0 / 2
6110003	Pending	t3		1	1 / 1	0 / 2
6110002	Pending	t4		1	1 / 1	0 / 2

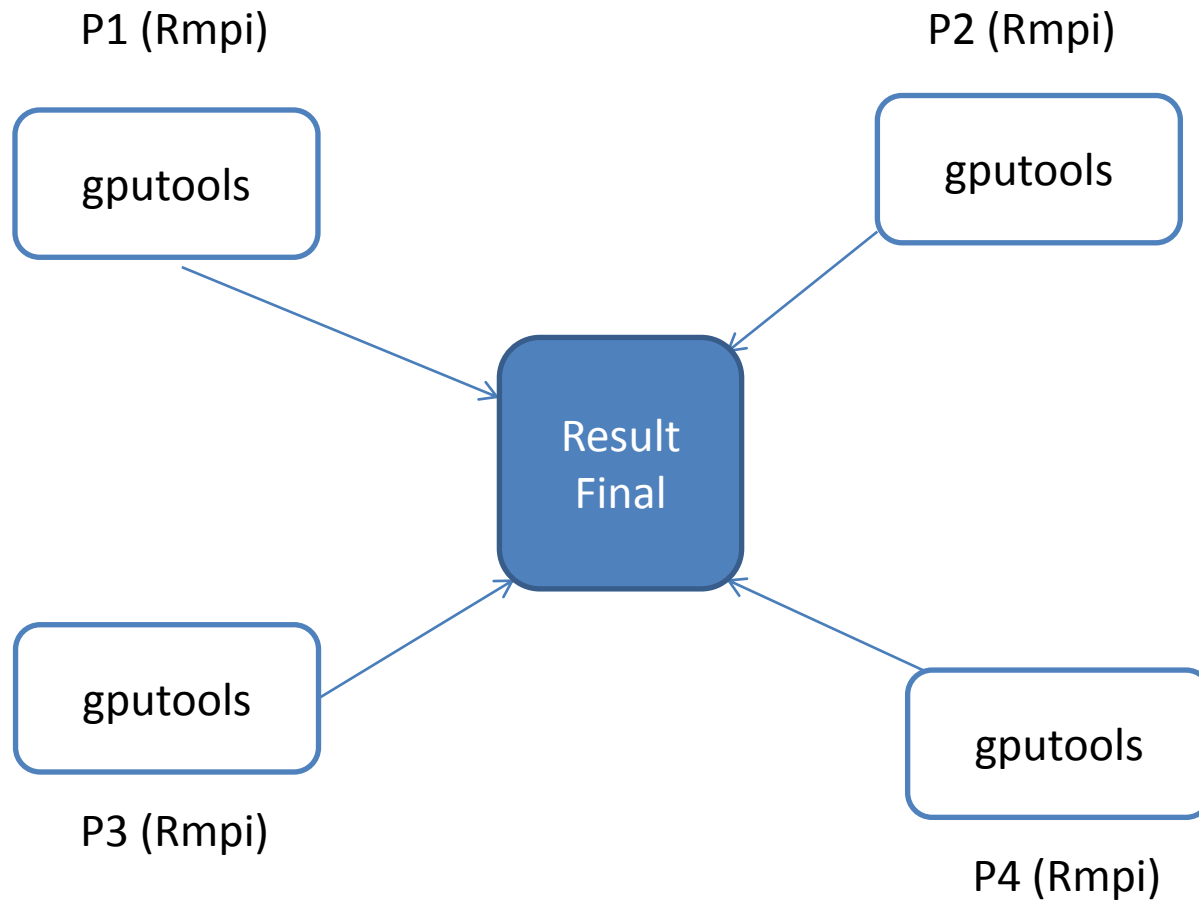
Job Info	Output	Server Log	Preview
Job Id:	611		
State:	Running		
Name:	PARJob4		
Priority:	Normal		
User:	demo		
Pending tasks:	4		
Running tasks:	1		
Finished tasks:	0		
Total tasks:	5		
Submitted time:	Mon Jan 13 10:11:24 GMT+100 2014, 7s ago		
Finished time:			
Pending duration:	1s 438ms		
Execution duration:			
Total duration:			

# *gpuStat R Extension API*

Based on 2 existing R packages

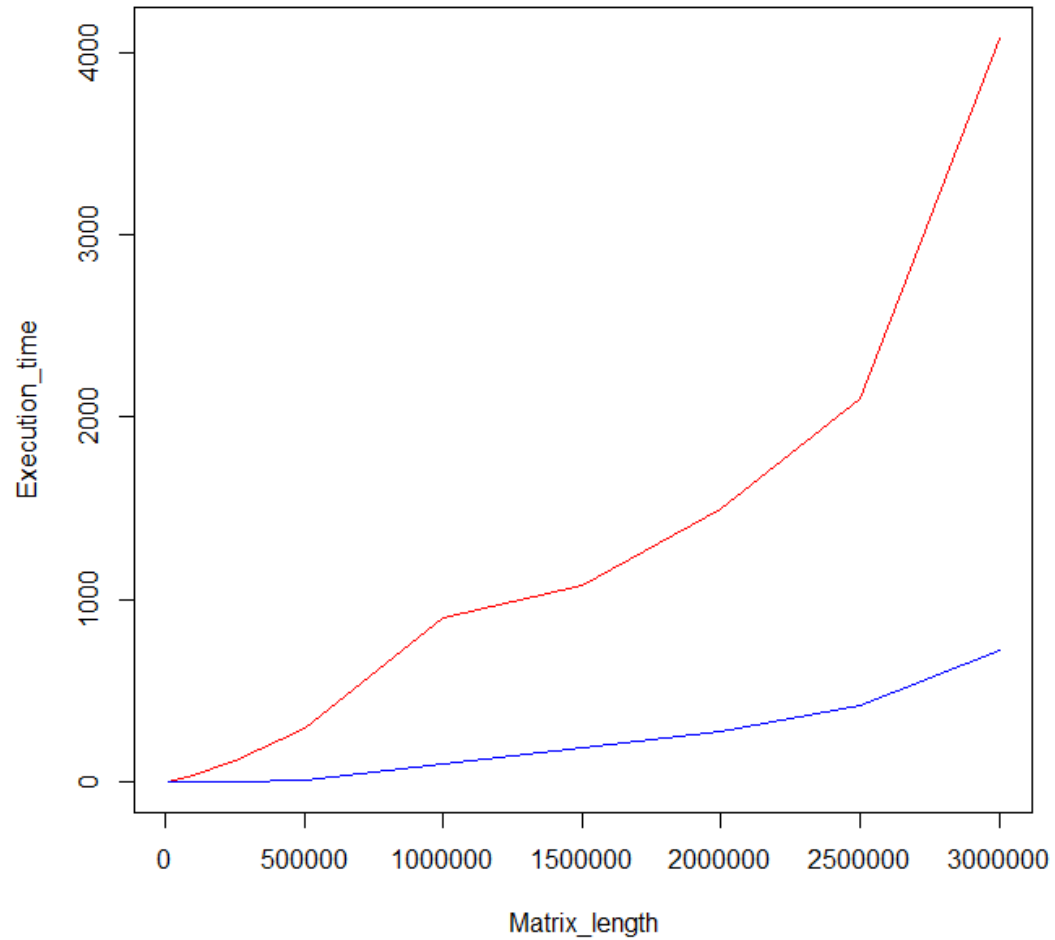


# *gpuStat*



## SOME BENCHMARKS

CPU VS GPU



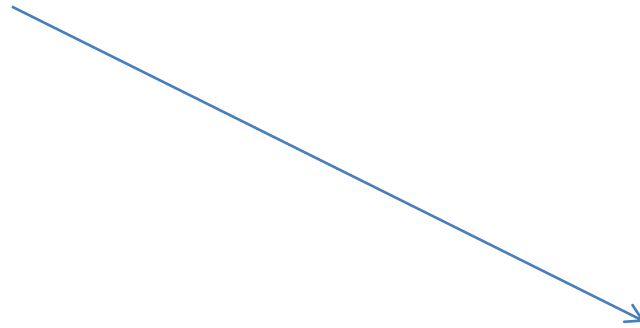
## *R Use Case Applications*

- ✓ *MetaOmineR GPU edition (E Le Chatelier, E Prifti):* data analysis of quantitative metagenomics
  
- ✓ *Bayesian Builder (J Abou Ghantous, J Tap):* bayesian estimates optimization applied to data from the study of the human intestinal microbiota.

## *Beyond MACH (HPC)*

Nahid Emad  
Professor, University of Versailles  
Resp. MIHPS ([mihps.prism.uvsq.fr](http://mihps.prism.uvsq.fr))  
PRISM Laboratory  
Maison de la Simulation

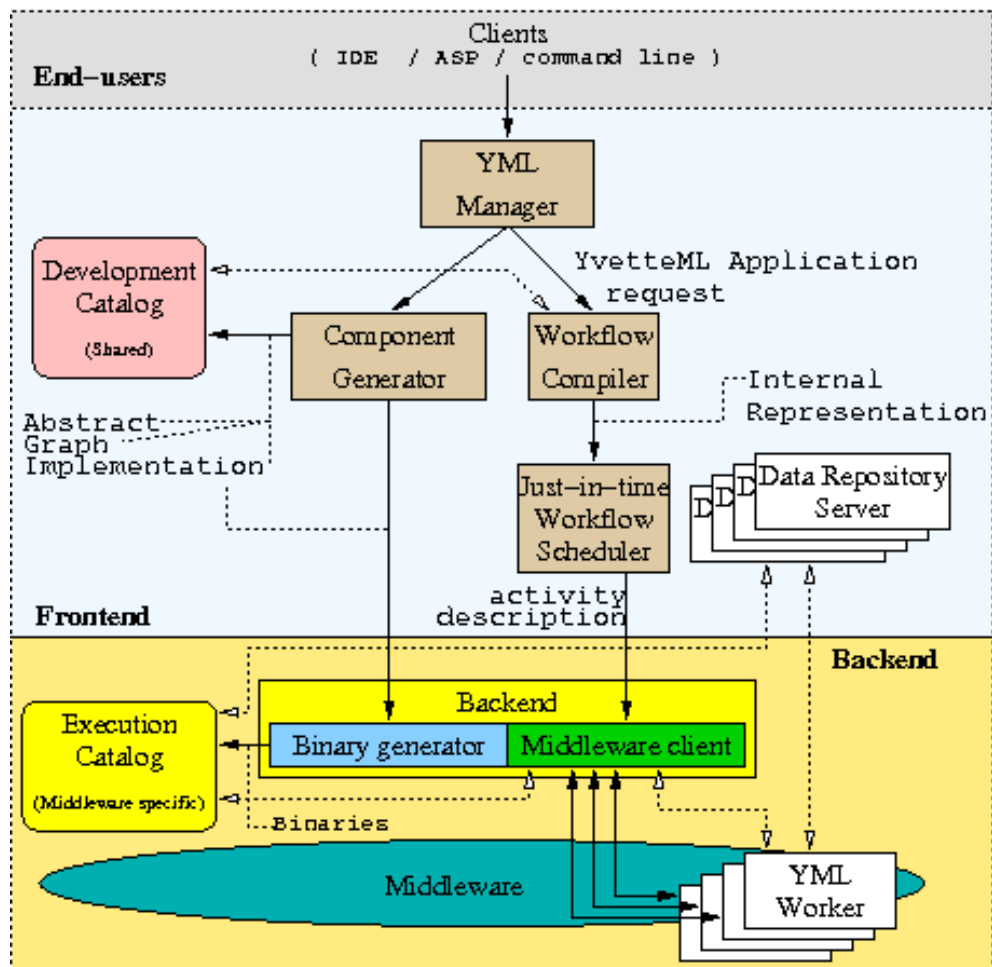
Datacenter: YML  
(YvetteLanguage)



My thesis



# YML ([yml.prism.uvsq.fr](http://yml.prism.uvsq.fr))



## *Conclusion*

- ✓ Our strategy allows us to start quickly
- ✓ Next milestone R Xeon Phi package
- ✓ Next milestone R in YML

# Thanks

*Jad Abou Ghantous*  
*Anne-Sophie Alvarez*  
*Jean-Michel Batto*  
*Amine Ghozlane*  
*Vincent Heuschling*  
*Emmanuelle Le Chatelier*  
*Pierre Léonard*  
*Nicolas Pons*  
*Edi Prifti*  
*Julien Tap*

