On Distributed R Computations over BOINC

Alexander Rumyantsev, **Anna Eparskaya**

29.08.2017

BOINC:FAST

Brief

1. Embarrassingly parallel applications
2. Usecase: RSLURM package
3. The RBOINC: concept and future plans

Embarrassingly parallel

A class of loosely coupled tasks that allow approximately linear speedup, being executed in parallel.
Examples:

- Rendering of computer graphics;
- Computer simulations;
- Distributed relational database;
- Evolutionary computation metaheuristics: genetic algorithms;
- NP-hard problems: scheduling etc;
- Monte-Carlo simulation;
- Map-Reduce type of problems.

Case: stochastic modeling

Example: stochastic modeling of a queueing system. We are interested in expected stationary performance of the system, but no analytical solution. Then we use simulations.
Consider an open queueing system with an input of tasks, each having random parameters (service time, size etc.).

1. Generate random driving sequence (task parameters, time of arrival etc.)
2. Evaluate the service process.
3. Calculate performance measure over the trajectory.

Repeat many times to obtain the desired accuracy, and calculate the simple average, which converges to the expectation.
This task is *perfectly parallel*.

The apply functions family in R

Alternative to the for loops.

Returns a vector or array or list of values obtained by applying a function to margins of an array or matrix.

+ Vectorized

```
    > sapply(1:3, function(x){sin(x)+cos(x)^2})
[1]  1.1333976  1.0824756  1.1212052 -0.3295525 -0.8784600  0.6425115
[7]  1.2253552  1.0105285  1.2422768  0.1600199
```

High-performance computing in R

Some basic options:

- SNOW — build a cluster of workstations
- multicore — (archive) execute on machines with multiple cores or CPUs
- parallel — = SNOW + multicore
- GridR — use the existing GRID (Condor, OpenScienceGrid)

`parallel`:

1. Each task starts in a separate process
2. Each process executes on a separate node
3. Processes exchange using sockets: get expressions, return the results of calculations

Contra: low-level solution

The best known solution: RSLURM

`apply`+ RParallel=?

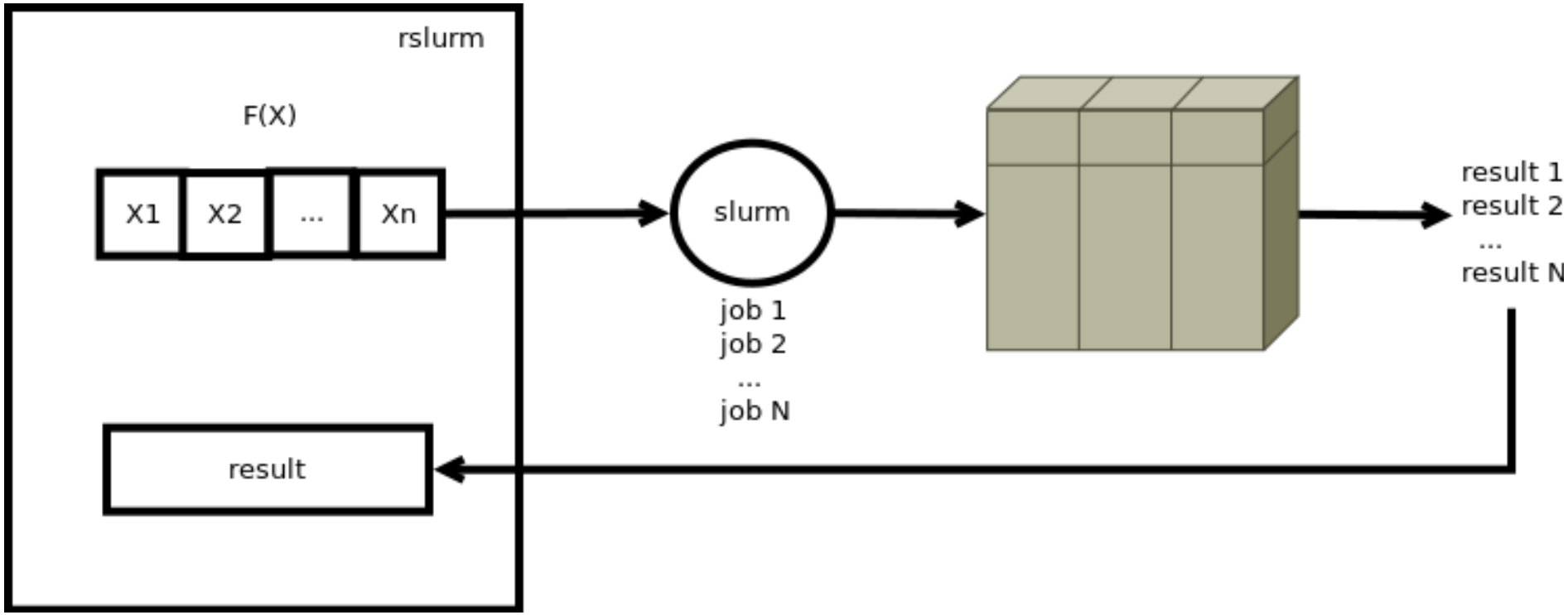
RSLURM

Functions that simplify submitting R scripts to a 'SLURM' cluster workload manager, in part by automating the division of embarrassingly parallel calculations across cluster nodes. (package description)

How to start working on cluster seamlessly

The idea behind

Decompose the parameter space, run via `sbatch`, asynchronous mode, get the results back.



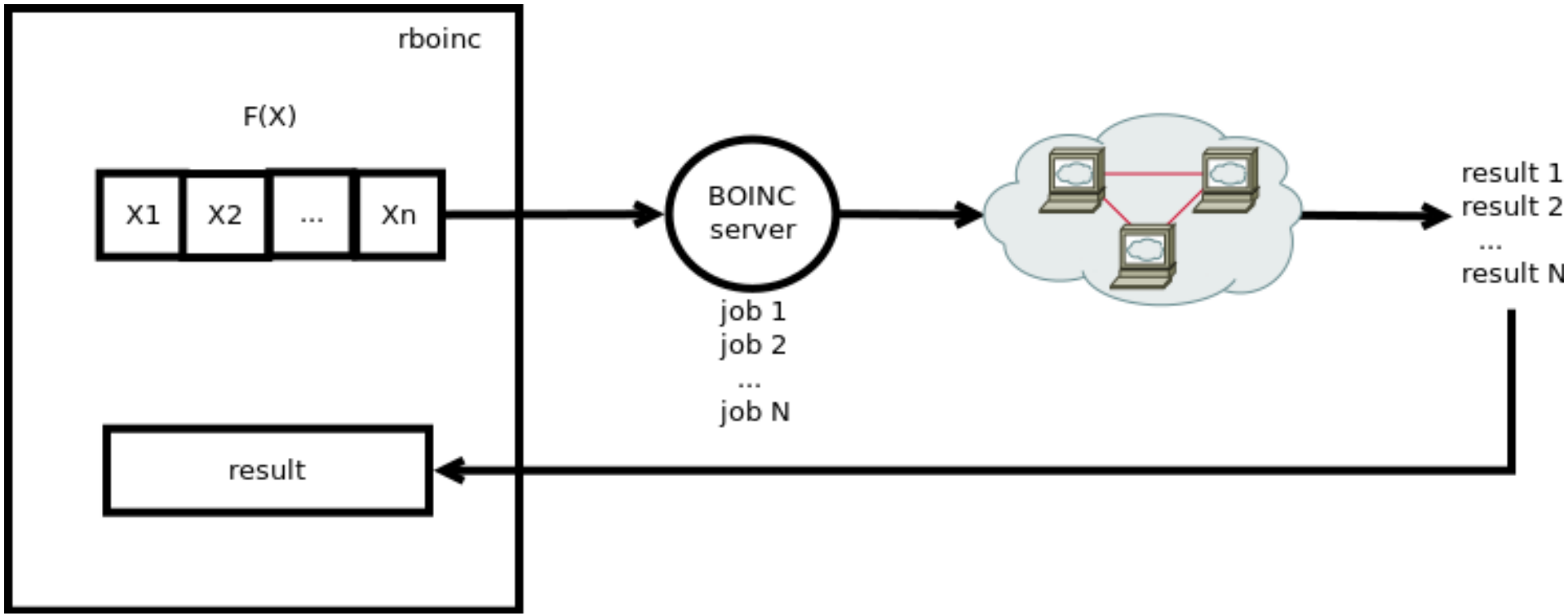
The drawback of this approach

Why use high-performance cluster? The tasks are totally independent and asynchronous!

Why not use BOINC?

The RBOINC concept

Same idea, other platform.



Current state and future

RBOINC owner at R-forge agreed to collaborate.

RSLURM and GridR as the starting point.

2-year Master study.

Thank you for attention!

Eparskaya Anna
Petrozavodsk State University (PetrSU)
anna1995e@yandex.ru

RSCI:
ResearcherID: N-7634-2017

ORCID: 0000-0003-0894-2169