University of Zurich[UZH]

S3IT

Overview of common GC3Pie use cases

Riccardo Murri <riccardo.murri@uzh.ch>

S3IT: Services and Support for Science IT

University of Zurich

July 11–14, 2016

What is GC3Pie?

GC3Pie is . . .

1. An *opinionated* Python framework for defining and running computational workflows;
2. A *rapid development toolkit* for running user applications on clusters and IaaS cloud resources;
3. The worst name ever given to a middleware piece. . .

As *developers*, you're mostly interested in this part.

Uses of GC3Pie

Uses of GC3Pie: parameter sweep

You have a simulation code that is dependent on a number of parameters.

Run the code for all possible combinations of parameters.

Then collect all the outputs and post-process to get a statistical overview.

Uses of GC3Pie: model calibration

You have a simulation code that is dependent on a number of parameters.

Run the code for all possible combinations of parameters, and find the ones that “best” approximate a given experimental result.

Uses of GC3Pie: parallel processing

Run the same program over and over again, feeding it different input files each time.

Then collect all the outputs and post-process to get a statistical overview.

Uses of GC3Pie: parallel processing

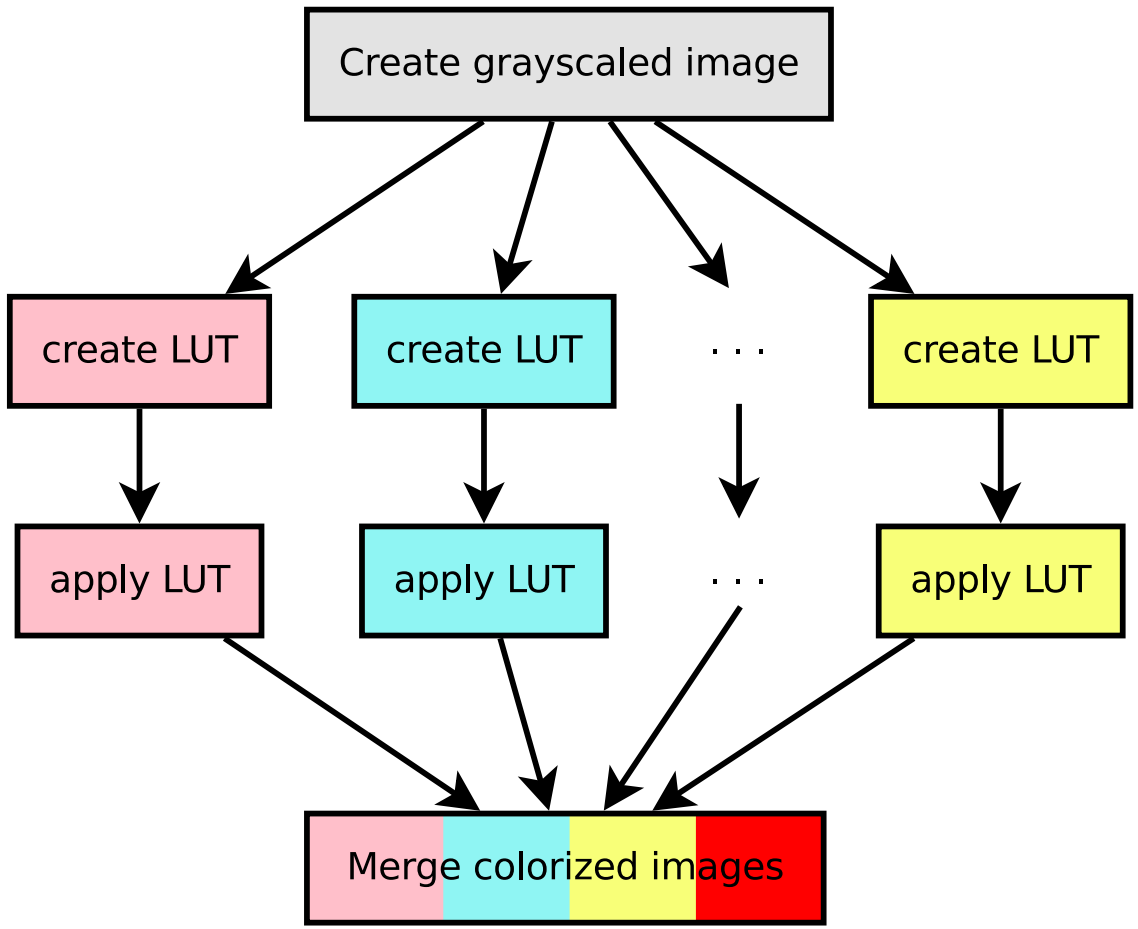
(At times, you chop a large input file into pieces and process each one separately instead.)

> *"For example, say we have a de novo assembly of 100,000 contigs. If we run 1 BLAST job against NR it could take as long as 50,000 minutes/35 days!! (30sec/query sequence), however if we split this job into subsets of 5,000 sequences and ran 20 jobs in "parallel" on a cluster, our total run-time is reduced to only 41 hours."*

Reference: http://sfg.stanford.edu/BLAST.html

Uses of GC3Pie: workflows

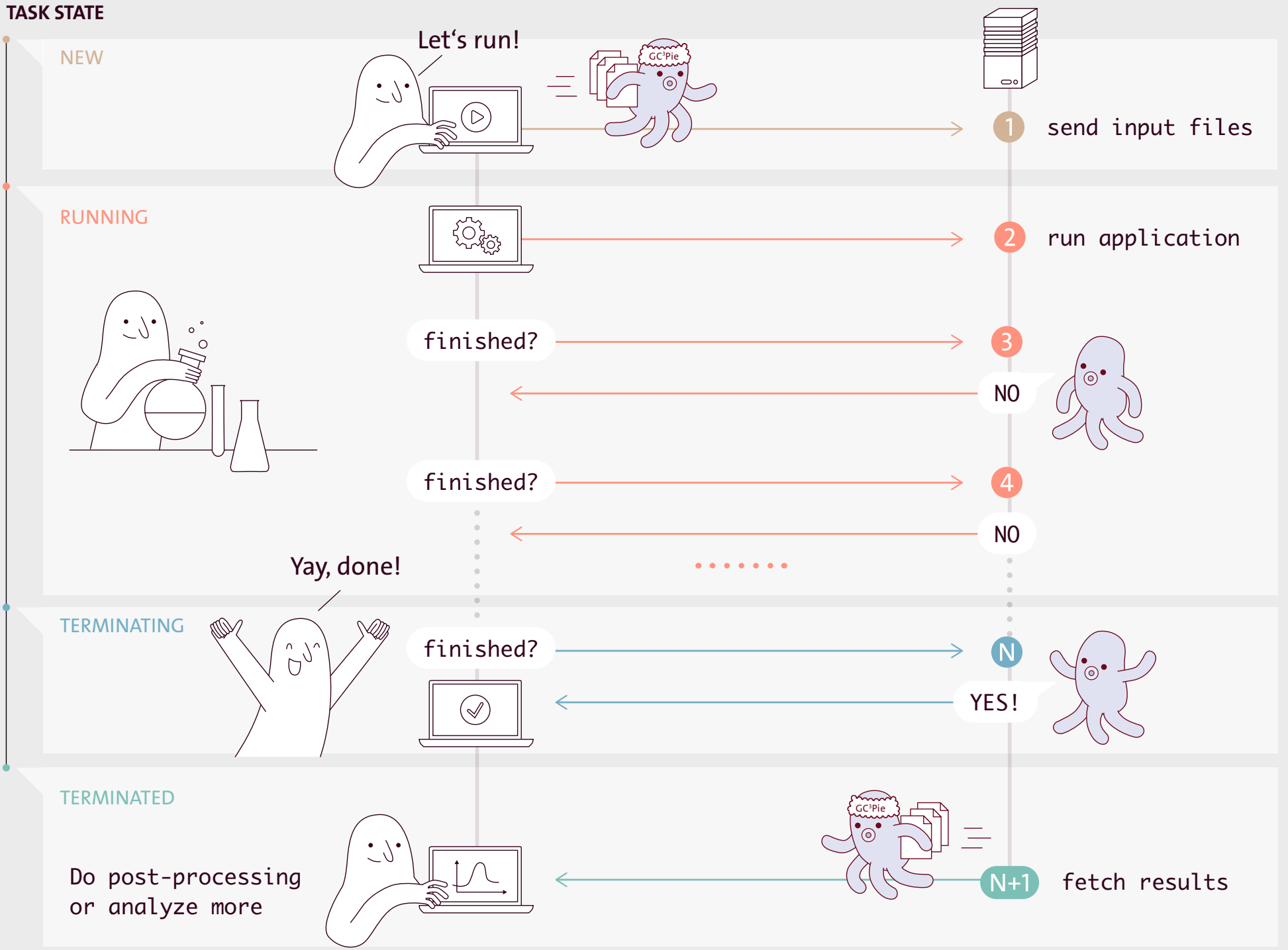
Orchestrate execution of several applications: some steps may run in parallel, some might need to be sequenced.

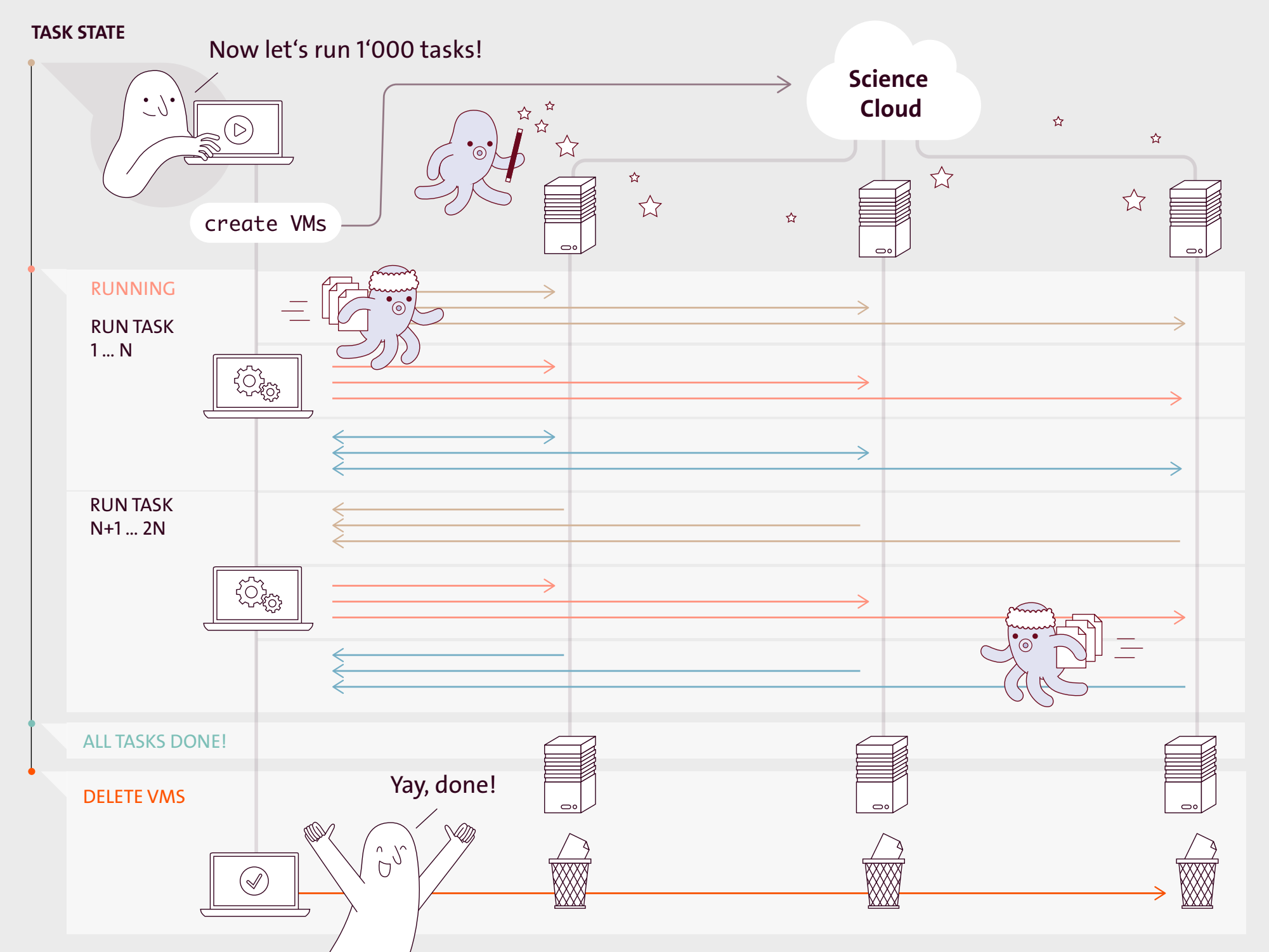


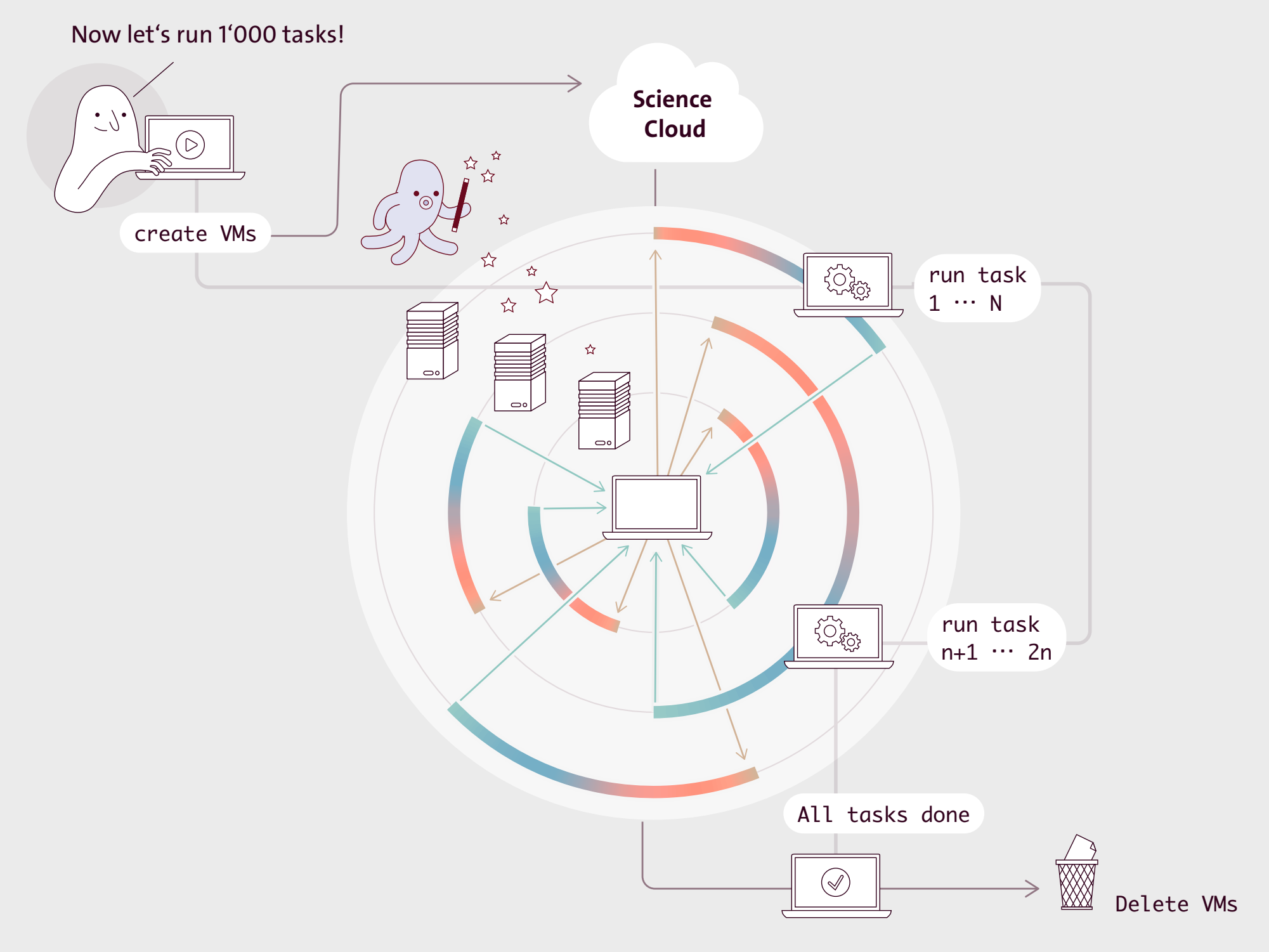
How GC3Pie works

A typical high-throughput script structure

1. Initialize computational resources
2. Prepare programs and inputs for submission
3. Submit tasks
4. Monitor task status (loop)
5. Retrieve results
6. Postprocess and display


TASK STATE
NEW
Let‘s run!
GC³Pie
1
send input files
RUNNING
2
run application
finished?
3
NO
finished?
4
NO
Yay, done!
TERMINATING
finished?
N
YES!
TERMINATED
Do post-processing
or analyze more
GC³Pie
N+1
fetch results


TASK STATE
Now let‘s run 1‘000 tasks!
Science Cloud
create VMs
RUNNING
RUN TASK
1 ... N
RUN TASK
N+1 ... 2N
ALL TASKS DONE!
DELETE VMS
Yay, done!


Now let‘s run 1‘000 tasks!
Science Cloud
create VMs
run task 1 ⋯ N
run task n+1 ⋯ 2n
All tasks done
Delete VMs

What GC3Pie handles for you

1. Resource allocation (e.g. starting new instances on ScienceCloud)
2. Selection of resources for each application in the session
3. Data transfer (e.g. copying input files in the new instances)
4. Remote execution of the application
5. Retrieval of results (e.g. copying output files from the running instance)
6. De-allocation of resources