Ensemble Toolkit

http://radical-cybertools.github.com
http://radical.rutgers.edu

Ensemble-based Applications (EBAs)

- Ensemble: a set of multiple units of execution (ensemble members)
- Degrees of freedom in EBAs: Interactions, dependencies, heterogeneity among ensembles and ensemble members
- Many applications in molecular science, climate science, **seismology,** astrophysics are ensemble-based

Ensemble Toolkit: Overview

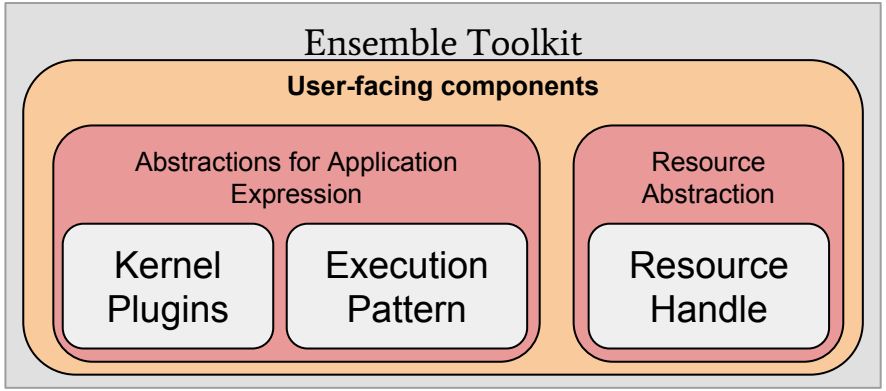
- **Problem space**: Toolkits enabling flexible and scalable execution of EBAs on HPCs are required
- Existing solutions:
  - Customized scripts: fragile, unportable, high user effort in task and resource management
  - End-to-end workflows: rigid, prolonged development/ modification time.

Ensemble Toolkit: Overview

- **Ensemble toolkit**:
  - Python-based API
  - Components to build EBAs
    - sweetspot between custom scripts and end-to-end workflow systems ?
  - Responsibility of task, data and resource management
  - Tested up to O(1000) tasks on multiple HPCs
  - will soon support asynchronous and adaptive scenarios*

Ensemble Toolkit: Basic Components

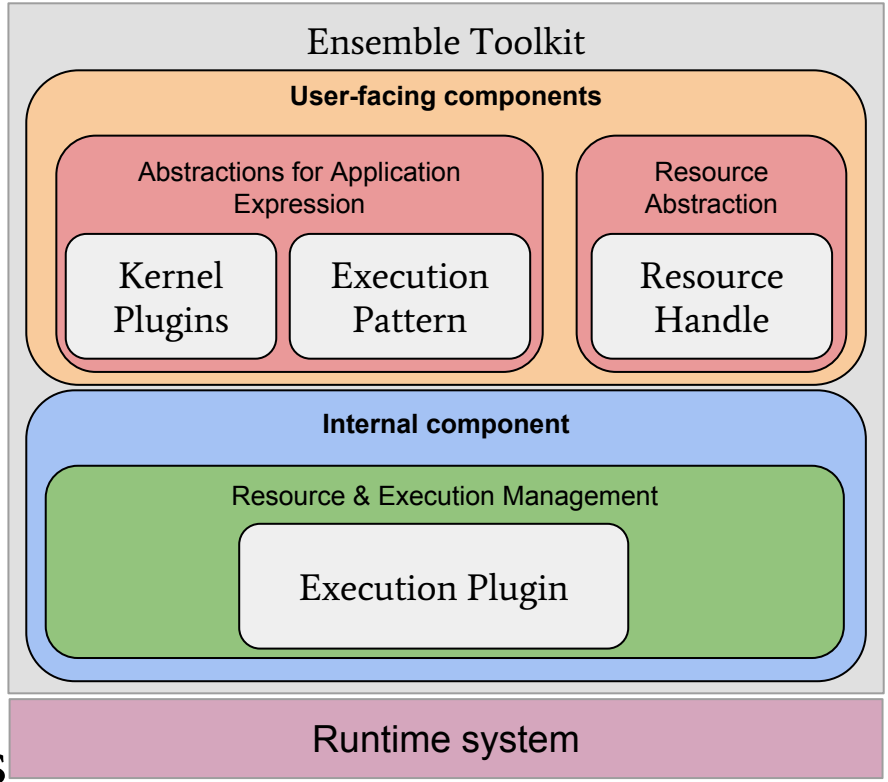
- User expresses application and specifies resource
- **Kernel plugins**: Abstraction of a compute task

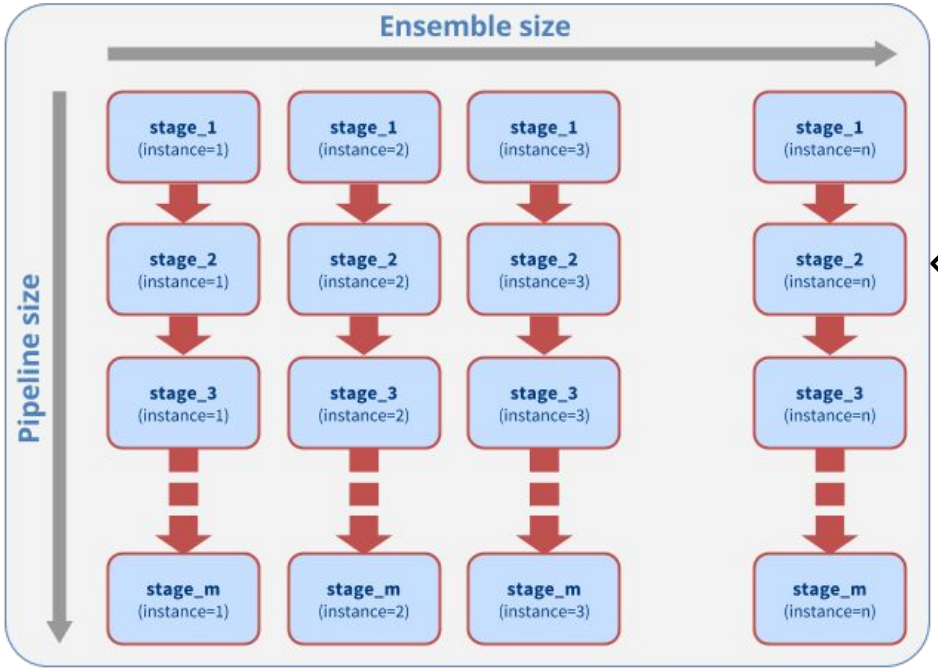


- **Execution pattern**: parameterized template representing communication & synchronization
- **Resource Handle**: Allows users to allocate, execute, deallocate resources.

Ensemble Toolkit: Basic Components

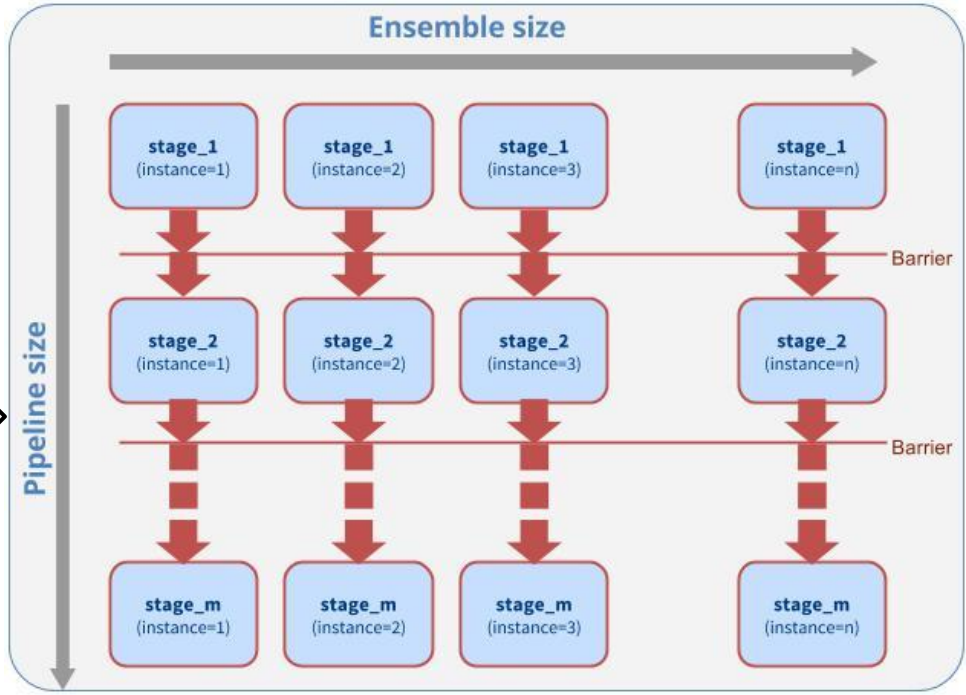
- **Execution plugin**: translates the tasks into executable units for the specific resource + executes via Runtime system
- **Runtime system** (RADICAL Pilot): provides placeholder jobs for application level control of resources and supports MPI applications, multiple heterogeneous systems





Ensemble of Pipelines

Pipeline of Ensembles



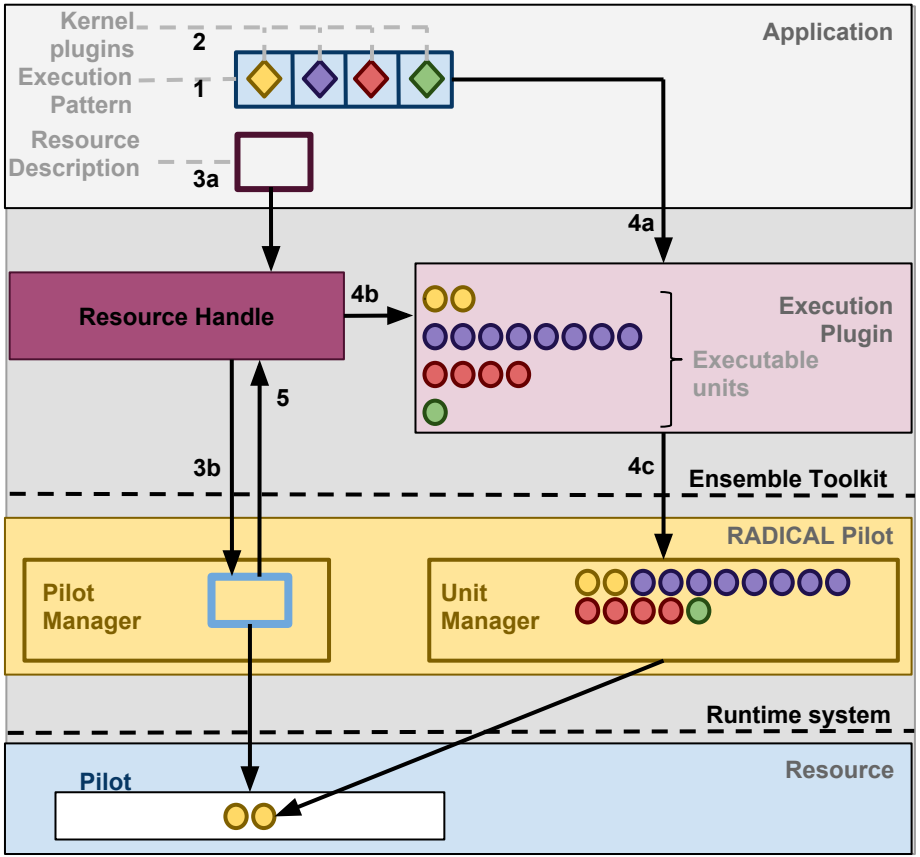
Ensemble Toolkit: Role of the components

Component	Role (answers the question..)	Responsibility
Kernel plugins	What to execute ?	User
Execution pattern	When to execute (the kernel) ?	User
Resource handle	Where to execute ?	User
Execution plugin	How to (best) execute ?	EnTK internal

Application

Ensemble Toolkit: User Perspective

1. Select required **execution pattern**
2. Add **kernel plugins**
3. Create **resource handle** targeting HPC and **allocate** resources.
4. **Run** the application
5. **Deallocate** resources



Ensemble Toolkit: UI Sample

```
1   class MyApp(PoE):
2
3           def __init__(self, stages,instances):
4                   super(MyApp,self).__init__(self, stages,instances)
5
6           def stage_1(self, instance):
7                   k = Kernel(name="misc.hello")
8                   k.arguments = ["--file=output.txt"]
9                   return k
10
11
12  app = MyApp(stages=1,instances=4)
13
14  cluster = SingleClusterEnvironment(
15                          resource="xsede.stmpade",
16                          cores=4,
17                          walltime=15,
18                          username='vivek',
19                          project='abc')
20
21  cluster.run(app)
```

Execution pattern

Kernel plugin

Resource handle

Execute workload on the resource

Ensemble Toolkit: Documentation

- **Quick start**: https://radical-cybertools.github.io/entk/quick_start.html

- User guide: http://radicalensemblemd.readthedocs.org/en/latest/
- Repository: https://github.com/radical-cybertools/radical.ensemblemd
- Technical paper: http://arxiv.org/abs/1602.00678