PLUGIN DEMO

Eduardo César

Anna Sikora, Antonio Pimenta

Universitat Autònoma de Barcelona

Overview

AUTOTUNE

www.autotune-project.eu

- Autotune goals and partners
- Periscope Tuning Framework (PTF)
- MPI Parameters Plugin
- Demo

Goals

AUTOTUNE

www.autotune-project.eu

- Tackle complexity of HPC architectures
  - Multicore, multisocket, accelerators, DVFS
- Enable higher productivity via auto-tuning
- Focus on static tuning in pre-production phase
  - Produce tuning recommendations
- Leverage state of the art performance analysis
- Implement an extensible environment
  - Support open and proprietary plugins

Partners

AUTOTUNE

www.autotune-project.eu

- UNIVIE
  - High level Parallel Patterns for GPGPU Plugin
- CAPS
  - Hybrid Manycore Tuning: HMPP Codelet Tuning Plugin
- LRZ
  - Energy Consumption via CPU Frequency Tuning Plugin
- UAB
  - Master-Worker MPI Plugin
  - MPI Runtime Plugin
- TUM
  - Compiler Flag Selection Plugin
  - User-level Tuning Plugin
- ICHEC

Overview

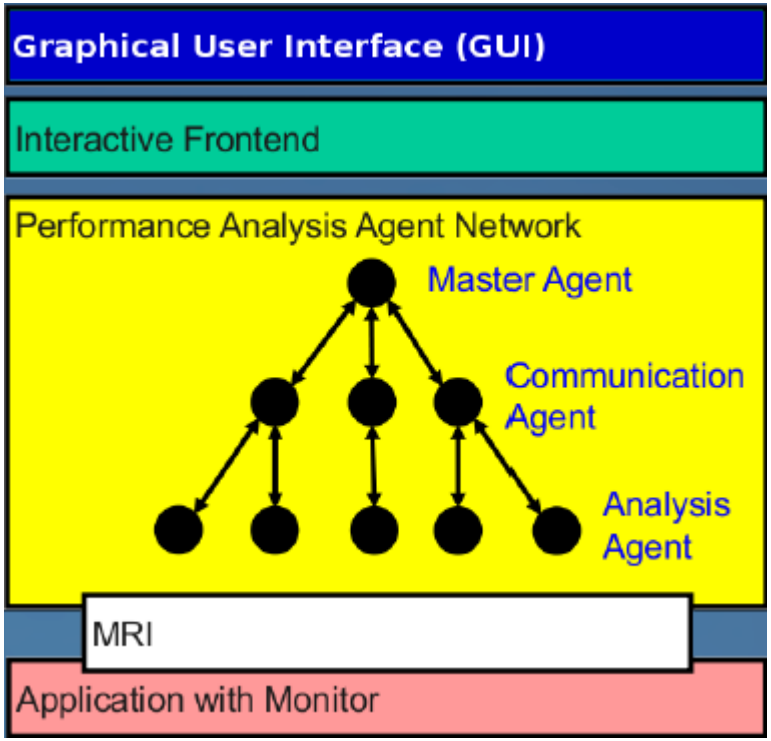
AUTOTUNE

www.autotune-project.eu

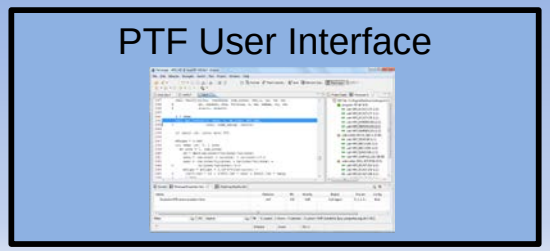
- Autotune partners and goals
- Periscope Tuning Framework (PTF)
- MPI Parameters Plugin
- Demo

Periscope

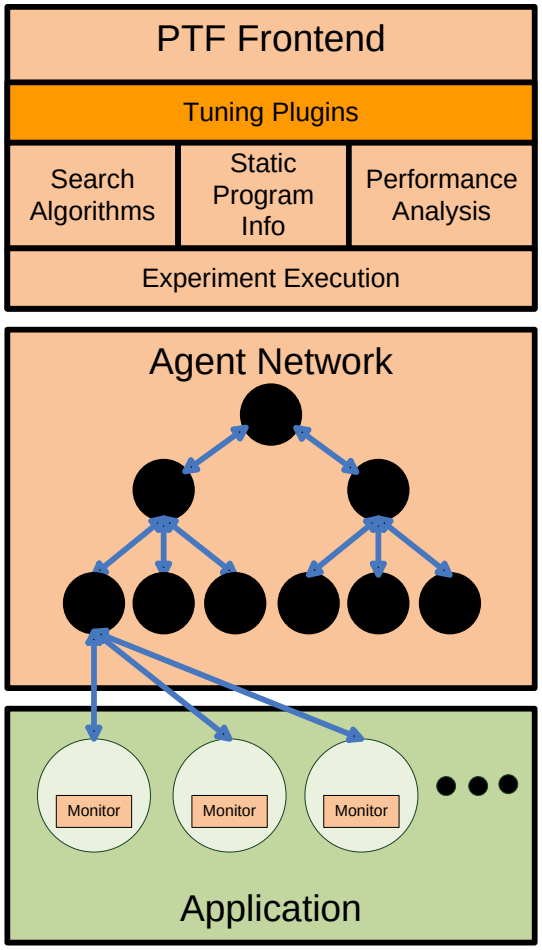
- On-line
  --no need to store trace files
- Distributed
  –reduced network overhead
  –based on autonomous cooperating agents
- Analyzes:
  - MPI Communication
  - Single-node Performance
  - OpenMP Performance
- Supports: Fortran, C/C++



Periscope Tuning Framework PTF



- Extension of Periscope
- Online tuning process
  - Application phase-based
- Extensible via tuning plugins
  - Single tuning aspect
  - Combining multiple tuning aspects
- Rich framework for plugin implementation
- Automatic and parallel experiment execution



Terminology

- Tuning Parameter
  - Name + integer range
- Tuning space TS_P of plugin P
  - $TS_P = TP_1 \times TP_2 \times \ldots \times TP_k$
- Variant space VS_r of region r
  - subset of tuning space
  - variant $v = (v_1, \ldots, v_k)$
- Objective
  - obj: $REG_{appl} \times TS_P \rightarrow \Re$
- Search Space
  - $((r_1, \ldots, r_n), VS, \{obj_1, \ldots, obj_n\})$

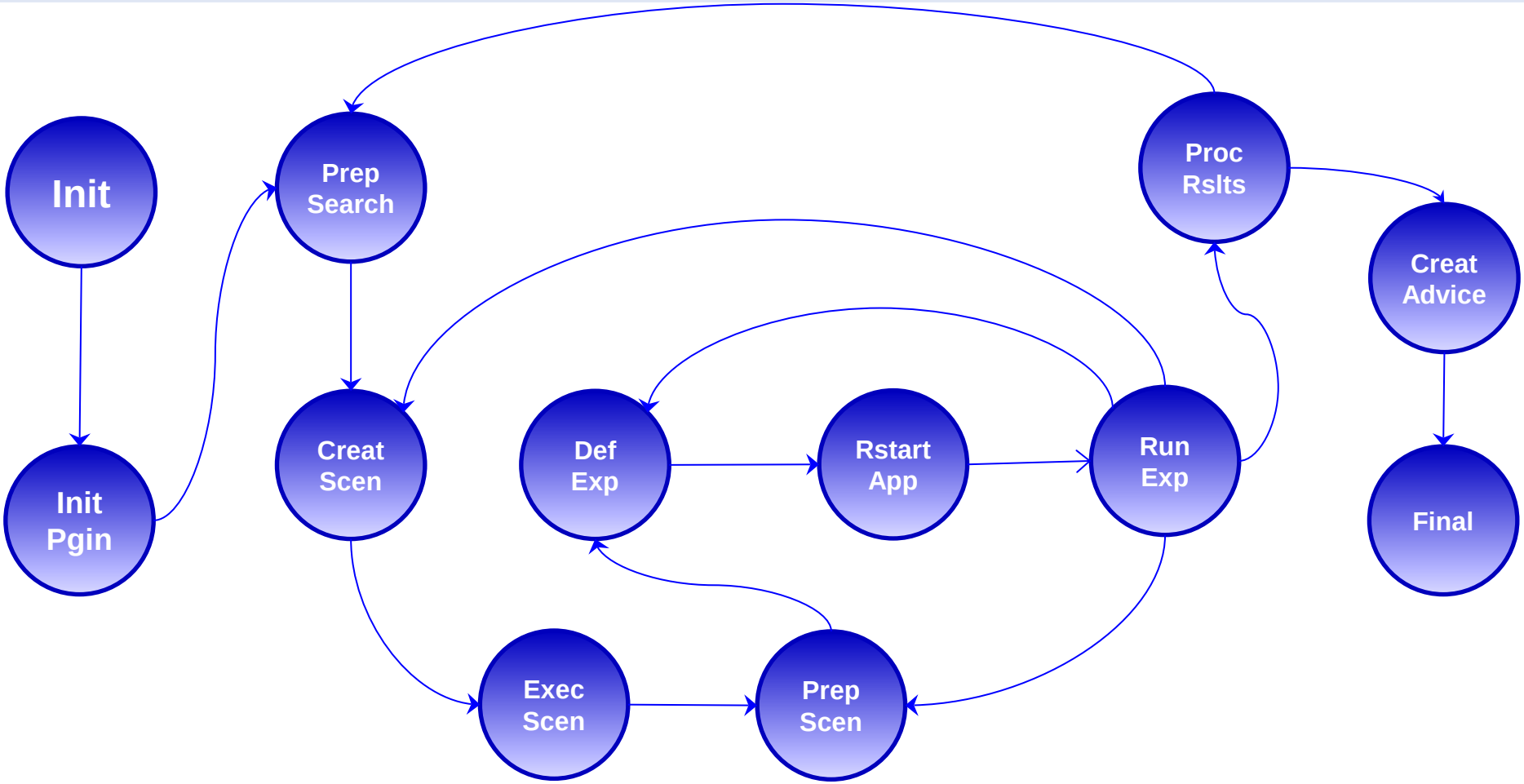
Terminology

- Tuning scenario sc of region
  - $sc_r=((r_1, \ldots, r_n), (v_1, \ldots, v_k) , \{obj_1, \ldots, obj_n\})$
- Tuning action TA_i
  - Enforces v_i for the tuning point *i* in variant υ
  - Runtime tuning actions: variable assignment and function call
  - Pre-runtime tuning actions: Compilation, #MPI tasks, MPI parameters
- Search algorithm
  - $\forall \upsilon \in$ VS create $((r_1, \ldots, r_n), v, \{obj_1, \ldots, obj_n\})$
  - Multiple search spaces: Scenarios are the xproduct of all scenarios of each search space.

PTF's State Machine

AUTOTUNE

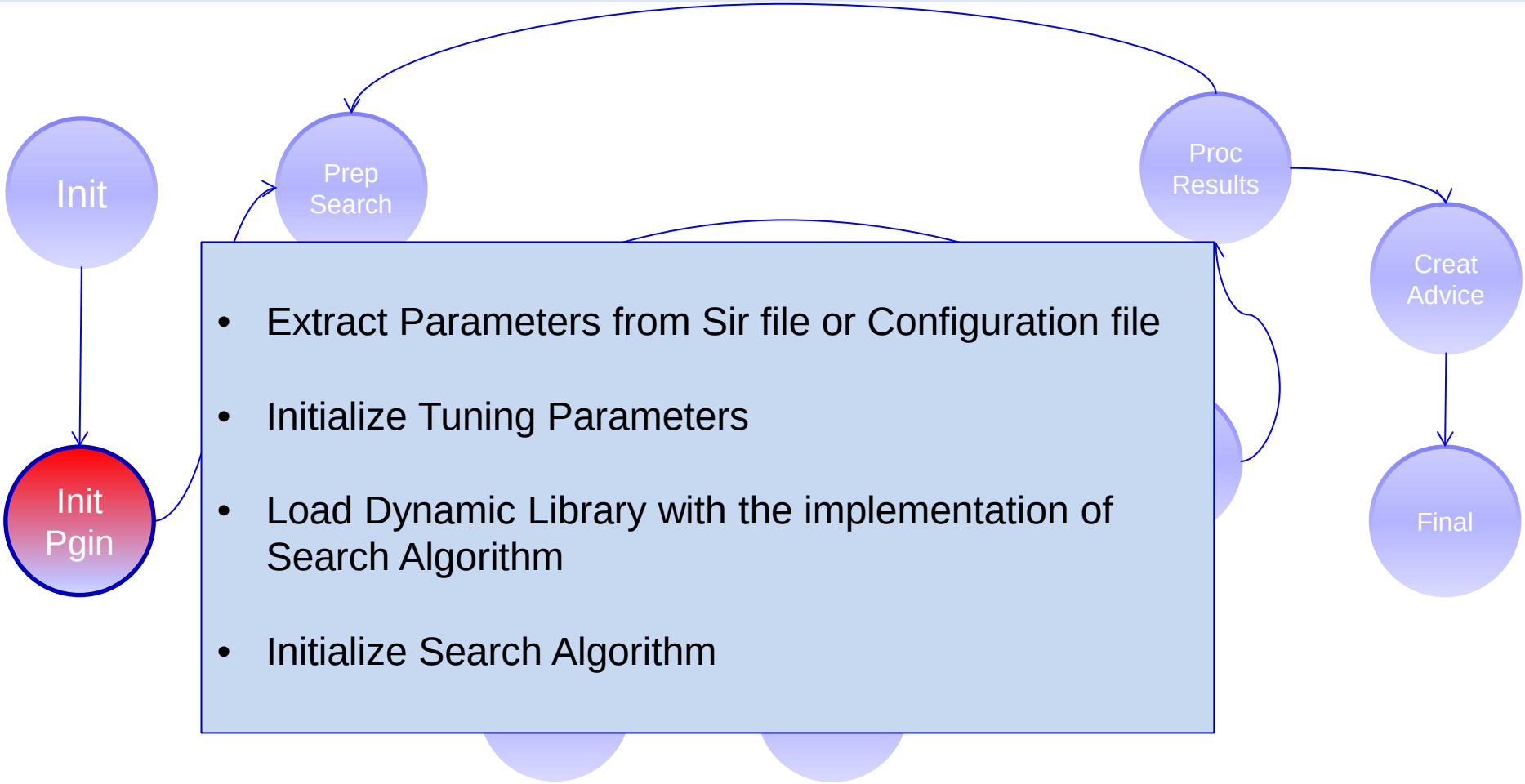
www.autotune-project.eu



Tuning Plugins

AUTOTUNE

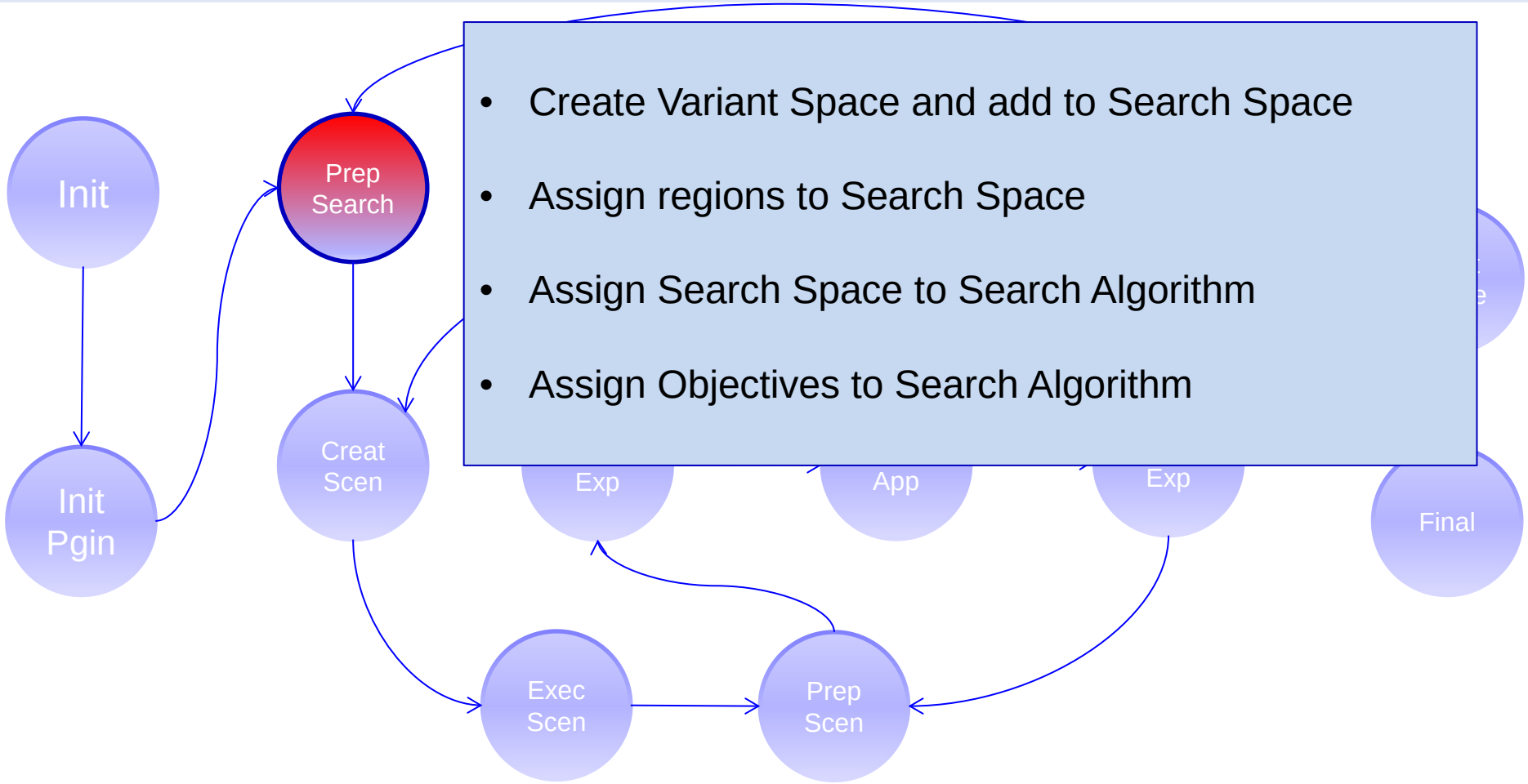
www.autotune-project.eu



Tuning Plugins

AUTOTUNE

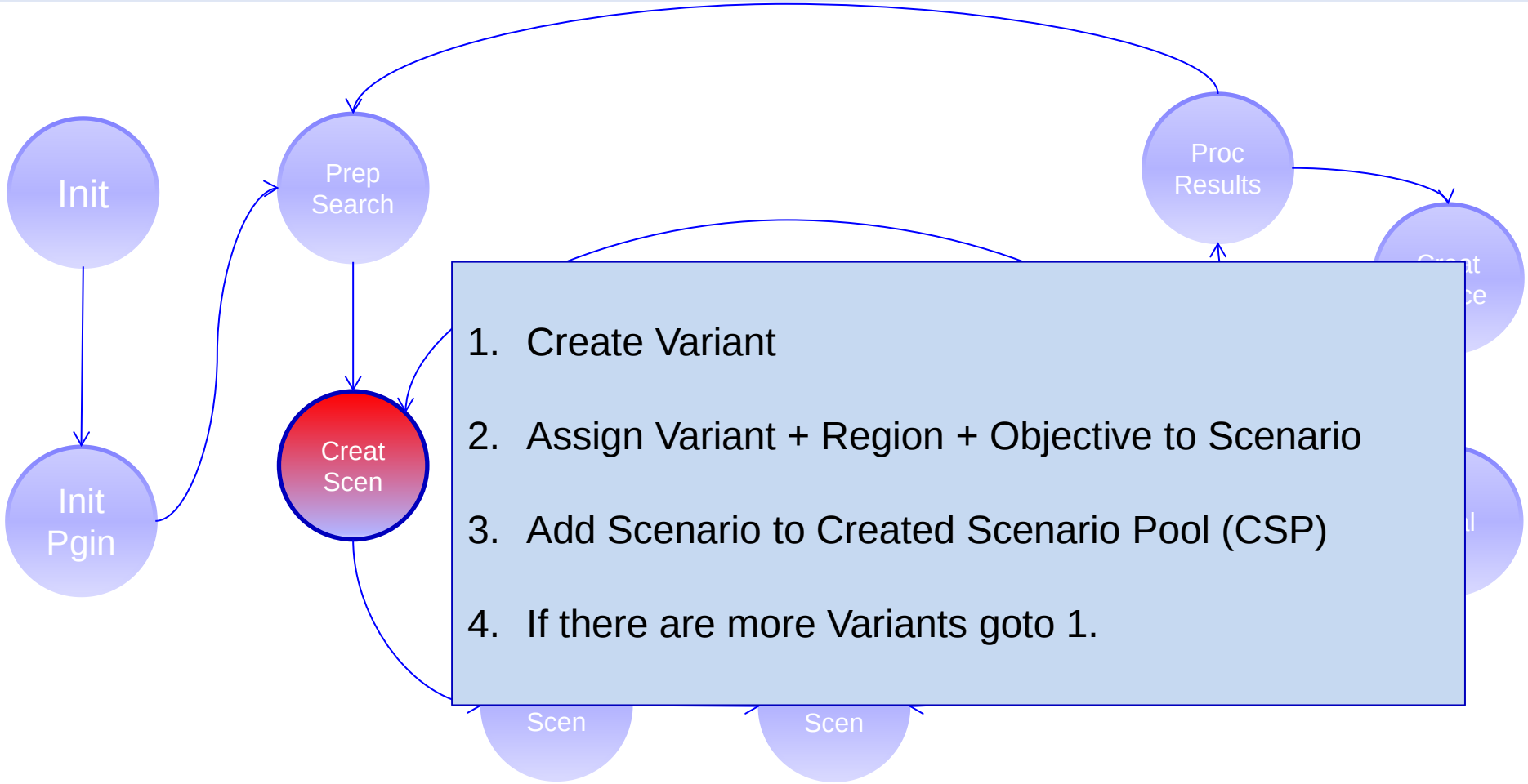
www.autotune-project.eu



Tuning Plugins

AUTOTUNE

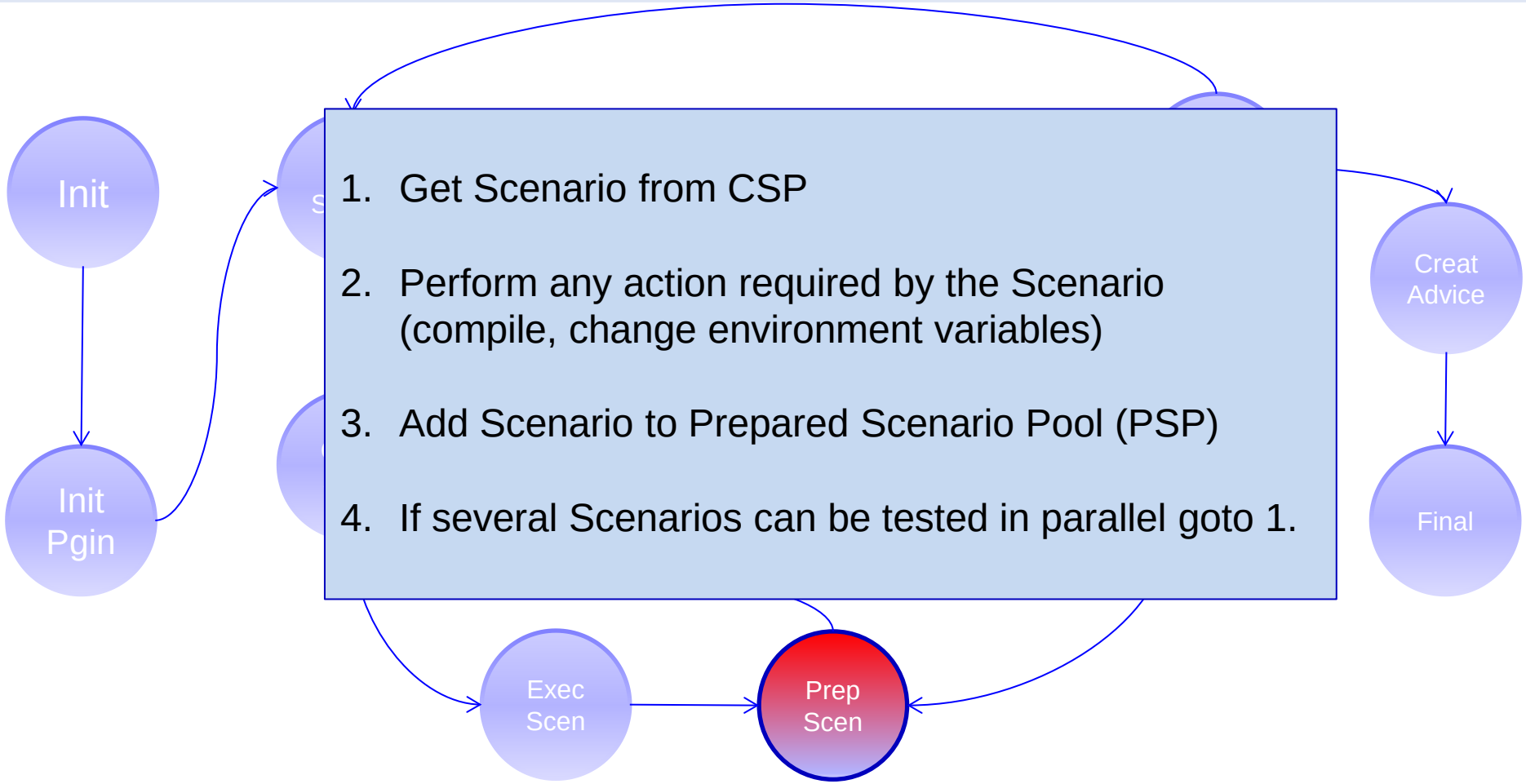
www.autotune-project.eu



Tuning Plugin

AUTOTUNE

www.autotune-project.eu

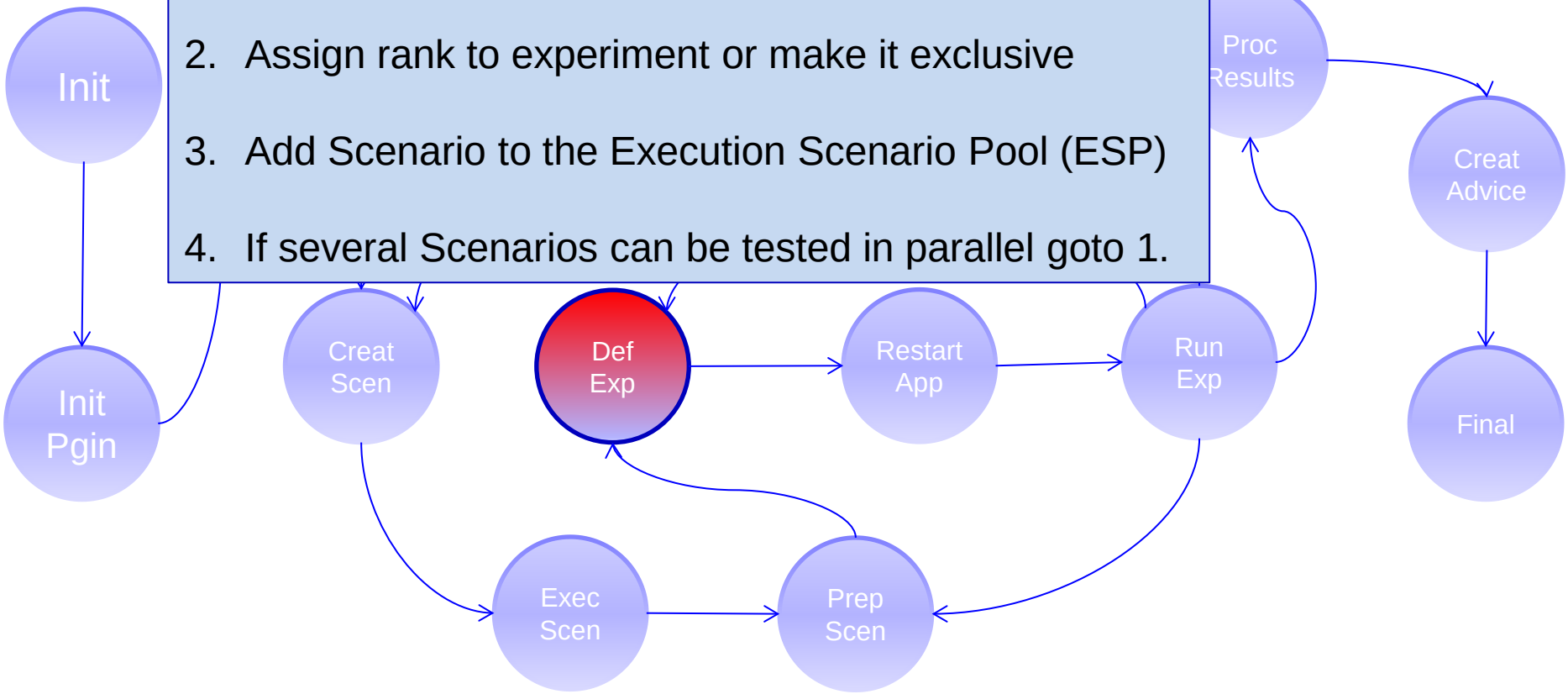


Tuning Plugin

AUTOTUNE

www.autotune-project.eu

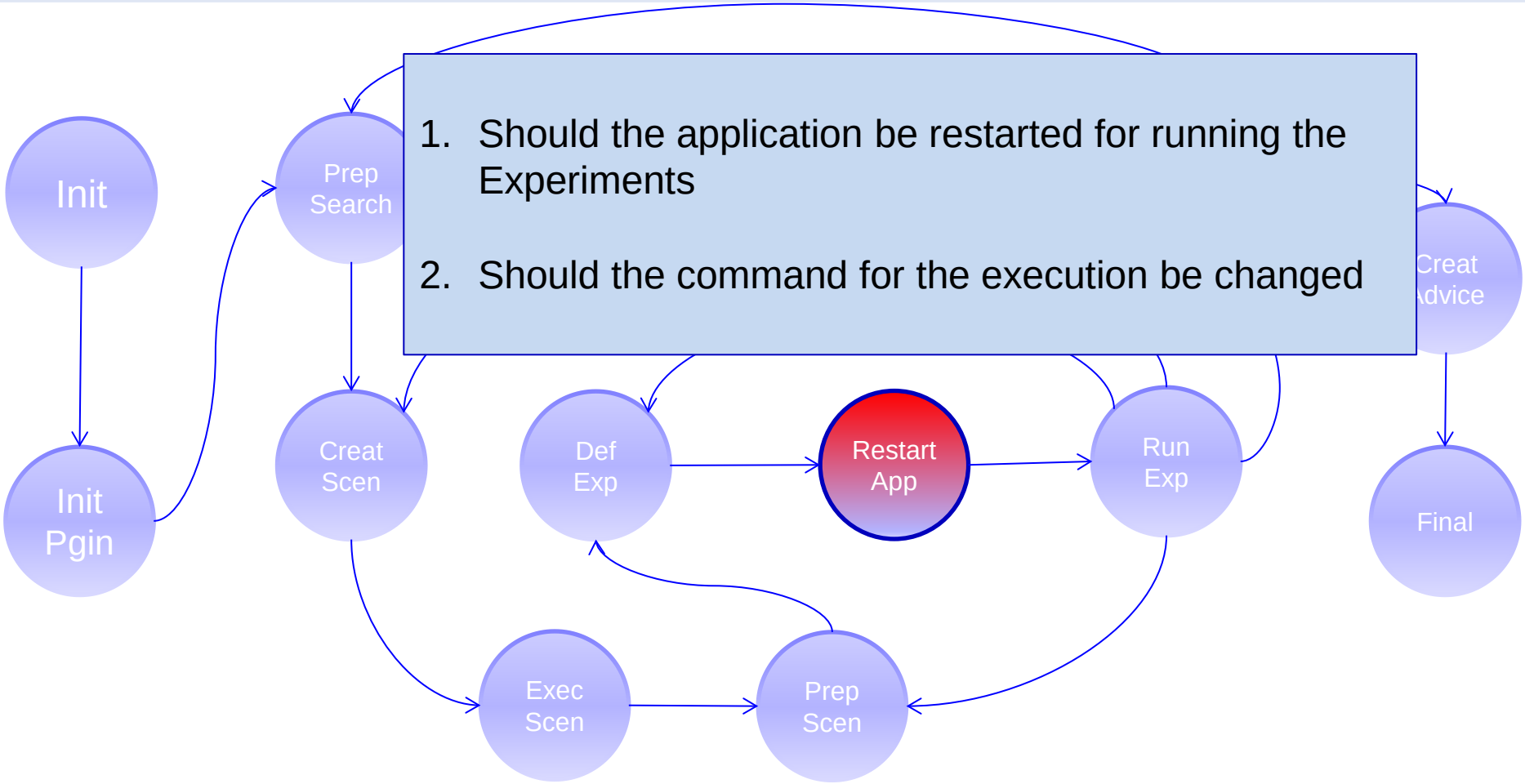
1. Get Scenario from PSP
2. Assign rank to experiment or make it exclusive
3. Add Scenario to the Execution Scenario Pool (ESP)
4. If several Scenarios can be tested in parallel goto 1.



Tuning Plugin

AUTOTUNE

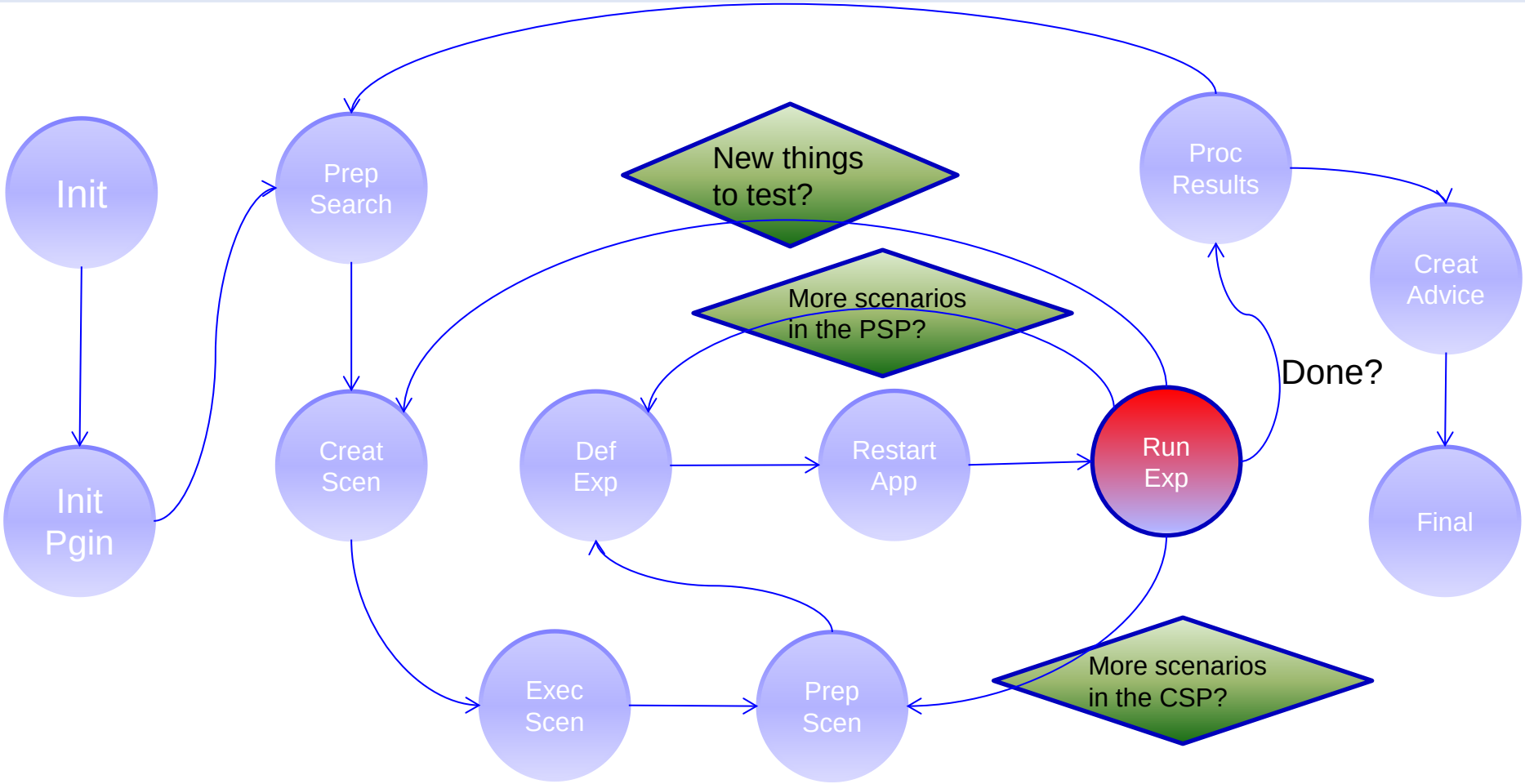
www.autotune-project.eu



Tuning Plugin

AUTOTUNE

www.autotune-project.eu



Overview

AUTOTUNE

www.autotune-project.eu

- Autotune partners and goals
- Periscope Tuning Framework (PTF)
- MPI Parameters Plugin
- Demo

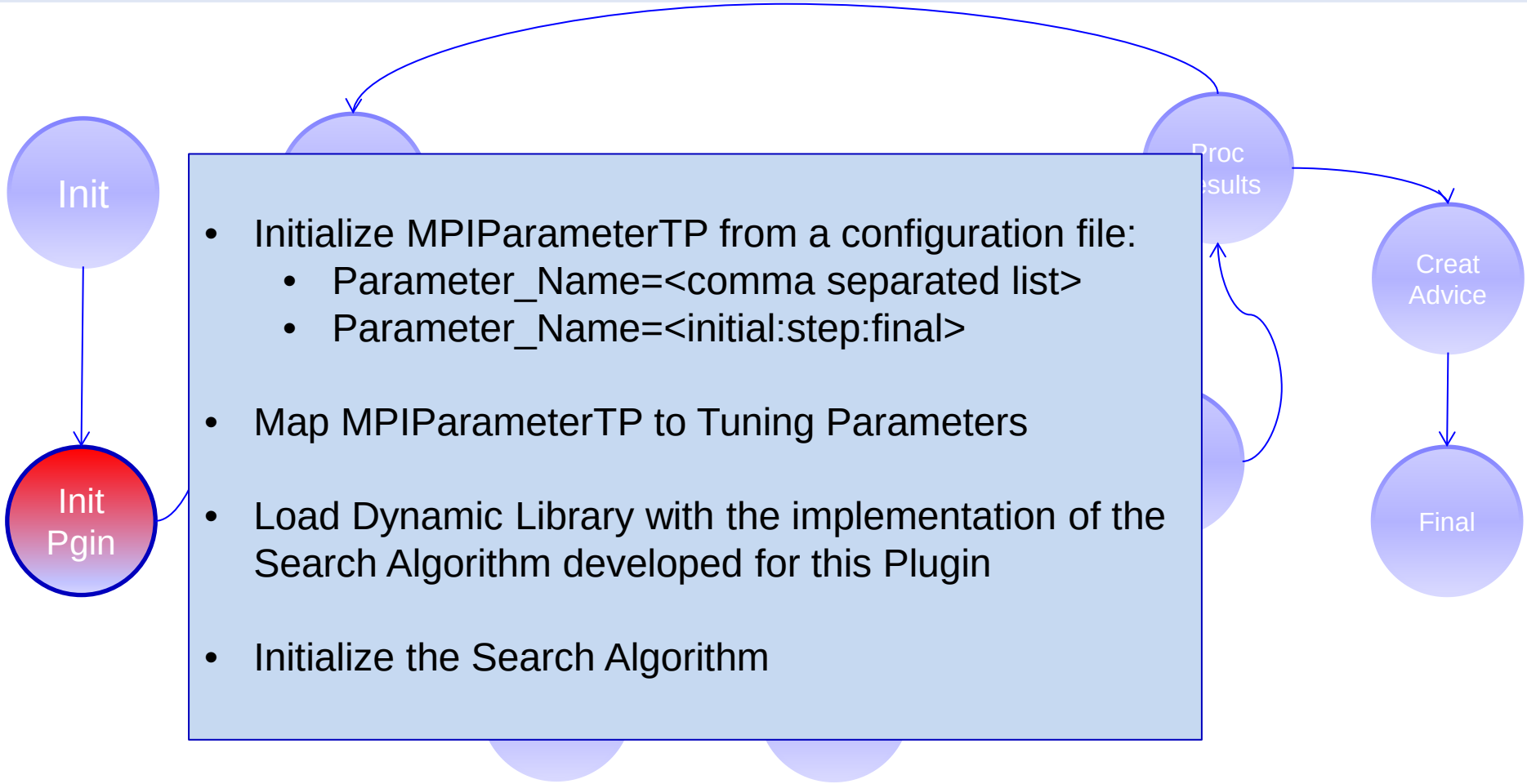
MPI Parameters Plugin:
Introduction

AUTOTUNE
www.autotune-project.eu

- Goal
  - Reduce time of user region based on finding the right combination of user provided MPI parameters.
- Tuning Parameters
  - Each parameter has a list of values to test that can be indicated explicitly or with a range
- Tuning Action
  - The tuning action is to set a variant of the parameters in the preparation phase of a scenario before reexecuting.

MPI Parameters Plugin: Control Flow

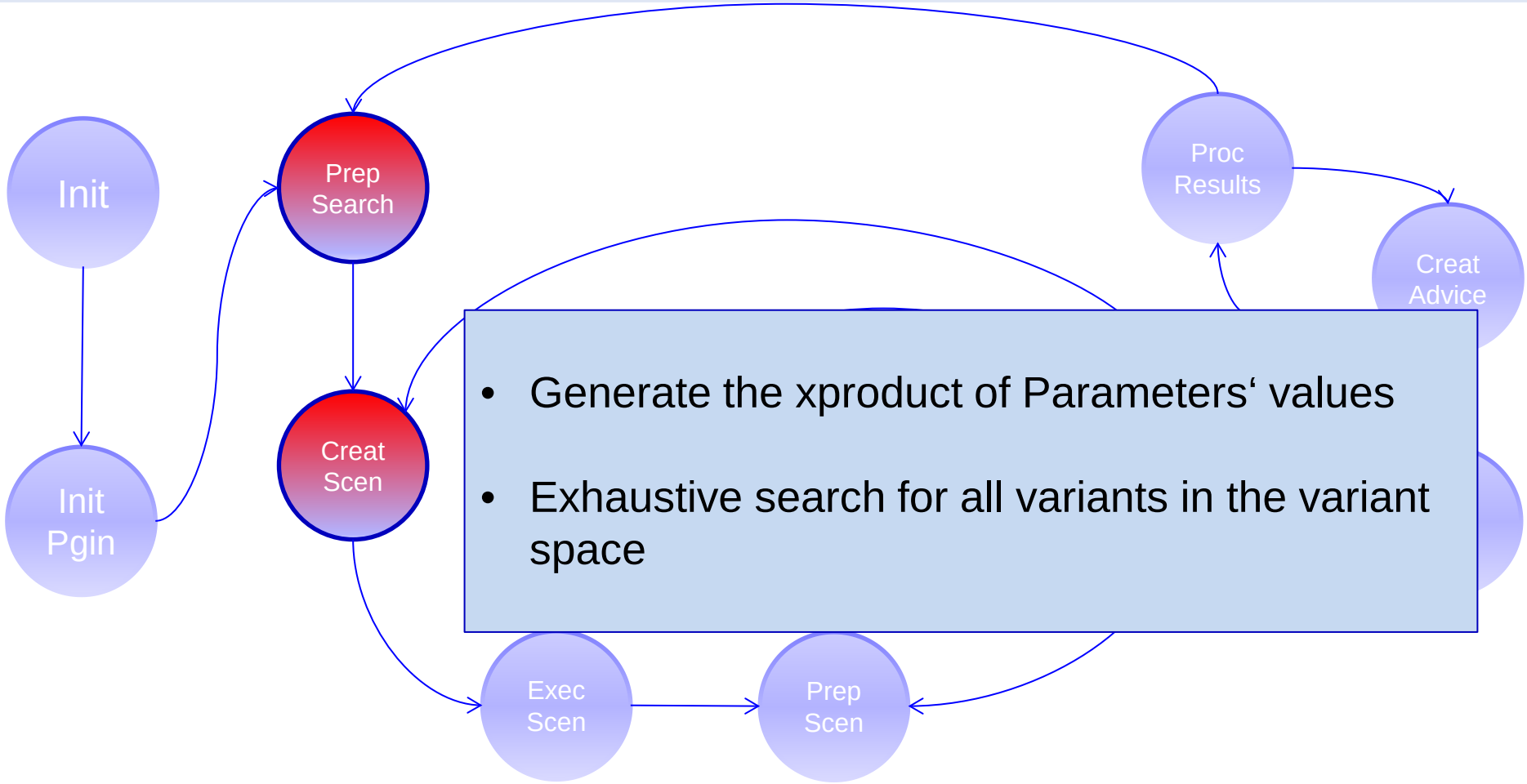
AUTOTUNE
www.autotune-project.eu



- Initialize MPIParameterTP from a configuration file:
  - Parameter_Name=<comma separated list>
  - Parameter_Name=<initial:step:final>
- Map MPIParameterTP to Tuning Parameters
- Load Dynamic Library with the implementation of the Search Algorithm developed for this Plugin
- Initialize the Search Algorithm

MPI Parameters Plugin:
Control Flow

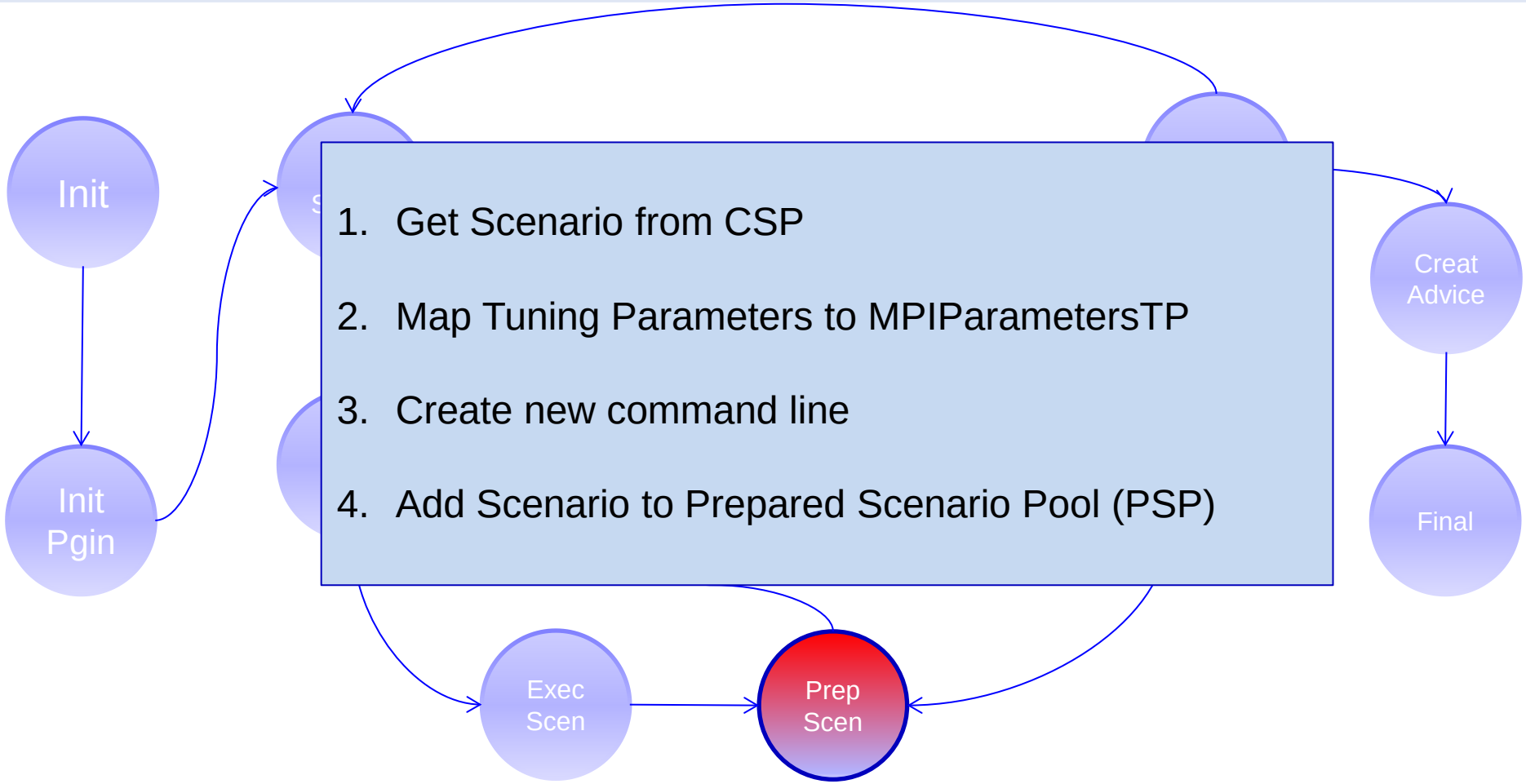
AUTOTUNE
www.autotune-project.eu



- Generate the xproduct of Parameters‘ values
- Exhaustive search for all variants in the variant space

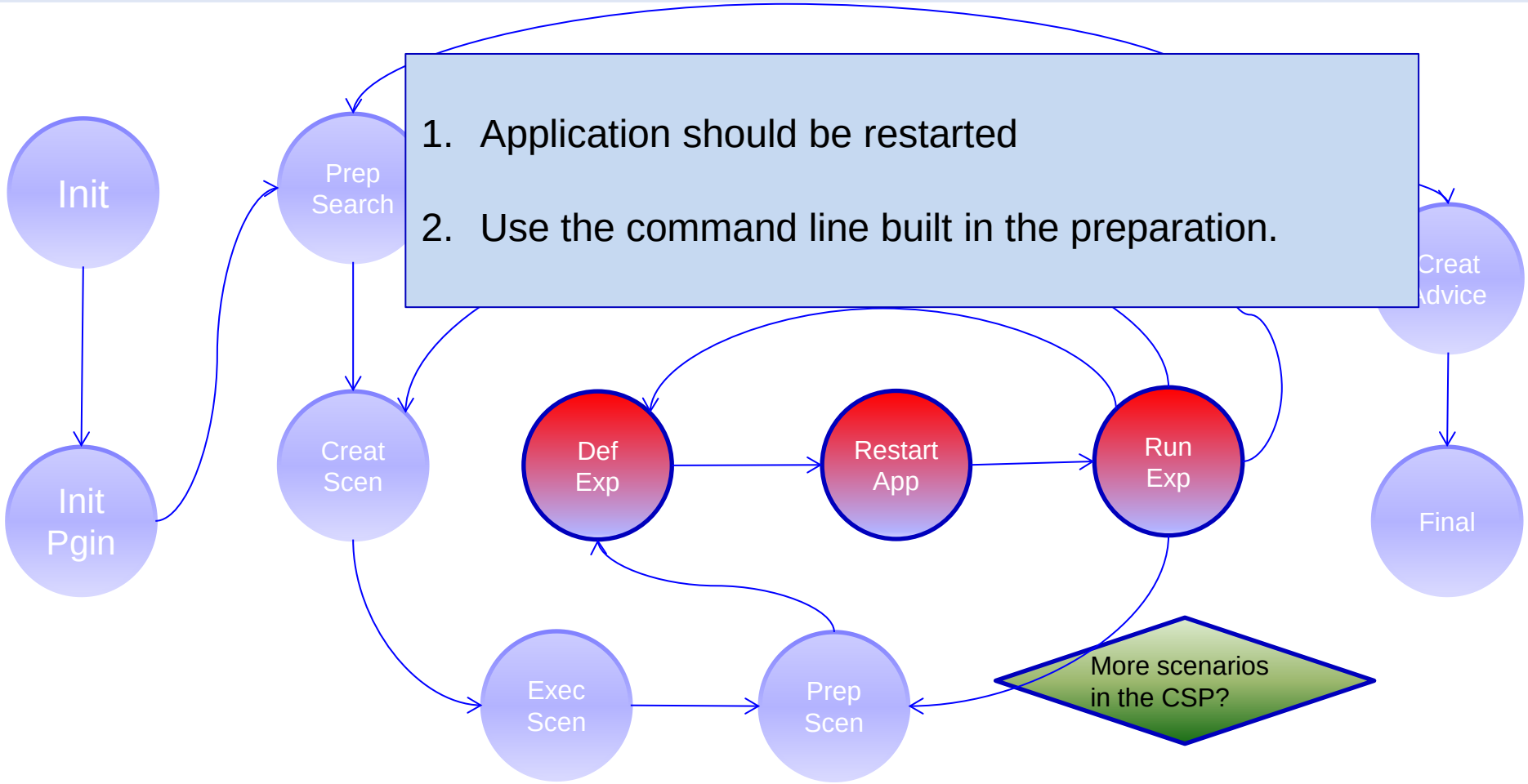
MPI Parameters Plugin:
Control Flow

AUTOTUNE
www.autotune-project.eu



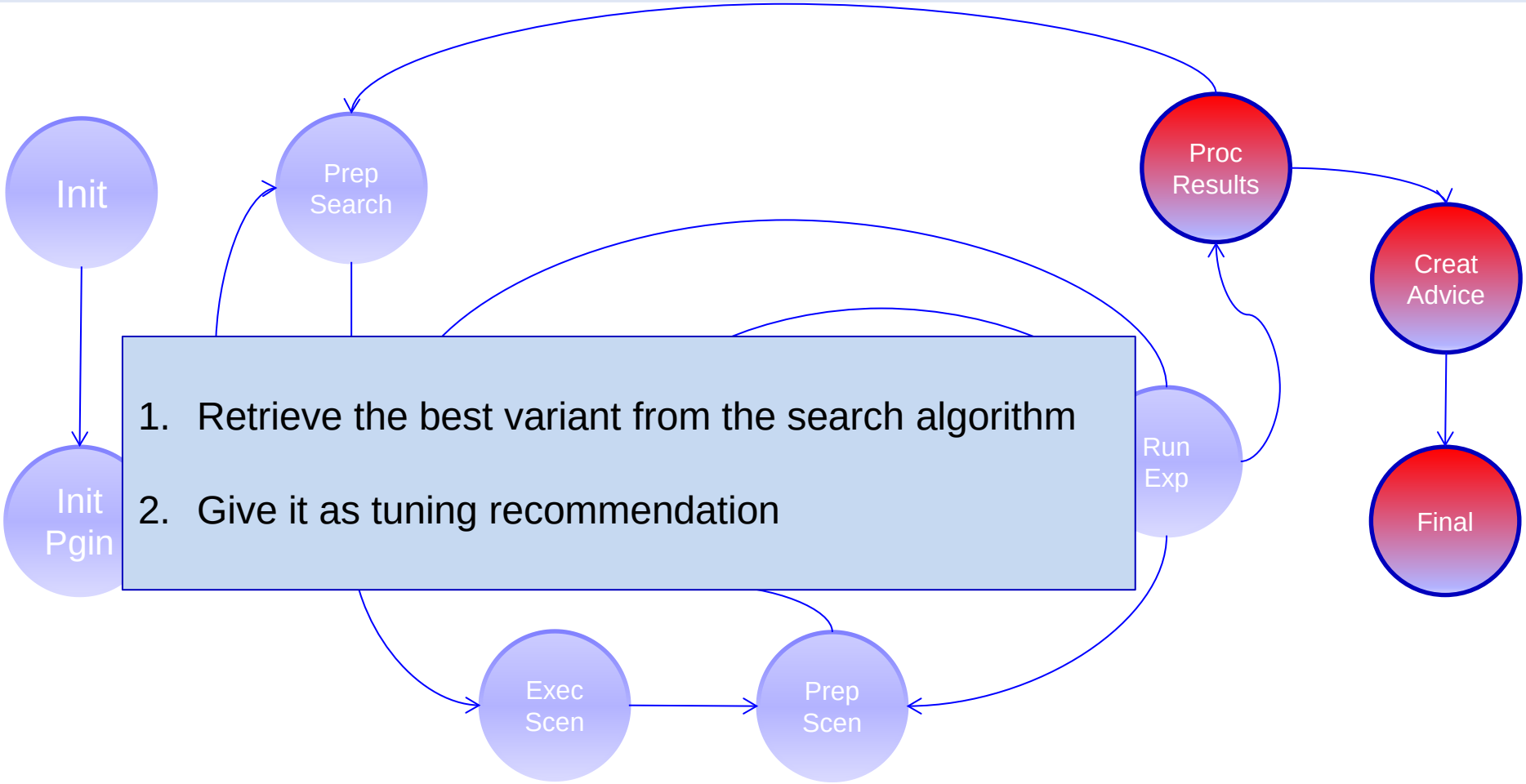
MPI Parameters Plugin:
Control Flow

AUTOTUNE
www.autotune-project.eu



MPI Parameters Plugin: Control Flow

AUTOTUNE
www.autotune-project.eu



Overview

AUTOTUNE

www.autotune-project.eu

- Autotune partners and goals
- Periscope Tuning Framework (PTF)
- MPI Parameters Plugin
- Demo

Tuning Model

- Tuning Parameter
  - Type: variable
  - Name: TEST
  - Range: 1..10
- Tuning space: TS = 1..10
- Variant space = TS
- Tuning scenario:
  - ((1, 485), (1), {ExecTime}) to ((1, 485), (10), {ExecTime})
- Tuning action
  - Assign 1 to variable `variant` in `start_region(1,485)`

Region Code


```
do k=1,20
  variant=k
  !$MON USERREGION TP name(Test) variable(variant) variants(10)
  tstart=MPI_Wtime()
     mm=5-variant+1
     if (mm<=0)
        m= -1*mm+1
     endif
     call sleep(mm)
  tend=MPI_Wtime()
  write (*,*) myrank, variant, tend-tstart
  !$MON END USERREGION
enddo
```