Resumen semana 12-18

lunes, 19 de marzo de 2018 18:20

Método principal: phinit::run_ph()

Uso del **objeto ph**.

Creado con el contexto:

```
# This context manages releasing pyro workers and
# closing file archives
with PHFromScratchManagedContext(options) as ph:
    run_ph(options, ph)
```

GenerateScenarioTreeForPH

PHAlgorithmBuilder

(l. 1022) Método solve()

(l. 3997) ph::solve

dual_mode actualiza los *xbar*

Check para ejecución en *async*. Se supone que se ejecuta como async cuando está funcionando con un phsolverserver.

(l. 3713) ph::async_iteration_k_plus_solves

/examples/pysp/scripting/solve_distributed.py

mpi

- pyomo_ns --> pyro.start_ns()
- dispatch_srvr
- phsolverserver --> el único que se ejecuta en paralelo
- runph --> usando *solver-manager=phpyro*

phsolverserver

Cosa de Pyro

PHPyroWorker : TaskWorker

```
try:
    # spawn the daemon
    TaskWorkerServer(PHPyroWorker,
                     host=options.pyro_host,
                     port=options.pyro_port)
```

Llamada al método *run()*

No hay método run, hay un método *process* que parece que funciona en base a mensajes como string.

Para el Debug, generar un archivo para llamar a las funciones que serían los comandos