



Beyond Performance Analysis: Automatic Program Tuning

Prof. Dr. Michael Gerndt
Technische Universität München
gerndt@in.tum.de

Incremental Tuning is Essential

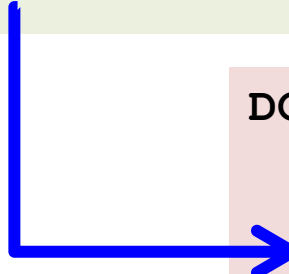


AutoTune FP7 Project Goals

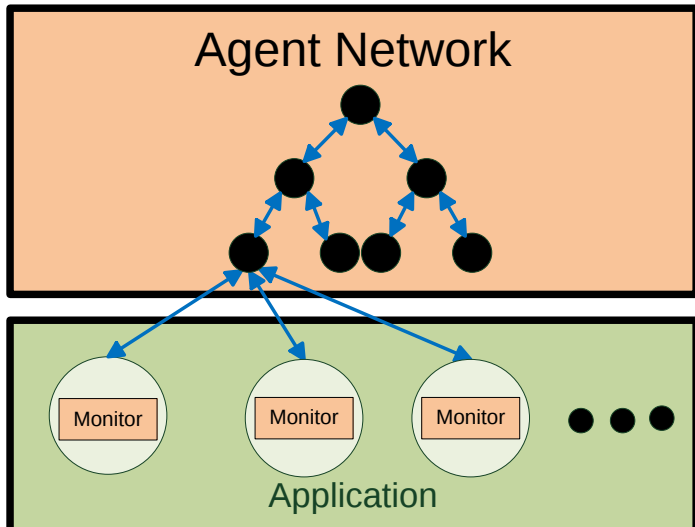
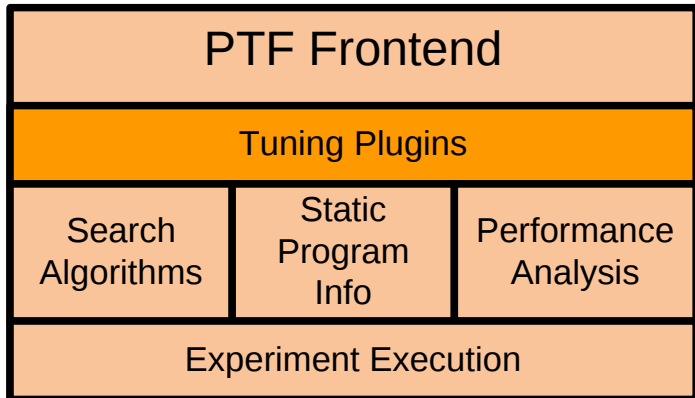
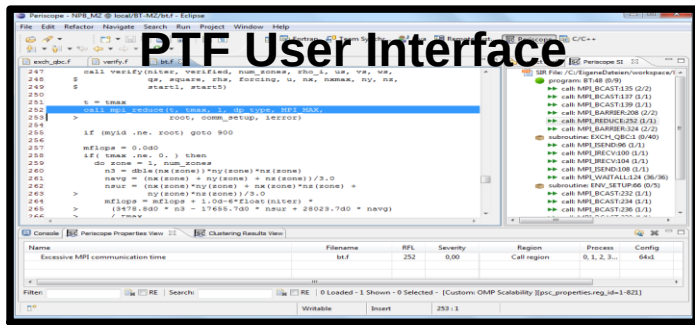
- Extend automatic performance analysis with automatic tuning
 - Many excellent performance analysis tools exist.
 - None is supporting the developer in the tuning step.
- Implement the Periscope Tuning Framework (PTF)
 - Extensible tuning framework based on **tuning plugins**
 - Rich framework supporting the implementation of tuning plugins.
 - Tuning plugins for diverse programming models and architectures

HMPP Tuning Plugin Demo

```
do k=1,20
  !$MON USERREGION TP name(HMPP) variable(v_sel) variants(3)
  !$hmp <convolution> filter5x5 &
  !$hmp& callsite variants(filter@<convolution>[CUDA], &
  !$hmp& filter_C1060@<convolution>[CUDA], &
  !$hmp& filter_C2050@<convolution>[CUDA]) &
  !$hmp& selector(v_sel)
  call filter(fullHeigh, width, stencil, rasterIn, rasterOut)
  !$MON END USERREGION
enddo
```



```
DO k = 1, 20
  CALL psc_map_tp_var('<HMPP>',v_sel)
  CALL start_region(27,1,485,0,-1)
  ...
  CALL end_region(27,1,485,0,-1)
END DO
```



```

Class DemoPlugin: Public Iplugin{ ...
void initialize(string sirFilePath);
void startTuningStep(void);

void createScenarios(void);
void prepareScenarios(void);
void defineExperiment(int numprocs);
    bool restartRequired(string *);
    bool searchFinished(void);
void finishTuningStep(void);
bool tuningFinished(void);
void getAdvice(void);
void finalize(void);
void terminate(void);
...}

```

Plugins under Development

- HMPP tuning
- MPI parameter tuning
- Parallel Pattern tuning
- Compiler Flag Selection tuning
- Energy DVFS tuning
- Energy Parallelism Throttling tuning

THANK YOU