

Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep  
Cube  
scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

# Gprof - Scalasca - VTune

Νίκος Νικολουτσάκος

ΕΔΕΤ

nikoloutsa [at] grnet.gr

[http://doc.aris.grnet.gr/development/  
#performance-analysis](http://doc.aris.grnet.gr/development/#performance-analysis)

## Profiling

N. Νικολου-  
τσάκος

### Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

- 1 Χρονοεπίδοση
  - Παράγοντες επίδοσης
  - Μετρικές Επίδοσης
  - Τεχνικές Μέτρησης
- 2 Time
- 3 Gprof
- 4 Scalasca
  - scorep
  - Cube
  - scalasca
- 5 VTUNE
- 6 Συμπεράσματα

## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

## Παράγοντες επίδοσης παράλληλων προγραμμάτων

### ● Σειριακοί παράγοντες

- Υπολογισμοί (αλγόριθμος, βελτιστοποίηση μεταγλωττιστή)
- Κρυφή & Κύρια μνήμη (αρχιτεκτονική, fine-tuning)
- I/O

### ● Παράλληλοι παράγοντες

- Τμηματοποίηση, Διαμοιρασμός
- Επικοινωνία (message passing)
- Multithreading
- Συγχρονισμός / Αποκλεισμός

## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

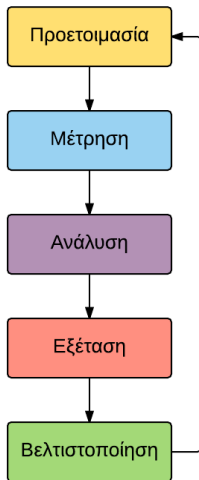
Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία



- Χρονομέτρηση καλύτερη από την εικασία
- Εύρεση καθυστερήσεων
- Σύγκριση εναλλακτικών
- Επικύρωση επιλογών βελτιστοποίησης

## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

## Τι μετράμε;

- Χρόνος εκτέλεσης
  - Wall-clock (υπολογισμός, I/O, σύστημα)
  - CPU time
  - **μη-ντετερμινιστικός** μέσο όρο μετρήσεων...
- Αριθμό κλήσεων συνάρτησης
- CPU cycles per instruction
- FLOPs
- Πλήθος μηνυμάτων , μέγεθος ...

## Profiling

N. Νικολου-τσάκος

## Χρονοεπίδοση

Παράγοντες επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

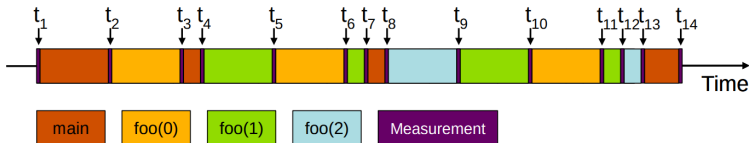
Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία



Πώς ενεργοποιούνται οι μετρήσεις επιδόσεων;

Δειγματοληψία

Code instrumentation

Πώς καταγράφονται τα δεδομένα?

Profiling / Runtime summarization

Tracing

- Profiling: Summarization of events over execution interval
- Tracing: Chronologically ordered sequence of event records

## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

## Τύποι profiling

- Flat profile
  - κατανομή μετρήσεων ανα συνάρτηση
- Call-path profile
  - κατανομή μετρήσεων ανα μονοπάτι εκτέλεσης
- Ειδικού σκοπού
  - εστίαση σε MPI calls , OpenMP constructs, ...

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία

*Fortran 90*

```
integer :: counti, countf, count_rate  
real    :: dt  
call system_clock(counti,count_rate)  
... work ....  
call system_clock(countf)  
dt=REAL(countf-counti)/REAL(count_rate)
```



## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

C

```
static double get_second (void){  
    struct timeval tv;  
    gettimeofday(&tv, NULL);  
    return (double)tv.tv_sec + (double)tv.tv_usec /  
        1000000.0; } //1.e6
```

```
start = get_second();  
...work...  
stop = get_second();  
double totaltime = stop - start;
```

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία

*MPI*

```
double precision :: t1,  
                  t2,dt  
...  
t1 = MPI_WTIME()  
... work ...  
t2 = MPI_WTIME()  
deltat = t2-t1
```

```
double t1, t2;  
t1 = MPI_WTIME();  
...work...  
t2 = MPI_WTIME();  
deltat = t2-t1;
```

## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

“Profiling allows you to learn where your program spent its time and which functions called which other functions while it was executing”

- gprof GNU Profiler
- [sourceware.org/binutils/docs/gprof/](http://sourceware.org/binutils/docs/gprof/)
- only user space

## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

## Προετοιμασία Μεταγλώττιση + Σύνδεση

```
gcc [flags] -g <source file> -o <output  
file> -pg
```

## Error

gprof: gmon.out file is missing call-graph data

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία

```
gprof options [executable-file  
[profile-data-files...]] [> outfile]
```

## ● Επιλογές εξόδου

- -b doesn't print the verbose blurbs
- -p print a flat profile
- -p<func name> function specific
- -P suppress flat profile
- -q, -Q call graph only

## ● Επιλογές ανάλυσης

- -a : suppress the printing of statically declared (private) functions.

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία

```
gcc -Wall -pg test_gprof.c -o test_gprof
```

```
gprof test_gprof gmon.out > analysis.txt
```

```
%           the percentage of the total running time of the
time        program used by this function.

cumulative  a running sum of the number of seconds accounted
seconds     for by this function and those listed above it.

self        the number of seconds accounted for by this
seconds     function alone. This is the major sort for this
            listing.

calls       the number of times this function was invoked, if
            this function is profiled, else blank.

self        the average number of milliseconds spent in this
ms/call     function per call, if this function is profiled,
            else blank.
```

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

## Gprof

## Scalasca

scorep

Cube

scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία

Flat profile:

Each sample counts as 0.01 seconds.

| %     | cumulative | self    | calls | self   | total  |  | name      |
|-------|------------|---------|-------|--------|--------|--|-----------|
| time  | seconds    | seconds |       | s/call | s/call |  |           |
| 33.74 | 10.52      | 10.52   | 1     | 10.52  | 20.96  |  | func1     |
| 33.51 | 20.97      | 10.45   | 1     | 10.45  | 10.45  |  | func2     |
| 33.48 | 31.42      | 10.44   | 1     | 10.44  | 10.44  |  | new_func1 |
| 0.16  | 31.47      | 0.05    |       |        |        |  | main      |

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 0.03% of 31.47 seconds

| index | % time | self  | children | called | name          |
|-------|--------|-------|----------|--------|---------------|
|       |        |       |          |        | <spontaneous> |
| [1]   | 100.0  | 0.05  | 31.42    |        | main [1]      |
|       |        | 10.52 | 10.44    | 1/1    | func1 [2]     |
|       |        | 10.45 | 0.00     | 1/1    | func2 [3]     |
| ----- |        |       |          |        |               |
|       |        | 10.52 | 10.44    | 1/1    | main [1]      |
| [2]   | 66.6   | 10.52 | 10.44    | 1      | func1 [2]     |

## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

```
gprof -Pfunc1 -b test_gprof gmon.out
```

Flat profile:

Each sample counts as 0.01 seconds.

| %      | cumulative | self    | self  | total  |        |       |
|--------|------------|---------|-------|--------|--------|-------|
| time   | seconds    | seconds | calls | s/call | s/call | name  |
| 102.69 | 10.52      | 10.52   | 1     | 10.52  | 10.52  | func1 |



## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία

- “Scalable performance analysis of large-scale parallel applications”
- <http://www.scalasca.org>
- Open source
- **ScoreP**: instrumentation, mpi/shmem, openmp/threads hybrid
- **Cube**: Analysis report
- **Scalasca**: trace analysis

module load scalasca

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep

Cube

scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία

## scalasca

Toolset for scalable performance analysis of  
large-scale parallel applications

usage: scalasca [OPTION]... ACTION <argument>...

1. **prepare application** objects and executable  
for measurement:

```
scalasca -instrument <compile-or-link-command> #  
skin (using scorep)
```

2. **run application** under control of measurement  
system:

```
scalasca -analyze <application-launch-command> #  
scan
```

3. interactively explore measurement **analysis**  
**report**:

```
scalasca -examine <experiment-archive|report> #  
square
```

## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

## Instrumentation **scorep**

```
MPIF77 = scorep mpif77
```

## Check scorep env

```
scorep-info config-vars -full
```

## Run **scan**

```
export  
SCOREP_EXPERIMENT_DIRECTORY=scorep_bt-  
mz_W_4x4_sum  
scan mpirun -np 4 ./bt-mz_W.4
```

## Profiling

N. Νικολού-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

## scorep

## Cube

## scalasca

## VTUNE

## Συμπεράσματα

## Βιβλιογραφία

## Analysis report **square**

```
square -s scorep_bt-mz_W_4x4_sum
scorep-score scorep_bt-mz_W_4x4_sum/profile.cubex
scorep-score -r scorep_bt-mz_W_4x4_sum/profile.cubex
```

| type | max_buf[B]  | visits     | time[s] | time[%] | time/visit[us] | region          |
|------|-------------|------------|---------|---------|----------------|-----------------|
| ALL  | 277,805,118 | 41,158,333 | 37.01   | 100.0   | 0.90           | ALL             |
| USR  | 274,792,492 | 40,418,321 | 14.82   | 40.0    | 0.37           | USR             |
| OMP  | 6,882,860   | 685,952    | 21.71   | 58.7    | 31.65          | OMP             |
| COM  | 377,156     | 46,744     | 0.19    | 0.5     | 4.07           | COM             |
| MPI  | 102,286     | 7,316      | 0.29    | 0.8     | 39.68          | MPI             |
| USR  | 85,774,338  | 12,516,672 | 5.32    | 14.4    | 0.43           | binvrhs_        |
| USR  | 85,774,338  | 12,516,672 | 3.84    | 10.4    | 0.31           | matvec_sub_     |
| USR  | 85,774,338  | 12,516,672 | 4.57    | 12.4    | 0.37           | matmul_sub_     |
| USR  | 7,974,876   | 1,170,624  | 0.43    | 1.2     | 0.36           | binvrhs_        |
| USR  | 7,974,876   | 1,170,624  | 0.46    | 1.2     | 0.39           | lhsinit_        |
| USR  | 3,473,912   | 526,848    | 0.20    | 0.5     | 0.39           | exact_solution  |
| OMP  | 410,040     | 25,728     | 0.01    | 0.0     | 0.52           | !\$omp parallel |
| OMP  | 410,040     | 25,728     | 0.01    | 0.0     | 0.52           | !\$omp parallel |
| OMP  | 410,040     | 25,728     | 0.01    | 0.0     | 0.53           | !\$omp parallel |
| OMP  | 410,040     | 25,728     | 0.01    | 0.0     | 0.54           | !\$omp parallel |

## Profiling

N. Νικολου-τσάκος

cube scorep\_bt-mz\_W\_4x4\_sum/profile.cubex  
square scorep\_bt-mz\_W\_4x4\_sum

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep

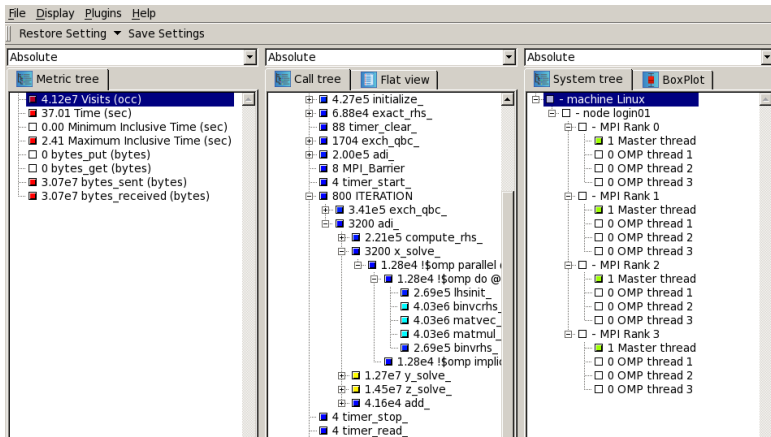
Cube

scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία



## Profiling

N. Νικολου-  
τσάκος

### Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

### Time

### Gprof

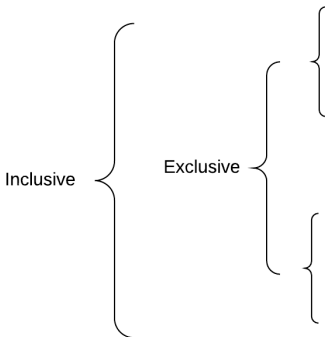
### Scalasca

scorep  
**Cube**  
scalasca

### VTUNE

### Συμπεράσματα

Βιβλιογραφία

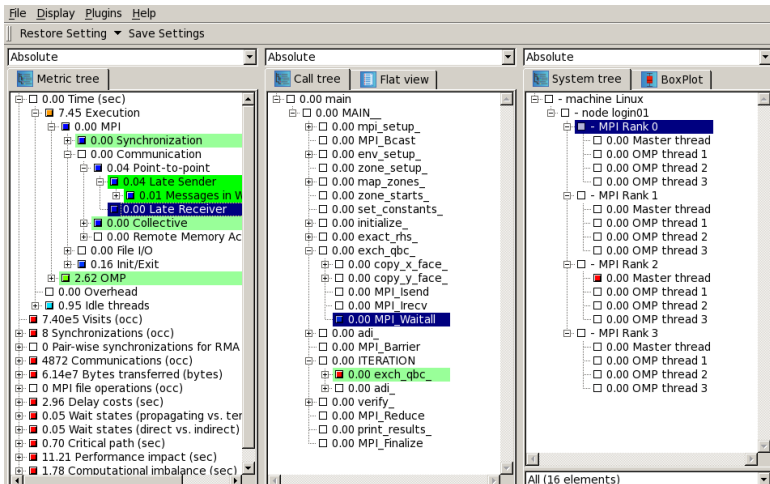


```
int func ()
{
    int x;
    x = 1 + 2;

    myfunc();

    x = x + 2;
    return x;
}
```

scan -t



## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep  
Cube  
scalasca

## VTUNE

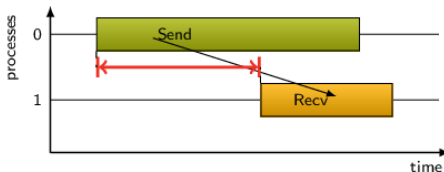
## Συμπεράσματα

Βιβλιογραφία

## Late Receiver Time

### Description:

A send operation may be blocked until the corresponding receive operation is called, and this pattern refers to the time spent waiting as a result of this situation.



Note that this pattern does currently not apply to nonblocking sends waiting in the corresponding completion call, e.g., `MPI_Wait`.

### Unit:

Seconds

### Diagnosis:

Check the proportion of [Point-to-point Send Communications](#) that are [Late Receiver Instances \(Communications\)](#). The MPI implementation may be working in synchronous mode by default, such that explicit use of asynchronous nonblocking sends can be tried. If the size of the message to be sent exceeds the available MPI internal buffer space then the operation will be blocked until the data can be transferred to the receiver: some MPI implementations allow larger internal buffers or different thresholds to be specified. Also consider the mapping of processes onto



## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

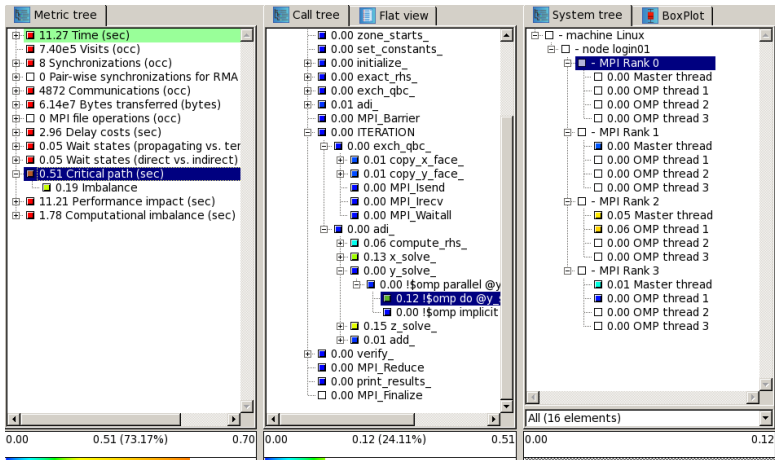
## Scalasca

scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία



## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

- Intel's VTune profiling tool for scalar, multi-threaded and MPI applications
- <http://software.intel.com/en-us/intel-vtune-amplifier-xe>
- “Although VTune can be used with MPI programs, its functionality is aimed at analysing multi-threaded programs.”

## Γραφικό περιβάλλον

amplxe-gui

## Γραμμή εντολών

amplxe-cl

## Profiling

N. Νικολου-  
τσάκος

### Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

### Time

### Gprof

### Scalasca

scorep

Cube

scalasca

### VTUNE

### Συμπεράσματα

Βιβλιογραφία

## Περιβάλλον γραμμής εντολών

```
amplxe-cl -help
```

```
amplxe-cl -help collect
```

```
amplxe-cl -collect hotspots -result-dir  
mydir home/test/sample
```

```
amplxe-cl -R summary -r mydir
```

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία

## Διαθέσιμες επιλογές

- advanced-hotspots: **Advanced Hotspots**
- bandwidth: **Bandwidth**
- concurrency: **Concurrency**
- hotspots: **Basic Hotspots**
- locksandwaits: **Locks and Waits**

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

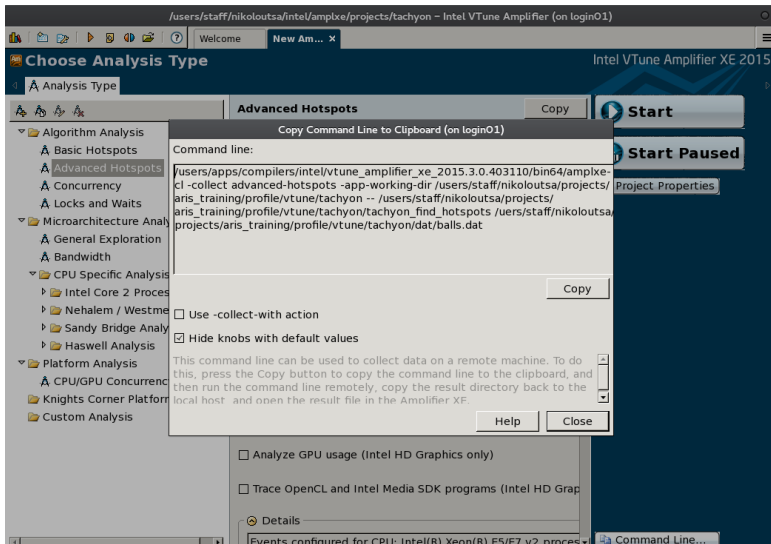
## Scalasca

scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία



Intel VTune Amplifier XE 2015.3.0.403110

Choose Analysis Type

Analysis Type

Advanced Hotspots

Copy Command Line to Clipboard (on login01)

Command line:

```
/users/apps/compilers/intel/vtune_amplifier_xe_2015.3.0.403110/bin64/amplxe-cl -collect advanced-hotspots -app-working-dir /users/staff/nikoloutsas/projects/aris_training/profile/vtune/tachyon -- /users/staff/nikoloutsas/projects/aris_training/profile/vtune/tachyon/tachyon_find_hotspots /users/staff/nikoloutsas/projects/aris_training/profile/vtune/tachyon/dat/balls.dat
```

Copy

Use -collect-with action

Hide knobs with default values

This command line can be used to collect data on a remote machine. To do this, press the Copy button to copy the command line to the clipboard, and then run the command line remotely, copy the result directory back to the local host and open the result file in the Amplifier XE.

Help Close

Analyze GPU usage (Intel HD Graphics only)

Trace OpenCL and Intel Media SDK programs (Intel HD Graphics)

Details

Events configured for CPU: Intel(R) Xeon(R) E5/E7 v2 proces...

Command Line...

## Profiling

N. Νικολου-  
τσάκος

#!/bin/bash

## Χρονοεπίδοση

#SBATCH --job-name=vtune

Παράγοντες  
επίδοσης

#SBATCH --partition=compute

Μετρικές Επίδοσης

#SBATCH --nodes=1

Τεχνικές Μέτρησης

## Time

#SBATCH --ntasks-per-node=16

## Gprof

#SBATCH --time=00:01:00

## Scalasca

#SBATCH --account=testproj

scorep

Cube

scalasca

## VTUNE

export OMP\_NUM\_THREADS=\$SLURM\_CPUS\_PER\_TASK

## Συμπεράσματα

srun ampxe-cl -r myresult -quiet -collect  
advanced-hotspots ./matrix.icc

Βιβλιογραφία

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

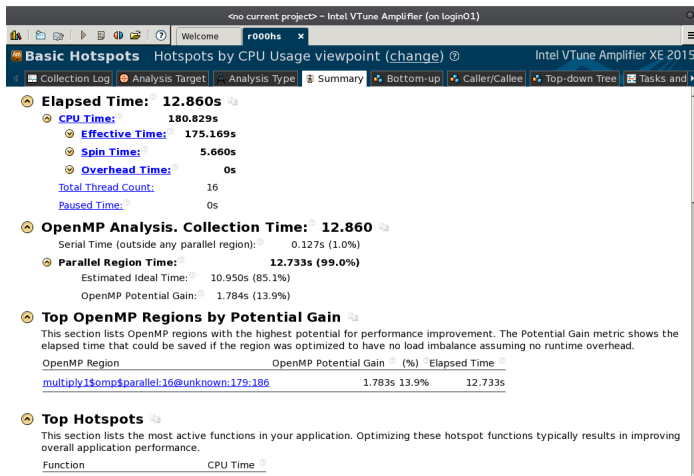
scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία

`amplxe-gui r000hs/r000hs.amplxe`



<no current project> - Intel VTune Amplifier (on login01)

Welcome **r000hs** x

**Basic Hotspots** Hotspots by CPU Usage viewpoint (change) Intel VTune Amplifier XE 2015

Collection Log Analysis Target Analysis Type Summary Bottom-up Caller/Callee Top-down Tree Tasks and

**Elapsed Time: 12.860s**

- CPU Time: 180.829s**
  - Effective Time: 175.169s**
  - Spin Time: 5.660s**
  - Overhead Time: 0s**
- Total Thread Count: 16**
- Paused Time: 0s**

**OpenMP Analysis. Collection Time: 12.860**

- Serial Time (outside any parallel region): 0.127s (1.0%)
- Parallel Region Time: 12.733s (99.0%)**
  - Estimated Ideal Time: 10.950s (85.1%)
  - OpenMP Potential Gain: 1.784s (13.9%)

**Top OpenMP Regions by Potential Gain**

This section lists OpenMP regions with the highest potential for performance improvement. The Potential Gain metric shows the elapsed time that could be saved if the region was optimized to have no load imbalance assuming no runtime overhead.

| OpenMP Region                                               | OpenMP Potential Gain (%) | Elapsed Time |
|-------------------------------------------------------------|---------------------------|--------------|
| <a href="#">multiply1\$omp\$parallel:16@unknown:179:186</a> | 1.783s 13.9%              | 12.733s      |

**Top Hotspots**

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

| Function | CPU Time |
|----------|----------|
|----------|----------|

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

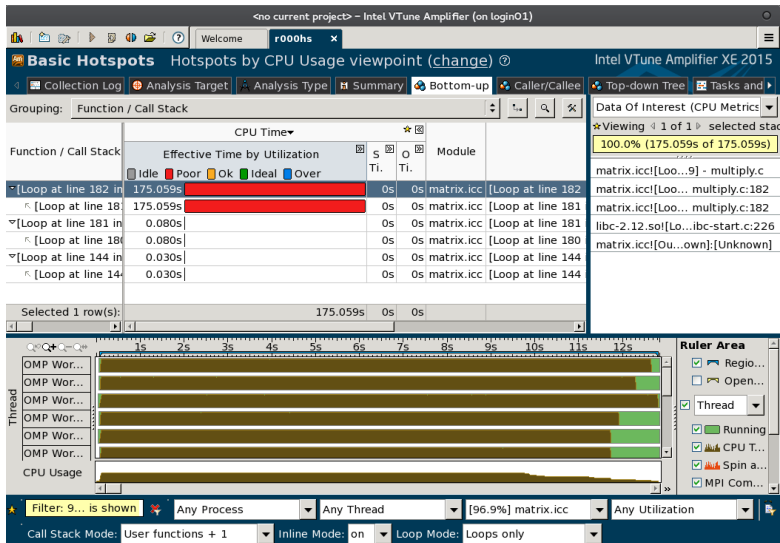
## Scalasca

scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία





## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep

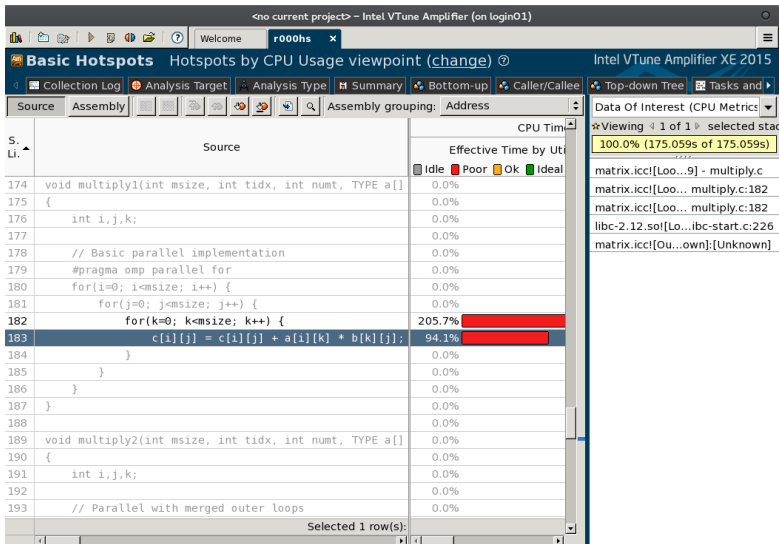
Cube

scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία



## Profiling

N. Νικολου-  
τσάκος

Χρονοεπίδοση

Παράγοντες  
επίδοσης

Μετρικές Επίδοσης

Τεχνικές Μέτρησης

Time

Gprof

Scalasca

scorep

Cube

scalasca

VTUNE

Συμπεράσματα

Βιβλιογραφία

- Υπάρχει πρόβλημα απόδοσης;
  - time, speedup, scalability
- Ποιά είναι η βασική καθυστέρηση; (υπολογισμός, επικοινωνία)
  - MPI/ OpenMP / flat profiling
- Που είναι η βασική καθυστέρηση;
  - Call-path
- Γιατί υπάρχει;
  - hardware count analysis
- Προβλήματα κλιμακωσιμότητας
  - load imbalance, compare profiles various sizes function by function

## Profiling

N. Νικολου-  
τσάκος

## Χρονοεπίδοση

Παράγοντες  
επίδοσης  
Μετρικές Επίδοσης  
Τεχνικές Μέτρησης

## Time

## Gprof

## Scalasca

scorep  
Cube  
scalasca

## VTUNE

## Συμπεράσματα

Βιβλιογραφία



time “*Linux man page*”



*GNU gprof* “Profiling a Program: Where Does It Spend Its Time?”



Scalable performance analysis of large-scale parallel applications “*Scalasca User Guide*”



*Intel® VTune™ Amplifier* “Performance Analysis User’s Guide”



NASA Advanced Supercomputing Division. “*NAS Parallel Benchmarks website*”