14th VI-HPS Tuning Workshop
RIKEN, Kobe, Japan

Performance Analysis with Vampir

Presenter: Jens Domke
GSIC, Tokyo Institute of Technology
Outline

Part I: Welcome to the Vampir Tool Suite
   – Mission
   – Event Trace Visualization
   – Vampir & VampirServer
   – Vampir Displays

Part II: Vampir Hands On
   – Visualizing and analyzing NPB-MZ-MPI / BT

Part III: Summary and Conclusion
Performance tools will not automatically make you code run faster. They help you understand, what your code does and where to put in work.
Mission

• Visualization of dynamics of complex parallel processes
• Requires two components
  – Monitor/Collector (Score-P)
  – Charts/Browser (Vampir)

Typical questions that Vampir helps to answer:
  – What happens in my application execution during a given time in a given process or thread?
  – How do the communication patterns of my application execute on a real system?
  – Are there any imbalances in computation, I/O or memory usage and how do they affect the parallel execution of my application?
Event Trace Visualization with Vampir

- Alternative and supplement to automatic analysis
- Show dynamic run-time behavior graphically at any level of detail
- Provide statistics and performance metrics

**Timeline charts**
- Show application activities and communication along a time axis

**Summary charts**
- Provide quantitative results for the currently selected time interval
• Directly on front end or local machine

```
% module load UNITE
% module load vampir
% vampir
```
Vampir – Visualization Modes (2)

- On local machine with remote VampirServer

```bash
% module load UNITE vampirserver
% vampirserver start -n 12
% module load UNITE vampir
% vampir
```

---

**Many-Core Program**

**Score-P**

**VampirServer**

**LAN/WAN**

**Large Trace File (OTF2)**

**MPI parallel application**

**1.0 ms**

**2.0 ms**
Usage order of the Vampir Performance Analysis Toolset

1. Instrument your application with Score-P

2. Run your application with an appropriate test set

3. Analyze your trace file with Vampir
   - Small trace files can be analyzed on your local workstation
     1. Start your local Vampir
     2. Load trace file from your local disk
   - Large trace files should be stored on the HPC file system
     1. Start VampirServer on your HPC system
     2. Start your local Vampir
     3. Connect local Vampir with the VampirServer on the HPC system
     4. Load trace file from the HPC file system
Main displays of Vampir

- **Timeline Charts:**
  - Master Timeline
  - Process Timeline
  - Counter Data Timeline
  - Performance Radar

- **Summary Charts:**
  - Function Summary
  - Message Summary
  - Process Summary
  - Communication Matrix View
Main displays of Vampir

Master Timeline

Process Timeline

Counter Data Timeline

Function Summary

Communication Matrix View

Function Legend

Process Summary

Context View
Vampir hands-on

Visualizing and analyzing NPB-MZ-MPI / BT
Option A: Overview

Step 1: Start VampirServer on fx10pi

Step 2: Port forwarding to fx10pi

Step 3: GUI on your machine
Option A: Step 0, Install the Vampir Client on your Laptop

- Start a new shell on your laptop
- Copy the appropriate Vampir package from fx10pi to your laptop

```
% ls /opt/VI-HPS/vampir/remote/vampir*
vampir-8.2.0-linux-x64.bin vampir-8.2.0-win-x64.exe vampir-8.2.0-mac.dmg
vampir-8.2.0-linux-x86.bin vampir-8.2.0-win-x86.exe vampir.license
% scp <user>@pi.ircpi.kobe-u.ac.jp:/opt/VI-HPS/vampir/remote/vampir/<version> ./
% scp <user>@pi.ircpi.kobe-u.ac.jp:/opt/VI-HPS/.../vampir/vampir.license ./
```

- Install and start the Vampir GUI (example linux-x64)

```
% ls
vampir-8.2.0-linux-x64.bin
vampir.license
% bash ./vampir-8.2.0-linux-x64.bin --instdir=$HOME/vampir-gui
...
Do you want to install Vampir to /home/domke/vampir-gui? [y/N]: y
Do you want to create a desktop icon? [Y/n]: n
Do you want to add menu items? [Y/n]: n
Do you want to register *.otf and *.otf2 trace files? [Y/n]: n
```
Option A: Step 0, Install the Vampir Client on your Laptop – License

- Select the `vampir.licence` which you copied before from fx10pi
Option A: Step 1, Start a Vampirserver on fx10pi

- **Load modules**

```bash
% module use /opt/vi-hps/UNITE/local
% module load UNITE
UNITE loaded
% module load vampirserver
Vampirserver 8.2 loaded
```

- **Start a vampirserver w/ one analyzing thread**

```bash
% vampirserver start -n 1
Launching VampirServer...
VampirServer 8.2.0  (r8690)
Licensed to VI-HPS Tools Workshop 03/2014
Running 1 analysis process... (abort with `vampirserver stop 29640`)
VampirServer <29640> listens on: pi:31088

Please run:
```bash
  ssh -L 30001:pi:31088 <user>@pi.ircpi.kobe-u.ac.jp
```
on your desktop to create ssh tunnel to VampirServer.

Start vampir on your desktop and choose 'File -> Remote Open' (Linux/Win)or 'Connect to other' (Mac OS X) with:
- Description: fx10pi,  Server: localhost,  Port: 30001
Option A: Step 2, Port Forwarding to fx10pi

• Write down the host on which the server runs

% vampirserver start -n 1
...
VampirServer <29640> listens on: pi:31088
...
Please run:
ssh -L 30001:pi:31088 <user>@pi.ircpi.kobe-u.ac.jp
...

• Establish Port Forwarding from your local machine to fx10pi

% ssh -L 30001:pi:31088 <user>@pi.ircpi.kobe-u.ac.jp

Remember the 30001, it’s needed later
Option A: Step 2, Port Forwarding to K/ppb

1. Write down the host on which the server runs

```bash
% ssh ppb
% export PATH=$PATH:/opt/aics/scalasca/vampir/bin
% vampirserver start
vampirserver start
Launching VampirServer...
VampirServer 8.2.0  (r8690)
Licensed to VI-HPS Tools Workshop 03/2014
Running 1 analysis process... (abort with vampirserver stop 22973)
VampirServer <22973> listens on: fe02s01:30094
```

Please run:

```bash
ssh -t -L 40094:localhost:40094 <user>@k.aics.riken.jp \\
'ssh -L 40094:fe02s01:30094 <user>@ppb'
```

on your desktop to create ssh tunnel to VampirServer.

Start vampir on your desktop and choose 'File -> Remote Open' (Linux/Win) or 'Connect to other' (Mac OS X) with:

Description: ppb,  Server: localhost,  Port: 40094

2. Establish Port Forwarding to K/ppb

```bash
% ssh -t -L 40094:localhost:40094 <user>@k.aics.riken.jp \\
'ssh -L 40094:fe02s01:30094 <user>@ppb'
```
Help! Where is my trace file?

- If you followed the Score-P hands-on up to the trace experiment, yours is in:

```bash
% ls ~/<local tutorial folder>/bin.scorep/bt-mz_B.8x4_pi
profile.cubex  scorep.filt  scout.cubex  summary.cubex  traces/
traces.otf2  scorep.cfg  scorep.log  scout.log  trace.cubex
traces.def  trace.stat
```

That’s what you will open with Vampir

- If you removed the trace or did not follow to that point, copy a prepared trace

```bash
% cd ~/<local tutorial folder>/bin.scorep
% cp /opt/vi-hps/UNITE/samples/bt-mz_B.8x4_pi/scorep_trace_with_metrics ./
```
Option A: Step 3, Open (1) a Trace w/ Vampir Client on your Laptop

Use the “Open Other” option
Option A: Step 3, Open (2) a Trace w/ Vampir Client on your Laptop

Select “Remote File”
Option A: Step 3, Open (3) a Trace w/ Vampir Client on your Laptop

Server is “localhost”

Connection type “Socket”

Port is “30001”
Option A: Step 3, Open (4) a Trace w/ Vampir Client on your Laptop
Option B: Starting Vampir on fx10pi (or K/ppb)

- **Load modules**

  ```
  % ssh -XC <user>@pi.ircpi.kobe-u.ac.jp
  % module use /opt/vi-hps/UNITE/local
  % module load UNITE
  UNITE loaded
  % module load vampir
  Vampir 8.2 loaded
  ```

- **Start a vampir as an X11 job**

  ```
  % vampir &
  ```

  - The GUI comes up running on the login node
  - Use the following to connect to fx10pi (or K/ppb):
    ```
    $ ssh -XC ...
    ```
  - This enables X11 forwarding AND compression, the latter is crucial for a good usage experience

  Don’t do this this, unless everything else fails. X11 forwarding for all participants via WLAN won’t work.
Vampir: Visualization of the NPB-MZ-MPI / BT trace

Master Timeline

Navigation Toolbar

Function Summary

Function Legend
Master Timeline

Detailed information about functions, communication and synchronization events for collection of processes.
Vampir: Visualization of the NPB-MZ-MPI / BT trace

Process Timeline

Detailed information about different levels of function calls in a stacked bar chart for an individual process.
Typical program phases

- Initialisation Phase
- Computation Phase
Counter Data Timeline

Detailed counter information over time for an individual process.
Performance Radar

Detailed counter information over time for a collection of processes.
Vampir: Visualization of the NPB-MZ-MPI / BT trace

Zoom in: Initialisation Phase

Context View: Detailed information about function “initialize_”.
Feature: Find Function

Execution of function “initialize_” results in higher page fault rates.
Computation Phase

Computation phase results in higher floating point operations.
Zoom in: Computation Phase

MPI communication results in lower floating point operations.
Zoom in: Finalisation Phase

“Early reduce” bottleneck.
Vampir: Visualization of the NPB-MZ-MPI / BT trace

Function Summary:
Overview of the accumulated information across all functions and for a collection of processes.

Process Summary:
Overview of the accumulated information across all functions and for every process independently.
Process Summary

Find groups of similar processes and threads by using summarized function information.
Option A: Step 4, Don’t forget to stop the running Vampirserver

- Stop the vampirserver

```
% vampirserver start -n 1
Launching VampirServer...
VampirServer 8.2.0  (r8690)
Licensed to VI-HPS Tools Workshop 03/2014
Running 1 analysis process... (abort with vampirserver stop 29640)
VampirServer <29640> listens on: pi:31088 ...

% vampirserver stop 29640
Shutting down VampirServer <29640>...
VampirServer <29640> is down.
```
Summary and Conclusion
Summary

- Vampir & VampirServer
  - Interactive trace visualization and analysis
  - Intuitive browsing and zooming
  - Scalable to large trace data sizes (20 TiByte)
  - Scalable to high parallelism (200,000 processes)

- Vampir for Linux, Windows, and Mac OS X

- Note: Vampir does neither solve your problems automatically nor point you directly at them. It does, however, give you FULL insight into the execution of your application.
Conclusion

• Performance analysis very important in HPC

• Use performance analysis tools for profiling and tracing

• Do not spend effort in DIY solutions, e.g. like printf-debugging

• Use tracing tools with some precautions
  – Overhead
  – Data volume

• Let us know about problems and about feature wishes

• vampirsupport@zih.tu-dresden.de
Vampir is available at http://www.vampir.eu, get support via vampirsupport@zih.tu-dresden.de