BEST PRACTICES FOR PERFORMANCE ENGINEERING TRAINING

MAY 18, 2020 | BERND MOHR
CONTEXT: HPCCF SKILLS
BACKGROUND: WHO AM I / WHO ARE WE?

- **SoftWare Analysis and Tools (SWAT) Team**
- Jülich Supercomputing Centre (JSC), Germany

**Tasks**
- **Service**: performance analysis of HPC applications of JSC users and POP customers
- **Development**: Score-P, Scalasca, Cube tools
- **Research**: parallel performance analysis methods and tools
- **Education and Training**
  - 10 to 15 events per year for last 15 years
  - Half- and full-day tutorials at conferences like SC, ISC, …
  - 1 to 3 day tool trainings at HPC centres and Universities
  - 3 to 5 day **VI-HPS Tuning Workshops**
Virtual Institute – High Productivity Supercomputing

- **Goal**: Improve the quality and accelerate the development process of complex simulation codes running on highly-parallel computer systems
- **Start-up funding** (2006–2011) by Helmholtz Association of German Research Centres
- **Activities**
  - Development and integration of HPC programming tools
  - Correctness checking & performance analysis
  - Academic workshops
  - Training workshops

http://www.vi-hps.org
Technologies and their integration

- **KCACHEGRIND**
- **PAPI**
- **MUST / ARCHER**
- **DDT**
- **STAT**
- **MEMCHECKER / SPINDLE / SIONLIB**

**MAP/PR / MPIP / O|SS / MAQAO**
- **Hardware monitoring**
- **Automatic profile & trace analysis**

**TAU**
- **EXTRA-P**
- **PERISCOPE**
- **SCALASCA**
- **SCORE-P / EXTRAE**
- **VAMPIR**
- **PARAVER**
- **PTF / RUBIK / MAQAO**

**JUBE**
- **Debugging, error & anomaly detection**
- **Visual trace analysis**

**Execution**
- **Optimization**
VI-HPS training & Tuning Workshops

- **Goals**
  - Give an overview of the programming tools suite
  - Explain the functionality of individual tools
  - Teach how to use the tools effectively
  - **Offer hands-on experience and expert assistance using tools**
  - Receive feedback from users to guide future development

- **For best results, bring & analyze/tune your own code(s)!**

- **VI-HPS Hands-on Tutorial series**
  - SC’08-11/13-17/19, ICCS’09, Cluster’10, EuroMPI’12/14, XSEDE’13, ISC-HPC’15-19

- **VI-HPS Tuning Workshop series**
  - 2008 (x2), 2009 (x2), 2010 (x2), 2011 (x2), 2012 (x2), 2013 (x2), 2014(x4), 2015(x3)
  - 2016 (Kobe/Japan, Garching/Germany, Cambridge/UK, Livermore/USA)
  - 2017 (Southampton/UK, Aachen/Germany, Bruyères-le-Châtel/France)
  - 2018 (Garching/Germany, London/UK, Reims/France)
  - 2019 (Barcelona/Spain, Knoxville/USA, Bristol/UK, Jülich/Germany)
VI-HPS Tuning Workshop series

1. 2008/03/05+3: RWTH, Aachen, Germany
2. 2008/10/08+3: ZIH, Dresden, Germany
3. 2009/02/16+5: JSC, Jülich, Germany
4. 2009/09/09+3: HLRN, Dresden, Germany
5. 2010/03/08+3: TUM, Garching, Germany
6. 2010/05/26+3: SARA, Amsterdam, Netherlands
7. 2011/03/28+3: HLRS, Stuttgart, Germany
8. 2011/09/05+5: GRS, Aachen, Germany
9. 2012/04/23+5: UVSQ, St-Quentin, France
10. 2012/10/16+4: LRZ, Garching, Germany
11. 2013/04/22+4: MdS, Saclay, France
12. 2013/10/07+5: JSC, Jülich, Germany
13. 2014/02/10+5: BSC, Barcelona, Spain
14. 2014/03/25+3: RIKEN AICS, Kobe, Japan
15. 2014/04/07+4: MdS, Saclay, France
16. 2014/04/29+3: EPCC, Edinburgh, Scotland
17. 2015/02/23+5: HLRS, Stuttgart, Germany
18. 2015/05/18+5: UGA, Grenoble, France
19. 2015/10/27+3: LLHPC, Santiago, Chile
20. 2016/02/24+3: RIKEN AICS, Kobe, Japan
21. 2016/04/18+5: LRZ, Garching, Germany
23. 2016/07/27+3: LLNL, Livermore, California, USA
25. 2017/03/27+5: RWTH, Aachen, Germany
26. 2017/10/16+5: Lab. ECR, Ter@tec, France
27. 2018/04/23+5: LRZ, Garching, Germany
29. 2018/10/15+5: ROMEO, Reims, France
30. 2019/01/21+5: BSC, Barcelona, Spain
31. 2019/04/09+4: UTK-ICL, Knoxville/TN, USA
33. 2019/06/24+5: JSC, Jülich, Germany
LESSONS LEARNED FROM TRAINING

• Performance tools are not easy-to-use and intuitive
  • Relatively easy to collect and display performance data (typically shown in tutorials)
  • But hard to draw the right conclusions from them (typically not shown in tutorials)
  • So, users tend to use their own methods (like printf debugging)

• Performance tools rarely work out-of-the-box for user applications
  • Differences in HW + SW stack and build systems used for applications

⇒ Important to demonstrate usefulness of tool
  ⇒ For the users application
  ⇒ During the training
POP CoE (https://pop-coe.eu)

- A Centre of Excellence
  - On Performance Optimisation and Productivity
  - Promoting best practices in parallel programming

- Providing FREE Services
  - Precise understanding of application and system behaviour
  - Suggestion/support on how to refactor code in the most productive way

- Horizontal
  - Transversal across application areas, platforms, scales

- For (EU) academic AND industrial codes and users!
Partners

• Who?
  • BSC, ES (coordinator)
  • HLRS, DE
  • IT4I, CZ
  • JSC, DE
  • NAG, UK
  • RWTH Aachen, IT Center, DE
  • TERATEC, FR
  • UVSQ, FR

A team with
  • Excellence in performance tools and tuning
  • Excellence in programming models and practices
  • Research and development background AND
    proven commitment in application to real academic and industrial use cases
FREE Services provided by the CoE

• **Parallel Application Performance Assessment**
  • Primary service
  • Identifies performance issues of customer code (at customer site)
  • If needed, identifies the root causes of the issues found and qualifies and quantifies approaches to address them (recommendations)
  • Combines former Performance Audit (?) and Plan (!)
  • Medium effort (1-3 months)

• **Proof-of-Concept (✓)**
  • Follow-up service
  • Experiments and mock-up tests for customer codes
  • Kernel extraction, parallelisation, mini-apps experiments to show effect of proposed optimisations
  • Larger effort (3-6 months)

Note: Effort shared between our experts and customer!
Make Performance Assessment simpler ...

- Goal: Describe Application performance with one number
- Based on Efficiencies
  - (0,1] value range
  - Multiplicative model
Efficiencies

The following metrics are used in a POP Performance Audit:

- **Global Efficiency (GE):** $GE = PE \times CompE$
  - **Parallel Efficiency (PE):** $PE = LB \times CommE$
    - **Load Balance** Efficiency (LB): $LB = \frac{\text{avg}(CT)}{\text{max}(CT)}$
    - **Communication** Efficiency (CommE): $CommE = SerE \times TE$
      - **Serialization Efficiency (SerE):** $SerE = \max \left( \frac{CT}{TT} \right)$ on ideal network
      - **Transfer Efficiency (TE):** $TE = \frac{TT}{TT}$ on ideal network / TT
  - **(Serial) Computation** Efficiency (CompE)
    - Computed out of IPC Scaling and Instruction Scaling
    - For strong scaling: ideal scaling -> efficiency of 1.0

Details see [https://sharepoint.ecampus.rwth-aachen.de/units/rz/HPC/public/Shared%20Documents/Metrics.pdf](https://sharepoint.ecampus.rwth-aachen.de/units/rz/HPC/public/Shared%20Documents/Metrics.pdf)
• Learn how to efficiently use the tools used in the project
• Learn about the POP metrics and methodology

• Currently project-internal only
  • Get new project members started
  • Educate existing members with advanced topics

• Potentially in the future
  • Provide performance analyst training for others
  • “Certified POP Performance Analyst”