

# Software Tools Live Notes

Open to all to write down your questions and observations - this is particularly useful to ensure follow up of open points.

After each meeting we will convert this document into the meeting minutes for the HSF website.

*Attendees: Sebastien P, Ben C, Marco Ca, Andre A, Marko P, Anna Z, Maciej Szymanski, Sebastien (SFT), Peter S, Andrei G, Andrea V, Guilo E, Graeme S, + Vidyo (15), Peter S.*

## Introduction

- 

## Profilers

- TODO - document tools and present them.
- Storing data - how useful is it to compare different tools (quantum effects) / different applications? Different tools for different aspects.
  - Lowest common denominator only?
- CodeCov extensions?
- Reuse tools
- Time, time, ... ?
- Andrea - main goal is to correlate with a line of code. Overlap with more deployment and infrastructure monitoring. How well does an application work on different architectures. Performance and Cost Model Group (WLCG) looking at this.
- Hooks to understand where certain parts of an application start/end are useful. E.g., identify the start of the event loop.
  - Would like to be able to profile (if needed) on the grid. Debug end to end.
- Is the idea of a warehouse useful? Could this be a service from CERN IT?
  - It's useful. Need to be a concrete proposal.
- Some vision of how to come up with common formats, but needs some expert input to put this into a DB that would scale.
- How do you want to analyse the data? This informs what we want to store. Maybe not so much need for long term archiving of the data.
  - Need a lot of metadata to link it back to the right code version.
  - Too much churn in the code - a lot of commits every night.
- Is it worthwhile to compare between different tools - they measure different things (explains the zoo of the tools - they do different things).
  - There have been some useful examples of cross profiling..
  - Also now much easier to analyse large profile dumps.

- Keep more 'raw' data and analyse on the fly.
- Is it possible extract data from VTune? Probably not - proprietary.
- Usefulness depends on the lifetime of the software.
  - Software in production becomes of historical interest; at least some summary data is important.
  - The bleeding edge has a useful lifetime of only a few days/weeks during the development cycles.
- Comparison between different developers work could also be useful as a collaborative tool.
- ATLAS SPOT monitoring is just a 'red flag' style.
  - Something more generic would help and share efforts between experiments.
- Progress by gathering people who could work on a prototype of the web based presentation.
- Bootstrap guide for different tools would be useful.
  - There is some training that people have developed and we should inventory that and point to it.
  - Getting users started is a very good thing to contribute to.
- Peter - as a developer got some very incomprehensible information back to try and understand a difficult problem. Developers are very busy and under a lot of pressure.
  - Need to optimise the efficiency of our coders - what we want to develop has to help that.
  - CMS igprof experience was that providing a *service* to developers did help a lot - remove.

## Static Analysers

-