Achilles

How to create apps for automatic design analysis
("How to find hidden gold in your logfiles in < 15 minutes")

Aaron Ng - Oct 9, 2013

Agenda

- The legend of Achilles
- The legend of Vivado
- Goals of this presentation
- Is there hidden gold in your logfiles?
- Who Achilles is for
- Demo
- How to start using Achilles for your current project (in 15 minutes or less)
- Q&A

The Legend of Achilles *(uh-kil-eez)*

The Legend of Achilles

The Legend of Achilles

The Legend of Vivado

- Vivado is strong
  - But there are always weaknesses
  - … and opportunities

- Achilles project
  - automated system to expose opportunities and weaknesses in Vivado
  - … that anyone can use right away (without fisusr, infra or IT support)

Goals of this presentation

- Make you powerful

- Fast-track you to the state of the art in the software world
  - trend: desktop → cloud; monolithic → distributed
  - programs replaced by web applications
  - “cloud computing” framework (used by Gmail, YouTube, Google+, etc)

- Show you how to create your first web application
  - extract data from an existing ABE run’s logfiles
  - analyze the data to reveal **opportunities** and **weaknesses** in Vivado
  - *in under 15 minutes*

- Make our group the most sophisticated in the company
  - those who missed this presentation will be like cavemen without fire

How to use modern technology to find hidden gold

- The old way of reporting data:
  - raw data → static PERL scripts → static reports

- Analogy: Mining – new technology lets you revisit old mines
  - resources not efficiently harvested with old tech
  - resources not reachable with old tech
  - “hidden gold” right under your nose… but need new technology to tap

- New powerful way (thanks to modern technology):
  - raw data → **Database** → **Web Application** → **Interactive Reports**
  - raw data → **SQL** → **web2py** → **Web Browser**

Model – View – Controller framework

- General pattern:
  - raw data → Model → Controller → View

- Can create many "controllers" to process data in DB
  - raw data → DB → Controller → View
```
                  |→ Controller → View
                  |→ Controller → View
```

- Upcoming examples in demo:
  - raw data → DB → "QOR spy" → HTML compare placer phase WNS
```
                  |→ "Critpath spy" → HTML investigate worst paths
```

Who Achilles is for

- Developers
  - Discover opportunities for new optimizations
  - Test your idea & compute expected gains before writing any code
  - Analyze effectiveness of your optimization after writing code
  - Nightly tracking of your code's health

- Managers
  - Explore opportunities for new projects
  - Detect degradations and triage problems quickly across nightlies
  - Track metrics (can go deeper than just final numbers in Excel sheet)
  - Personal/group quality "dashboard" or "command center"

How to start using Achilles for your projects (in 15 minutes or less)

- Demo starting now (Achilles – web2py / python / PERL)
- Instructions to follow along: http://confluence/pages/viewpage.action?pageId=10174117

Magic show

- Demo – using your existing logfiles, show you “hidden gold”
  - Gold you might miss with “old technology”

- *Example 1:* Compare RDS / SDS QOR by phase
  - “QOR Blamefinder”
- *Example 2:* Analyze post-Place or post-Route critical paths
  - Any common symptoms? Anything look fishy?
- *Example 3:* Prototyping / validate your idea
  - Test opportunities / How much gain can you expect?
- *Example 4:* Compare critical paths before/after your opt
  - Expected wins? Unexpected loss? Something else became critical?

Try it

- "How to start using Achilles for your current project (in 15 minutes or less)": http://confluence/pages/viewpage.action?pageId=10174117
- Questions?
- Ideas?

Appendix

- Approx retime gain:
  - … → FD → LUT → …
  - … → LUT’ → FD →→ …