shinyloadtest Webinar

load testing Shiny apps

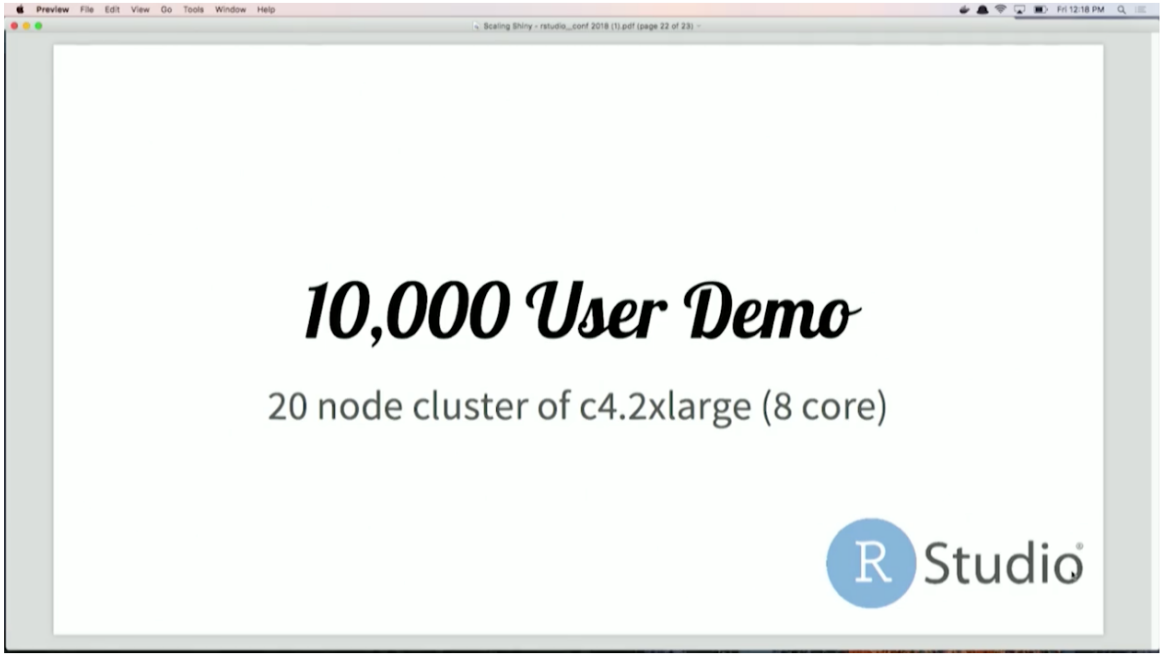
Alan Dipert (@alandipert)

September 5, 2018

Motivation

Shiny doesn't scale? Ha!

Scaling Shiny



DOWNLOAD MATERIALS

Abstract

About the speaker

Sean Lopp
Solutions Engineer, RStudio

Sean has a degree in mathematics and statistics and worked as an analyst at the National Renewable Energy Lab before making the switch to customer success at RStudio. In his spare time he skis and mountain bikes and is a proud Colorado native.

Public Release

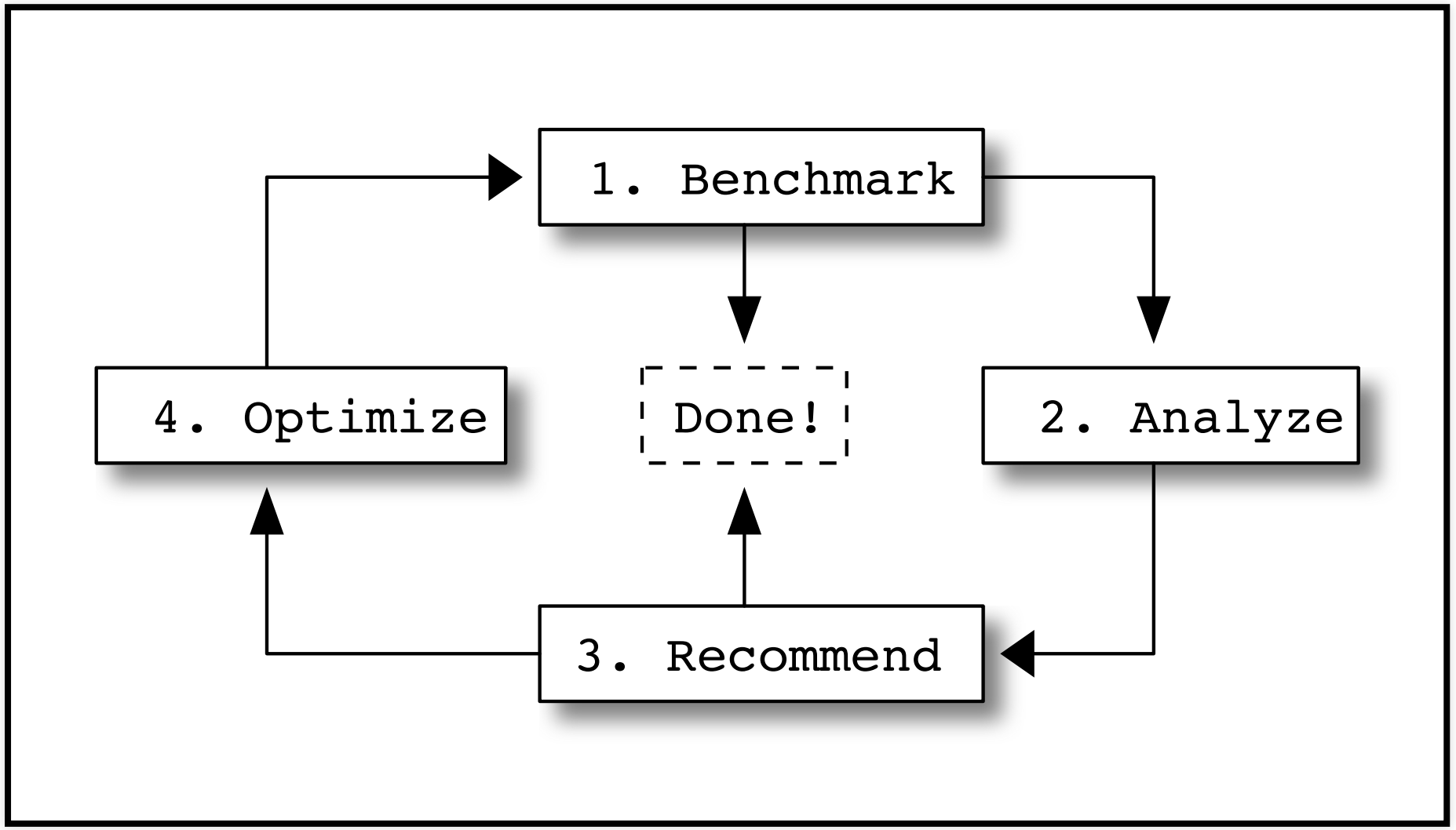
Today is public release of tools used for rstudio::conf 2018 10,000 user demo.

1. `shinyloadtest`: recording and analysis
   - https://github.com/rstudio/shinyloadtest
2. `shinycannon`: load generation
   - https://github.com/rstudio/shinycannon

Agenda

1. Introduce optimization loop methodology
2. Use `shinyloadtest` to record a session with a Shiny app
3. Generate load with `shinycannon`
4. Analyze metrics with `shinyloadtest`
5. Make changes to application
6. Generate load and analyze again

Optimization Loop Method



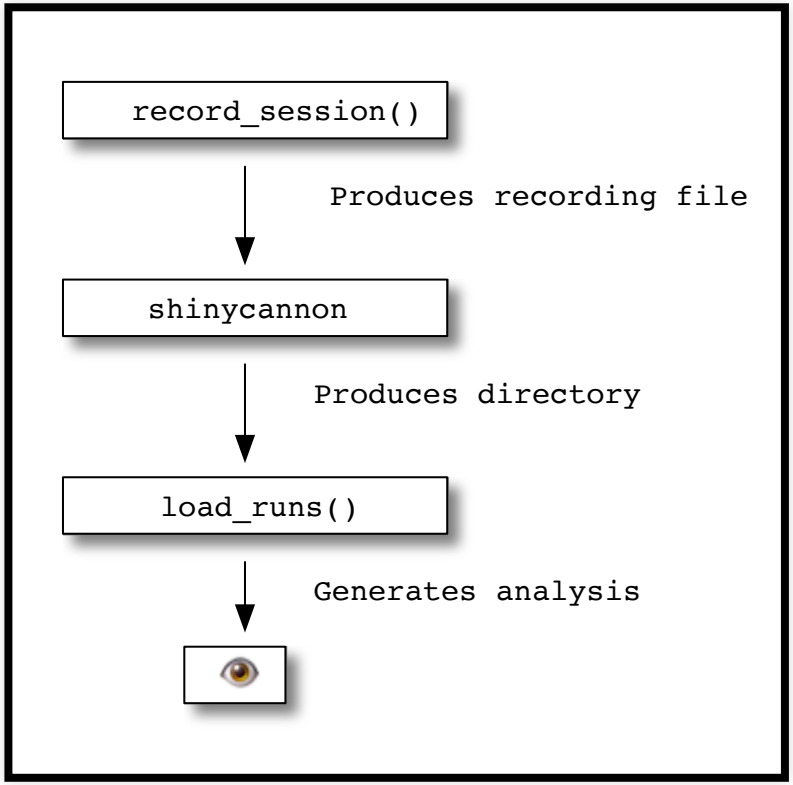
Optimization Loop

- *Benchmark*: Use `shinyloadtest::record_session` to record interaction, `shinycannon` to simulate users.
- *Analyze*: Visualize and interpret the metrics.
- *Recommend*: Propose ways for the capacity of the application to be increased.
- *Optimize*: Implement recommendations and benchmark again. Repeat until satisfied.

Example application

Sleepy Geisers

Workflow


record_session()
Produces recording file
shinycannon
Produces directory
load_runs()
Generates analysis

record_session

Install shinyloadtest and then:

```
library(shinyloadtest)
record_session('https://beta.rstudioconnect.com/content/3912/',
  output_file = 'geiser30seconds.rec')
```

`shinycannon` Run 1

Run our recording using 1 worker for at least 2 minutes and output to `run1`

```
shinycannon geiser30seconds.rec
https://beta.rstudioconnect.com/content/3912/ --workers 1 --
loaded-duration-minutes 2 --output-dir run1
```

`shinycannon` Run 2

Run our recording using 1 worker for at least 2 minutes and output to `run1`

```
shinycannon geiser30seconds.rec
https://beta.rstudioconnect.com/content/3912/ --workers 5 --
loaded-duration-minutes 2 --output-dir run2
```

Analyze

Generate a report and see if app was usable for the simulated users

```
library(shinyloadtest)
df <- load_runs("Run 1" = "run1", "Run 2" = "run2")
shinyloadtest_report(df, "report1.html")
```

Profile app or tune deployment

- Profile app and tune performances
- OR tune deployment/hosting parameters (scheduling)

Test again

After making our changes and redeploying the application, we can generate load and analyze again to see if our changes helped.

Thank you!

- https://github.com/rstudio/shinyloadtest
- https://github.com/rstudio/shinycannon