

# FusionReactor Webinar:

Troubleshooting with FR, part 3:  
When requests are slow for less obvious reasons

**INTERGRAL**  
information solutions

 fusion  
reactor™

## Introductions



Charlie Arehart

Independent Consultant, [CArehart.org](http://CArehart.org)

(Focused on server troubleshooting)

## Agenda (nearly all demos)

- Foreword
- Stack tracing
- Profiling requests
- CPU sampler
- Memory profiler
- What's coming in Part 4
- Resources for learning more
- Questions & answers

# Foreword

- Third of a planned 4-part series:
  - Troubleshooting with FR, part 1: What has just happened on my server?
  - Troubleshooting with FR, part 2: Why are requests/transactions running slowly?
  - Troubleshooting with FR, part 4: Post-crash troubleshooting
- Audience: presumed to already be using FR
  - But may be overwhelmed by richness of FR, where to turn to solve problems
- Concepts apply generally to any Java/CFML server that FR can monitor
- Preso is being recorded, so you will be able to revisit details
- Also, note that FR 8 was released last week, with interesting new features

## Quick Recap of Parts 1&2

- Warning not to presume you know “what’s going wrong”
- Overview of current and recent processing, via “Web Metrics” feature
  - And Requests Activity and History/Slow/Longest/Error History pages
  - Resource usage tracked via FR’s CPU and memory tracking, garbage collection tracking
- JDBC, httpclient/cfhttp, and other transactions within a request
  - And over history, slow, longest; across all apps
- Today we move on to understanding when requests are slow for less obvious reasons
  - Seeing JVM-level details about running or finished slow requests
  - Or high-cpu threads, or memory-hungry java objects

# Demos

- Stack tracing
- Profiling requests
- CPU sampler
- Memory profiler

# Conclusion

- Today we saw features that help when those covered in parts 1&2 do not
  - Stack tracing: what's going on in a request now
  - Profiling requests: what happened in slow requests in the past
  - CPU sampler: what threads (request or otherwise) are using CPU now
  - Memory profiler: what objects are using the most heap, from Java perspective

## Again, this is part three in a series

- The final part will be presented in two weeks:
  - Troubleshooting with FR, part 1: **What has just happened on my server?** (4 weeks ago)
  - Troubleshooting with FR, part 2: **Why are requests/transactions running slowly?** (2 wks ago)
  - Troubleshooting with FR, part 3: **When requests are slow for less obvious reasons** (today)
  - Troubleshooting with FR, part 4: **Post-crash troubleshooting** (Mar 14)
    - Covering logs, archived metrics, FR alerts, and FR Cloud alerts and profiling
- You can watch recordings of all past ones : [fusion-reactor.com/webinars](https://fusion-reactor.com/webinars)
- Again, even these 4 parts have not covered ALL available FR features
  - Just key ones to help with typical troubleshooting
  - Be sure to see the 15 webinars created before this series, also available there



## Other FR resources

- **FR web site: [fusion-reactor.com](http://fusion-reactor.com)**
  - Downloads
  - Docs, videos, technotes, forums, and much more
- **Email: [sales@fusion-reactor.com](mailto:sales@fusion-reactor.com), [support@fusion-reactor.com](mailto:support@fusion-reactor.com)**
- **Phone: (978) 496-9990 (sales)**
- **Consulting assistance: [cfconsultant.com](http://cfconsultant.com)**
- We welcome your feedback on these or the other webinars, or any you would like to see

A photograph of several people in a dimly lit office or server room. They are sitting at desks with computer monitors, looking intently at the screens. One person in the foreground is resting their chin on their hand, appearing thoughtful. The room is illuminated by desk lamps, creating a focused and professional atmosphere.

# Questions & Answers