# 2015-11-09 Performance - Scale Meeting

## Time/Place

- Time: 1:00pm Eastern Standard Time US (UTC-5)
- Dial-in Number: (712) 775-7035
    - Participant Code: 479307#
    - International numbers: Conference Call Information
    - Web Access: https://www.freeconferencecallhd.com/wp-content/themes/responsive/flashphone/flash-phone.php

## Attendees

- Andrew Woods (DuraSpace)
- Unknown User (daniel-dgi) (discoverygarden)
- Aaron Elkiss (University of Michigan)
- Esmé Cowles (Princeton University)
- Michael J. Giarlo (Stanford University/Hydra-in-a-Box project)
- Bill Branan (DuraSpace/Hydra-in-a-Box project)
- David Wilcox (DuraSpace)
- Brad Spry (UNC Charlotte)

## Agenda

1. Review assessed relevance and reusability of prior work in Fedora testing
    a. F4 performance benchmarking (summary)
    b. Unimplemented "Technical Working Group" performance assessment plan
2. Establish consensus on categories of next round of F4 performance benchmarking
3. Define actions towards initial benchmarking
4. Define actions towards collecting representative datasets and infrastructure
5. Next meeting? Tues Dec 1st or Thurs Dec 3rd?

## Minutes

1. Prior performance benchmarking and assessment work
    a. Three of the performance areas highlighted previously have not yet been sufficiently re-tested
        i. total data size
        ii. ingest rate
        iii. LDP/SPARQL Update performance (per Hydra practice)
    b. Clustering
        i. Primary use case seems to be high availability
        ii. What increased scale clustering affords is unclear, partly because we haven't yet fully probed how far a single instance scales (as a baseline)
    c. Would be good to know how performance (response time) changes:
        i. as a file size increases
        ii. as # of files increases
        iii. as # of resources/containers increases
    d. We endeavor here to establish a process & baselines for the more isolated tests (1a in the agenda) so that we can make progress on "real-world"-type tests (1b in the agenda)
        i. How should we treat other axes?
            1. authorization
            2. transactions
            3. concurrency
            4. versioning
        ii. Initial questions which tests should answer
            1. How does performance change as the size of the file increases?
            2. How does performance change as the number of files increases?
            3. How does performance change as the number of objects increases?
            4. How does performance change as the number of mixed resources increases?
            *Note: In all of these cases, "performance" will be measured by requesting CRUD operations after every x-number of ingest events.*
        iii. **Decision: Defer at first, then examine later once the process & baselines are clear.**
        iv. **Decision: Process should include writing a number of objects/files into the repository (testing the speed of the writes)**
            1. **Every so often (# of writes), test a suite of operations (gets, deletes) to see how the speed of those change**
            2. **That way we test reading, writing, and a number of other operations as overall repository size increases.**