# Motivation

- Goal: Create and document a black box e-mail availability benchmark
- Improving dependability requires that we quantify the ROC-related metrics
- Benchmarks have helped to define metrics
- Problem: Benchmarks are generally performance oriented
- So *performance* metrics are generally well understood, but not *dependability* metrics
- So what are the "good" dependability metrics?

# Anderson's Categories

- **Eric Anderson identifies eight categories as the main axes for evaluating work on systems administration**
  1. Dependability
  2. Automation
  3. Scalability
  4. Flexibility
  5. Notification
  6. Schedulability
  7. Transparency
  8. Simplicity

# Simplified Categories

- **Anderson's categories are important to ROC since the SysAdmin is often the primary recovery mechanism**
- **Problem: Anderson's categories aren't orthogonal**
  - It isn't clear how to differentiate between them in experimental measurements
- **Solution: Divide the categories into three broader categories**
  - Dependability
  - Scalability
  - Human Impact/Productivity

# Target Environment: E-mail

- E-mail today is a mission critical service
  - often the critical service for many companies
- Users expect 24/7 availability of e-mail
- However: E-mail designed to be a “best effort” system
- Dependability metric neglected in most e-mail software and benchmarks today
- Gap between user expectations and systems reality results in…
- Great chance at Making A Difference in real world systems!

# Scoping the Problem

- **Focus is on measuring the dependability of the e-mail service**
  - We want to focus on end user reliability, so we look at overall e-mail service rather than just a server
  - A service can comprise multiple servers in a cluster or just single server
  - We treat the service as a sink
    - » E-mail is delivered *to* not relayed *through* the service
    - » Emphasis on *store* of "store & forward"

# Tentative Benchmark Structure

- **Want to follow basic idea of previous availability/maintainability benchmarks**
  1. Apply workload
  2. Perturb system with faults and human-driven pre-specified maintenance tasks
  3. Ramp workload to measure scalability
- **Treat e-mail system as a black box for generality**

# Potential Metrics

- **Dependability Metrics**
  - Fault-free performance
  - Performance under failure scenarios
  - Delivery delays and errors
  - Dropped/corrupted mail
- **Scalability Metrics**
  - Changes in performance metrics as workload is increased or system configuration is modified
- **Human Impact Metrics**
  - Amount of time operator spends with system to repair and maintain system
  - Human failure rates (fatal and non-fatal)
  - Qualitative assessment by participants of task complexity and system forgivingness

# Metric Measurement

- **Dependability can be measured using a variety of scenarios:**
  - Fault-free, during failure(s), during recovery, during failure + overload, etc.
- **System Perturbation Techniques**
  - Fault injection
    - » hardware, system-level, network-level, etc.
  - Overload
  - Configuration Management (Humans!)
    - » Move a mailbox, add server to cluster, install mail filter, etc.

# Plans and Challenges

- **Plans**
  - Build a heavily-instrumented workload generator with parameterizable workload
    - Start with SPECmail benchmark and expand to cover more scenarios?
  - Start experiments with iPlanet e-mail server
- **Challenges**
  - Developing an accurate and flexible workload generator
  - Extracting useful measurements while treating e-mail service as black box
  - Developing a realistic failure model
  - Creating appropriate tasks for human admins to perform
  - Dealing with human variability

# Benchmarking E-mail Dependability

Aaron Brown (abrown@cs.berkeley.edu) &
Leonard Chung (leonardc@uclink4.berkeley.edu)