SpringFS: Bridging Agility and Performance in Elastic Distributed Storage
Lianghong Xu, James Cipar, Elie Krevat, Alexey Tumanov, Nitin Gupta, Greg Ganger, Michael Kozuch* (CMU, *Intel Labs)

MOTIVATION
- Cloud storage can and should be elastic
  - Ability to extract/re-integrate servers on demand
  - Elasticity is most useful when it is “agile”
  - Agility: quickness of elastic resizing
  - Value: machine-hour (money) savings
  - Challenge: Data migration is expensive
- Agile resizing → 50% less machine hour usage
- State-of-the-art elastic storage designs
  - Sierra and Rabbit force painful tradeoff between elasticity, performance and agility
- Need a new elastic storage design that
  - Fills the gap in the tradeoff space
  - Achieves great agility
  - Maintains performance and elasticity goals

SPRINGFS DATA LAYOUT
- Continuum between “Rabbit” and “Sierra”
  - Elasticity of Rabbit
  - Peak write performance of Sierra
  - Maximized agility along continuum between best cases

MOTIVATION
- Agile resizing → 50% less machine hour usage
- State-of-the-art elastic storage designs
  - Sierra and Rabbit force painful tradeoff between elasticity, performance and agility
- Need a new elastic storage design that
  - Fills the gap in the tradeoff space
  - Achieves great agility
  - Maintains performance and elasticity goals

RESULTS WITH INDUSTRIAL TRACES
- SpringFS achieves “close-to-ideal” machine hour usage
- Better than Rabbit when extracting servers
- Better than Sierra when re-integrating servers

SPRINGFS PERFORMANCE & CLEANUP WORK
30 nodes, each with a 2GB file, 128MB block size

- Machine hour usage: 6-120% improvement
- Data migration: 9-208X improvement

SPRINGFS PERFORMANCE & CLEANUP WORK
- Machine hour usage: 6-120% improvement
- Data migration: 9-208X improvement