Visualizing the Allocation and Death of Objects

Raoul L. Veroy, Nathan P. Ricci, Samuel Z. Guyer
Redline Research Group
Tufts University
Death of Objects

- In a garbage collected language, where do objects die?
- Relate death to allocation
- Understanding death behavior improves program comprehension
- Check our design assumptions
- Find memory leaks
- Solution: visualization
What are we visualizing?

• Use Elephant Tracks\(^1\) to generate a detailed trace of allocations/reclamations for Java programs
• ET computes when object becomes unreachable

\(^1\) [http://www.cs.tufts.edu/research/redline/elephantTracks/](http://www.cs.tufts.edu/research/redline/elephantTracks/)
What are we visualizing?

- Use **Elephant Tracks** to generate a detailed trace of allocations/reclamations for Java programs
- ET computes when object becomes unreachable
What are we visualizing?

- Use **Elephant Tracks** to generate a detailed trace of allocations/reclamations for Java programs
- ET computes when object becomes unreachable
- Lots of objects! Rendering all is challenging.
Hive plot

- Hive plots - rational approach to visualizing networks

Hive plot future work

- Hairball
  - Need edge bundling
- Address scalability
- Arrangement along axis
  - Using context, but which context?
  - Anything else?
Matrix plot

- Suggested by reviewer
- Uses time as context
- Column – allocation time
- Row – death time
Allocation time

Death time
Allocation time

Death time
Matrix plot future work

- Lower triangular matrix by definition
  - Eliminate wasted space?
- Need better rendering techniques
- Need clustering
  - Automatic
  - Interactive
Thank you.
Questions, suggestions and comments?
rveroy@cs.tufts.edu