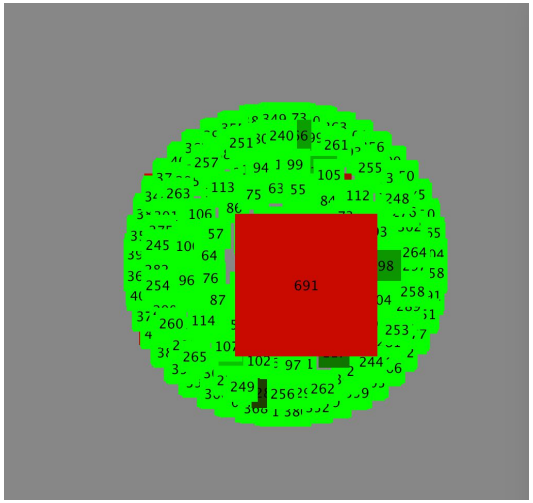
01 - Visual Process Manager

Dan Bye, Dan Brill, & James Draper

Project Proposal

- The Visual Process Manager is an interactive process manager like top, but with a graphical interface where processes are visualized as nodes in an undirected graph
- Each node is a button which is dynamically created from each process currently running by polling a periodically updated database
- The user can manage their processes by selecting a node in the graph and then further selecting a software interrupt to send to that process

Details



- The button is labeled by the Process I.D.
- The size of the button is based on ram usage.
- The color of the button represents the CPU usage of each process.
  - This ranges from green (low usage) to red (high usage).
- The locality of each button is based on the scaled size of the largest node, pushed out radially with an increasing number of nodes at each layer

Design Pattern: Flyweight

- Because we're generating so many nodes for process it would be viable to architect a class like NodeProc that refactors it's major methods by way of extraction and puts them into a class like GraphNode

- This way we can instantiate many NodeProc objects without putting an overly large load on our system

Design Pattern: Observer

- We could’ve used the observer pattern to monitor the database
- Each time the database changed the observer could

  invoke our graph class

  and re-render the visualization,
- Right now, both are timed updates.

Use Cases and Requirements

- US-01: Able to review processes visually.
- US-02: Be able to manage processes by interacting with the visualization.
- Database Implementation
  - FR-06: Process Data can be stored in a Database
  - NF-05: Processes updated periodically

Demo