2014–2015 HMC Computer Science Clinic SpaceX Project

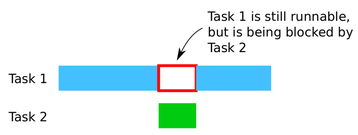
Process Time Analysis in Spaceflight

Problem Statement

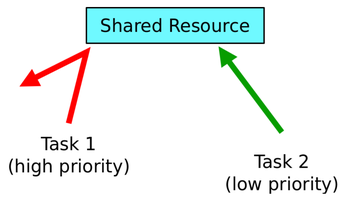
Our team will make it **easier** and **faster** for SpaceX engineers to find software problems by creating a tool that **visualizes** rocket data and highlights problem areas. We will create an **improved user experience** and provide more visualizations of the data than the current tool, KernelShark.

SpaceX, Rockets, and Software

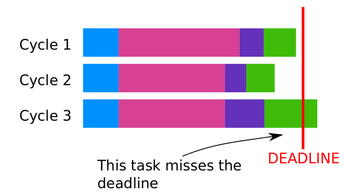
SpaceX rockets contain computers that run flight software. Using data captured of a schedule of the tasks that these computers were running, developers look for problems like those below:



Preemptions: Tasks 1 and 2 are running on the same CPU. When Task 2 runs, Task 1 is blocked by it, and must wait for Task 2 to stop until it can run again.

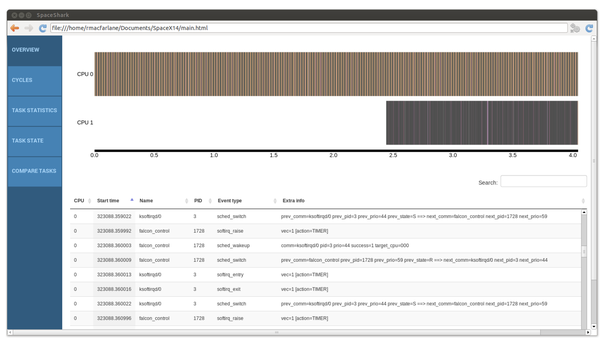


Priority Inversions: Task 2 is using a shared resource that Task 1 needs access to, so Task 1 is blocked from running until it can access that resource, even though it is higher priority than Task 2.



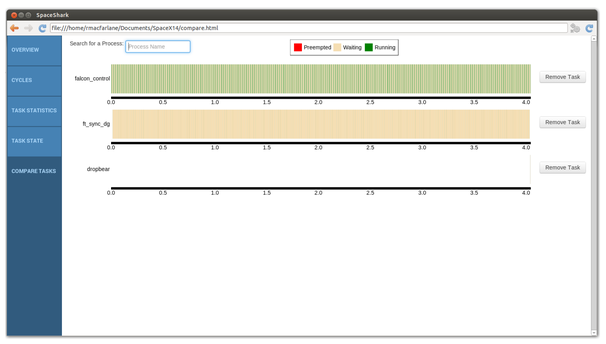
Cycles: Tasks run cyclically on the computer, with a deadline at the end of each cycle. The pattern of tasks generally looks similar between different cycles.

Visualizing a Rocket's Software



This page displays an overview of the data. The top chart is a timeline of running tasks, each with a unique color. Users can hover over events on the chart, zoom, pan, and click to jump to the event in the tabular view of the data below.

Tasks run repeatedly in a fixed cycle on SpaceX rockets. This page breaks the timeline up into cycles and stacks them so that users can easily compare across them.



To help users see interactions between tasks, this page displays multiple task state graphs at once. Users can add, remove, and drag to reorganize these graphs.

Building Our Visualizer

Architecture

Our application has three main portions. The raw input, the parser and its output, and the web-based desktop application. The sections are modular, so as long as output/inputs remain the same, internal changes to one section will not affect the other at all.

Rocket Data — *ftrace*

```
trace-cmd-4442  [002]
1681.955167: sched_switch:
   prev_comm=trace-cmd prev_pid=4442
prev_prio=120 prev_state=S ==>
next_comm=swapper/2 next_pid=0 next_prio=120
python-2418  [001]  1681.955177:
sched_stat_wait:
   comm=trace-cmd pid=4441 delay=101985 [ns]
```

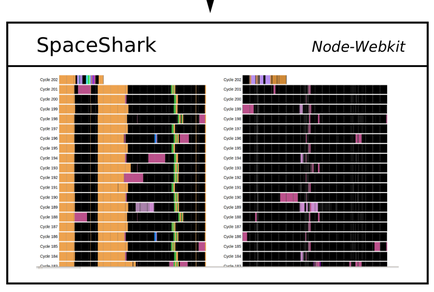
parse — *Java*

Human Readable Format — *JSON*

```
"events":[
   {
    "startTime":83067,
    "name":"trace-cmd",
    "cpu":"[001]",
    "pid":3939,
    "eventType":"sched_wakeup"
   }    ...
```

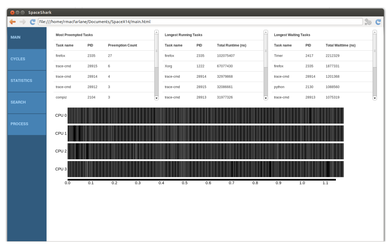
visualize — *d3.js, DataTables.js, IndexedDB*

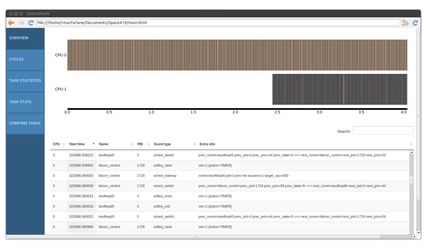
SpaceShark — *Node-Webkit*



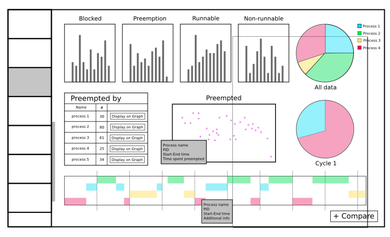
User Testing

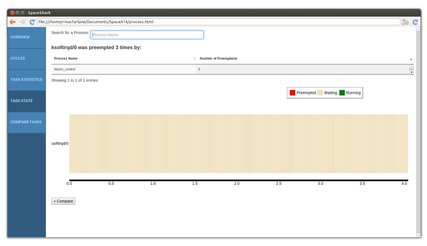
User testing played a major component in making sure that we were creating a tool that SpaceX engineers would find useful to find problems within their rocket software.



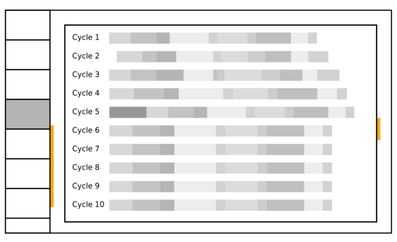


Our previous Overview page featured statistics at the top and a graphical representation of the events. Through user testing we found that the statistics did not prove useful to SpaceX engineers. SpaceX engineers are more interested in having a tabular version of the events list.





There was no view in Kernelshark that looked at particular task. The left image shows our mock-up. We later refined this design to include more focused statistics like preemptions, which are important to SpaceX engineers.



The purpose of the Cycles page was to be able to quickly find anomalies within the rocket software, which didn't exist in Kernelshark. This page was one example of positive user testing, as there were minimal changes between our initial mockup and the current tool now.

Acknowledgments

Thanks to our sponsor, SpaceX, for technical and equipment support. Thanks also to the Clinic staff, who made this project possible.

Team Members

Wendy Brooks (Fall PM)
May Lynn Forssen (Spring PM)
Alix Joe
Rachel Macfarlane

Advisor

Prof. Ben Wiedermann

Liaisons

Jesse Keller, Jim Gruen, Jessica Hester ('13)