Lessons Learned from Developing and Operating the Kepler Science Pipeline and Building the TESS Science Pipeline

Jon M. Jenkins
NASA Ames Research Center

Monday September 4, 2017

WP11 Exoplanet Science Meeting
University of Warwick
Coventry UK
Overview

- What did it take to build the *Kepler* science pipeline?
- Major modifications to pipeline over lifetime
- High fidelity simulations
- Commissioning, commissioning, commissioning
- High performance computing
- Developing the TESS Science Pipeline
- Communication
- Summary
• Design started in earnest in 2004 with launch in March 2009 and operations through May 2013 and reprocessing through 2017
• A total of ~100 person years of effort went into the first complete version of the pipeline (from pixels to planets)
• The staffing was at ~20 individuals per year through 2016, tapering off thereafter (~280 FTEs over project lifetime)
• Build 5.0 was the launch-ready software release
• There were 4 major builds thereafter, with substantive point releases to mitigate issues subsequently identified in flight or full volume re-processing
• Build 9.0, 9.1, 9.2, 9.3 really represented at least two full builds of effort (issues identified in full re-processing and in completeness and reliability processing)
• Unexpected instrumental effects/stellar variability/hardware failures motivated significant software modifications on orbit
Kepler’s Science Pipeline

- **Artificial Transit & BEB Injection Machine**
- **CAL**
  - Pixel level Calibration
- **PA**
  - Photometric Analysis
- **PDC**
  - Pre-search Data Conditioning
- **TPS**
  - Transiting Planet Search
- **DV**
  - Data Validation
- **TCERT**
  - Threshold Crossing Event Review Team + Robovetting

**Raw Data**

- **Raw Light Curves & Centroids**
- **Corrected Light Curves**

**Diagnostic Metrics & Reports**

- **Threshold Crossing Events (TCEs)**

**Auto-Vetting**

- Applying machine learning to candidate evaluation

>1,000,000 Lines of Code; 26 different Modules
Every component of the science pipeline saw major evolution over mission

Pixel level calibrations:
- Updates based on actual electronics behavior
- Flagging of electronic image artifacts causing false positives

Identifying optimal apertures
- Use of reconstructed pointing
- Added ability to correct errors in Kepler Input catalog

Photometric analysis
- Major improvements to identifying cosmic rays

Pre-search Data Conditioning
- Development of Maximum a Posteriori approach
- Addition of multi-scale analysis
- Detection of Sudden Pixel Sensitivity Dropouts

Transiting Planet Search
- $\chi^2$ vetoes added

Data Validation
- Difference image analysis
- Ghost Diagnostic + other metrics
Kepler is sensitive to its thermal environment

Signature of a heater cycling on the reaction wheels 3/4
Instrumental Effects in Photometry

- A Search for Earth-size Planets

Graph showing instrumental effects in photometry, with high pass filtering applied to the data.
Correcting Systematic Errors

We apply a Maximum A Posteriori approach as per Stumpe et al. 2014
End-End Model (ETEM) drove design of SOC and testing of entire ground segment

Simulated data were so good that we didn’t need to update the compression tables after launch (the achieved compression (~4.5-5 bits per pixel) was within 0.1 bits of ideal performance)
Difference image analysis was key for Kepler for excluding false positives from background eclipsing binaries. Especially important for bright, saturated (bleeding) targets.

KIC 3542116
Commissioning tools require special attention and data sets
• Effort for commissioning tools may be as great as that for major science pipeline modules
• Don’t leave commissioning tool development to the last
Noise is given by for a short time interval for the time of the flight image so that the gate can get a homogeneous distribution which provides exactly 1125 output channel. To achieve this distribution the intersection is close.

This background polynomial is subtracted from the pixel value.

Background targets are selected to create a 2D polynomial representation of an irregular Cartesian mesh. Parallactic range functions (PRFs) are not smeared.

The fraction of each target’s flux that falls in its optimal aperture that is due to the target star. The result is useful for identifying uncrowded stars.

$C_{\text{target}} = \sqrt{\sum b^2}$

where $b$ is the sky crowding metric, $C_{\text{target}}$ is useful for estimating the dilution impact on the detectability of transits.

Measure on a 21 pixel square provides a presentation of the background on each channel.

$\nu$ is the value for this channel from the read noise and $\nu_{\text{quant}}$ is the quantization noise. Quantization noise formula (1) includes Poisson shot noise of $w = \text{well depth}$. For purposes of comparison with the sky crowding metric $n$ is the value for this channel from the read noise and $n_{\text{quant}}$ is the quantization noise.
Keeping Up with the Data
Improving the Throughput

Some fast code; Some slow code

Step 1: Parallelize all code

Step 2: Make slow code fast(er)
Hardware Architecture: 
*Kepler Science Operations Center*

- 64 hosts, 712 CPUs,
- 3.7 TB of RAM,
- 148 TB of raw disk storage
A Search for Earth-size Planets

Hardware Architecture:
NAS Pleiades Supercomputer

5.34 Pflop/s peak cluster
211,872 cores
724 TB of memory
15 PB of storage
Characterizing completeness and reliability of software/people pipelines is extremely resource intensive
Kepler shipped the final light curve products in April 2015
We’ve spent the remainder of the time until present adding artificial transits, BEBs, scrambling the data temporally, inverting the light curves etc., etc.

Mapping completeness and reliability and characterizing the candidate vetting process is difficult

Recommendation: Pursue machine learning for conducting or modeling the candidate vetting process

>1,000,000 Lines of Code; 26 different Modules
Developing the TESS Pipeline

- ~13X pixel data rate over Kepler
- Leveraged heritage from Kepler pipeline
- Significantly lower cost (~46 FTEs over project lifetime)
- Significant speed improvements:
  - Colocated servers and storage with NAS Pleiades supercomputer
  - Moved pixel-level calibrations to C++
  - Sped up Presearch Data Conditioning by 10X
  - Originally projected 20+ days to process one sector
  - Complete pipeline requires ~5 days to process one sector
The TESS Project is distributed geographically with the Science Pipeline separated by a continent from the Science Office.

Resolving data issues requires good communication between the Payload Operations Center, the Science Processing Operations Center and the Science Office.
New ideas for improving photometry/astrometry will emerge, both within the team and without

- “Halo” photometry on K2 data on Pleiades (White et al. 2017, MNRAS 471)

Preserving ability to re-process the pixel data with better algorithms and tuned parameters is a really good idea

Take advantage of the compressibility of your data
- *Kepler* achieved compression rates of 4.5 bits per pixel
- TESS should achieve compression rates of ~3 bits per pixel for 2 minute data and ~4 bits per pixel for 30 min FFIs
Summary

- Science pipelines require significant planning and effort
- Previous pipelines can be leveraged to reduce development time (but this does not reduce time required for V&V testing)
- Plan to rewrite the majority of the science code in light of unexpected in-flight characteristics/behavior/hardware changes
- High fidelity simulations are indispensable
- Determining $\eta_{\text{earth}}$ is computationally intensive and huge effort
- Give adequate attention to developing commissioning scenarios and associated tools
- Take advantage of data compression to increase the amount of pixel data downlinked from PLATO
SOC Cluster Architecture

6 Clusters:
4 Operations Clusters: Flight Ops, Quarterly, Monthly & Archive)
2 Test Clusters: LAB & TEST

Science Processing Pipelines

Long Cadence Photometry Pipeline

```
DYN -> CAL -> TAD -> PA -> PDC -> PAD -> PMD -> PAG
```

Transit Search Pipeline

```
TPS -> DV -> CA
```

FFI Pipeline

```
CAL -> PA
```

Short Cadence Pipeline

```
CAL -> PA -> PDC
```