

# Tellus 2.0 Specifications

Monday, September 11, 2017 7:08 AM

## [Concept Draft]

### Architecture Changes

- ☐ 1. New job scheduler, with job queuing and priority features [RPD:34796867](#)
  - a. Chronos or Kubernetes
  - b. Replaces Tellus Scheduler, JobQueue, and STAF Cron
  - c. For SGE only: Tests will directly run on Tellus server (without STAF/STAX)
  - d. For FPE only: Job will call STAF STAX to execute test on FPE VMNotes:
  - i. Mesos/Kubernetes for Windows could replace STAX, but it is not ready yet
  - ii. Docker for Windows can potentially replace FPE, but it is not ready yet; and this option is not in the plans
- ☐ 2. Tellus scaling and load balancing (multiple instances of Tellus)
  - a. Mesos or Kubernetes platform
  - b. [RPD:31813040](#) Tellus load balancing
  - ☒ c. [RPD:34621075](#) Evaluate Kubernetes vs Mesos => completed preliminary eval => they are about equal, each with strength/weakness
- ☐ 3. Tellus API - Tellus server side will be Web Services only ("Tellus as a Service")
- ☐ 4. Complete separation between browser and Web server tiers  
(Browser communicates with Web server only through Tellus API)

### Browser Side Changes

- ☐ 1. Single Page Application (SPA) -- i.e. switch to Angular 4
  - a. Web application works more like desktop application
  - b. Merge JEP, Calendar, Job List into one single application (with multiple views)
  - c. But still separate Results page -- linkable from both Tellus and QAI
  - d. But still separate Progress page: Users might open multiple Progress pages to monitor multiple jobs in separate windows
  - e. But still separate Admin app
- 2. Optimizations:
  - a. Bundle JavaScript files into one -- faster
    - i. Will be part of new build system (such as angular-cli)
    - ii. Prefer WebPack (via angular-cli, or vue-cli)
  - ☐ b. [RPD:34658408](#) Merge browser-side configuration files

### Browser Side Development Workflow Changes

- ☐ 1. New Web JavaScript Framework
  - a. [RPD:35677222](#) Web UI Framework Evaluation - Angular 4 vs. React vs. Vue.js
  - b. Note: Engineering and Tellus Job List use AngularJS v1
  - c. Standards: AngularJS v1 is FactSet standard, but not Angular 2 or 4.  
Choices:
    - i. [Angular 4](#)
    - ii. [React](#)
    - iii. [Vue.js](#)
  - d. Must be able to embed component in QAI and Tellus
- 2. New Web programming language

- a. TypeScript, only if Angular 4
  - b. JSX, only if React
  - c. ES2015, only if VueJS
- ☐ 3. New build system
  - a. If Angular 4, use angular-cli ([RPD:34626945](#))
    - i. Includes typescript, tslint, jasmine, karma, protractor, istanbul, [codelyzer](#), node-sass, sourcemap, webpack, minifying, uglify.
    - ii. Generates new project template
    - iii. Includes build tasks: build, lint, test, serve (embedded Web server)
  - b. If React: create-react-app
  - c. If Vue.js: vue-cli
  - ☐ d. Gulp (only for any additional build tasks beyond tools like angular-cli)
- ☐ 4. New package manager: Yarn
  - a. Yarn handles concurrency well (better than npm)
  - b. Yarn is FactSet Standard: [Package Manager Standard Recommendation](#)
- ☐ 5. Mono-repo
  - ☐ a. [RPD:35639920](#) Evaluate mono-repo solution
  - b. Multiple sub-packages but one single Git repo; each sub-package has own package.json
  - c. Job Entry Form in separate sub-package (QAI can install it as separate package)
  - d. Choices:
    - i. [OAO](#)
    - ii. [Yarn Workspaces](#)
    - iii. [Lerna](#)
- ☐ 6. Separation of application logic from Web framework code
  - a. Put application logic in separate file, away from UI framework code
  - b. => Can switch Web JavaScript framework more easily
  - c. Note: But the design of Angular might make that hard -- some application behavior is embedded in app template code
  - d. Example: form/field validation can definitely be in separate files
- 7. Responsive Design (adapting to window sizes):  
[https://en.wikipedia.org/wiki/Responsive\\_web\\_design](https://en.wikipedia.org/wiki/Responsive_web_design)
- 8. Semantic versioning standard <http://semver.org/>
- 9. Recommended IDEs:
  - a. Browser Side:
    - i. Visual Studio Code (for TypeScript)
    - ii. IntelliJ IDEA (for ES2015)
    - iii. Eclipse?
  - b. Java (Server Side):
    - i. IntelliJ IDEA for Java EE
    - ii. Eclipse
  - c. PHP (Server Side):
    - i. PhpStorm or IntelliJ Ultimate Edition

## Server Side Changes

- ☐ 1. New job scheduler -- Chronos or Kubernetes
  - ☐ a. [Nice to have] Show resource availability in UI (helps with why jobs are still waiting)
  - ☐ b. [Nice to have] Calculate estimated start time while job is waiting in queue
- ☐ 2. [RPD:18044118](#) Tellus upgrade to PHP 7
- ☐ 3. [RPD:35880911](#) Evaluate Tellus API switch to Java or Go
- ☐ 4. **Java-based Tellus API???**
  - ☐ a. **Live in TomEE or standalone microservice???**
- ☐ 5. PHP Web service framework
  - ☐ a. [RPD:35872618](#) Evaluate PHP framework for Tellus API

- ☐ b. [RPD:23938484](#) Production Tellus API to use PHP Web service framework
- ☐ 6. PackageName (of test script) should be at test plan level? (currently at test case level)
- ☐ 7. Move test case management to QAI? Else QAI will need to notify Tellus when test case move to different test plan
- ☐ 8. DB Optimization -- SQL Server seems to be getting slower in high volume, with so many table joins and rows
  - ☐ a. [RPD:34343484](#) database speed optimization
  - b. ~~Evaluate fast hierarchical DB~~ -- post Tellus 2.0
- 9. Report Database ("data warehouse")
  - a. Metrics team can query the report database
  - b. Write to report database -- daily
  - c. Report table format is more suited for reports than OLTP
  - d. Consolidate/rework current results tables and metrics tables?  
(To address issue: They are spread out over several tables and they cannot be joined for getting test results and metrics in one single query.)
  - e. Include Chronos/Mesos/Kubernetes stats???
- 10. Limit Fixed Time jobs: there are too many running
- 11. **Move away from mod\_wsgi (Python Web)???** Reasons:
  - a. Last update was 2 years ago (2015)
  - b. Errors are all "500 Internal Server Error" -- no explanations -- which makes debugging difficult
  - c. Every new script needs Apache configuration, for example:  
WSGIScriptAlias /myapp "C:/Tellus/cgi-bin/Tellus\_p1.py"
- 12. [RPD:36534612](#) Tellus upgrade to Python 3
- 13. [RPD:35918764](#) Tellus upgrade to Java 9
- 14. STAF upgrade - current version is too old and it uses a very old version of IBM Java

## Continuous Integration & Delivery (CI & D)

- ☐ 1. [RPD:25471794](#) Jenkins for deployment to tellus server
- 2. Docker-based Tellus deployment -- need Mesos or Kubernetes -- Post Tellus 2.0?

## Other RPDs

- ☒ 1. [RPD:29543242](#) Tellus frontend move to GIT
- ☒ 2. [RPD:33113283](#) Tellus merge DEV and production versions
- 3. [RPD:31247818](#) Add Tellus API Status retrieval to the Tellus Results Page
- 4. [RPD:31247826](#) Add Tellus API Status retrieval to the Tellus Progress Page
- 5. [RPD:31247838](#) Add Tellus API Status retrieval to the Tellus Calendar Page
- 6. [RPD:36142339](#) Tellus should consolidate BrowserType/SGBrowserName, BrowserVersion/SGBrowserVersion
- 7. [RPD:36179882](#) Tellus PackageName, ProjectType, SCM, etc. should be at test plan level

## Deprecations

[Tellus 2.0 List of deprecations](#)

# Tellus 2.0 Deprecations List

Tuesday, September 26, 2017 11:23 AM

## Tellus 2.0 List of Deprecations

[WORK IN PROGRESS]

1. JEP - Needed
2. Calendar – Needed
  - a. Future recurrences can be updated
  - b. Ad-hoc jobs
  - c. End user prefers information on calendar when we point out a problem in an RPD
  - d. Should allow filter by user for relevance to specific user, and reduce size
3. **Job list Page – Can be retired**
  - a. The online recurrence jobs generated by swp-green should be displayed on calendar if job list page has to be retired
  - b. And this the jobs will not exist with QAI launch automation in play
4. **TTM - should be merged with QAI [we can take a survey]**
  - a. Vasu – NO, we need to enhance TTM to be better instead
  - b. Krishna – YES, If not done we got to enhance TTM
  - c. Avanthi – YES, If not done we got to enhance TTM
  - d. Joseph - YES, test case/script configuration fits naturally in QAI; we usually don't need test script info in reports/job queries, but can call Justifier API for job creation in JEP
5. **CLP/Kibana for TTM – dependence on TTM**
  - a. Alternative: app dynamics
6. Progress page: Needed
  - a. We need to make the UI and data displayed more stable
  - b. We need to display some additional filters and grouping, something similar to QAI Status web page
7. **Tellus List Builder: Deprecated, as per Rajul (replaced by reports DB)**
8. **Swp-green**
  - a. Can be deprecated if QAI-Launch automation is in play and stable
9. **Chunkid results page - Can be retired**
  - a. We need to merge it with the groupid results page
  - b. And also need new reports for detailed errors and screenshots
10. Consider deprecating ancient browsers like IE9 etc.
  - a. Ancient browsers hold back progress and features
    - i. Example: support for HTTP 2.0
    - ii. Example: CSS transition (in IE 10+) helps with more "intuitive" UI (animate state transitions)
  - b. [RPD:34549342](#) Tellus - log actual user browser version usage for browser support
    - i. Support N-1 (one back version)

# Load Balancing Choices

Wednesday, November 08, 2017 10:17 AM

## Mesos/Kubernetes vs. Conventional Load Balancer

Differences between resource-aware load balancing with Mesos or Kubernetes and conventional load balancer are:

1. Conventional load balancer's algorithm are generally of these types: round robin, least connections, source (user), URI, or URL parameters. Note that they do not include CPU or memory utilization. In contrast, Mesos or Kubernetes takes CPU and memory into account.
2. Also, conventional load balancers are not responsible for deploying Web servers or applications. But Mesos and Kubernetes can deploy Web servers, services, or applications.

**Note:** Mesos does not have load balancer by itself, but needs to work with Marathon-lb and HAProxy for load balancing. Kubernetes has built-in load balancer.

## Load Balancing with DNS

[DNS](#) has the ability to load balance a domain over multiple servers. When performing traffic load balancing with DNS, you don't have any control over the balancing algorithm. It always uses [round robin](#) to dedicate a server it has listed for a given record.

Also, DNS caching can be very limiting. If a server becomes unavailable in the listed pool, DNS still provides the IP address to clients that are sending the requests. This can continue for hours until that DNS record is removed and all instances of that record expire from every DNS servers cache.

## Commercial Load Balancing Software

<https://www.capterra.com/load-balancing-software/>

## Hardware Load Balancers

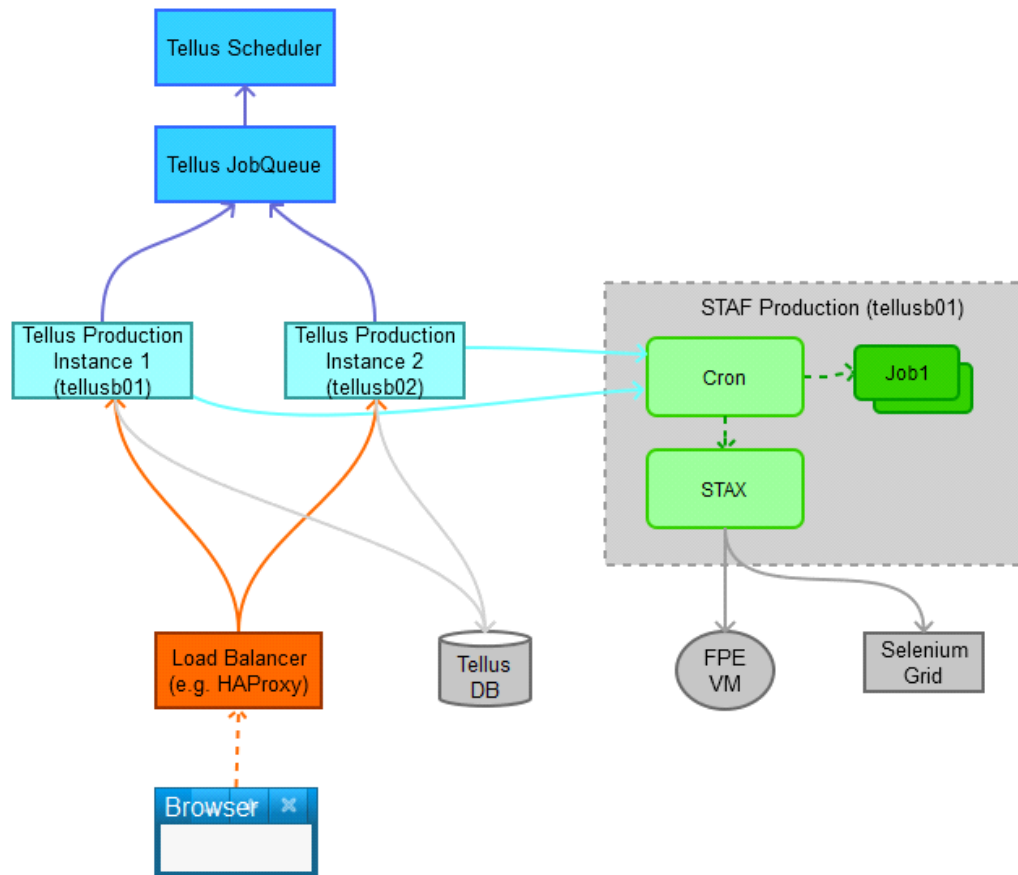
Very expensive.

## Reference

- <https://blog.stackpath.com/load-balancing-haproxy>

# Figure 1. Tellus Load Balancing Architecture With STAF Cron

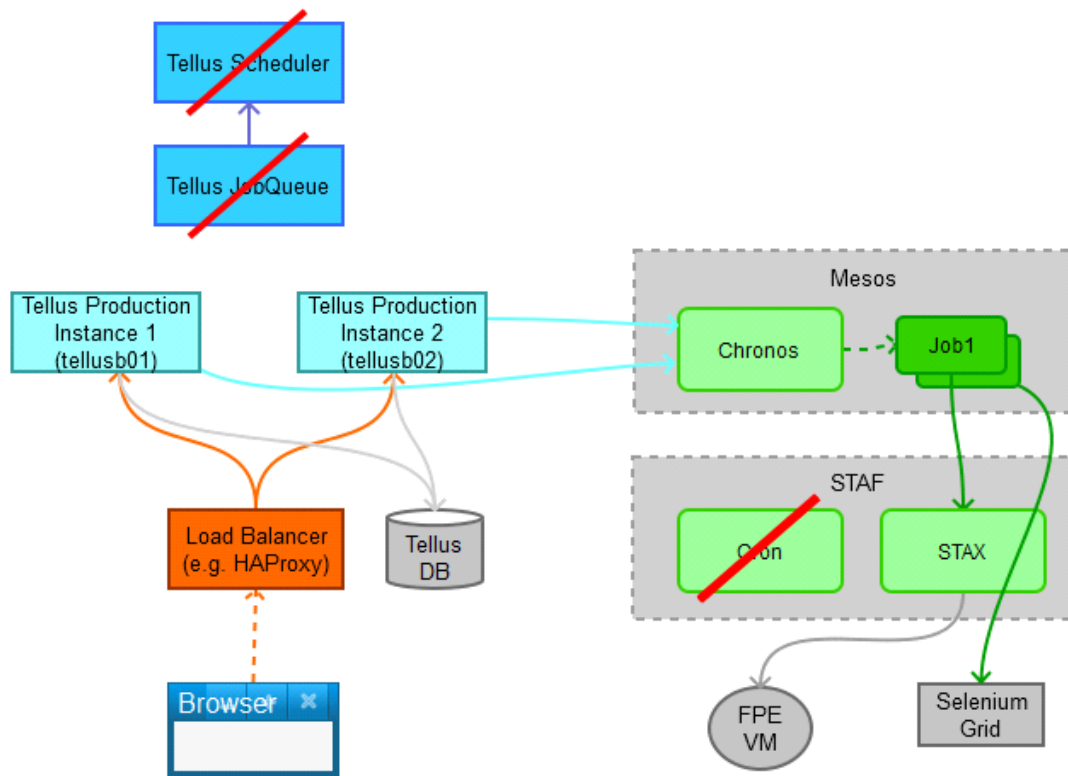
Friday, November 10, 2017 7:48 AM



**Figure 1.** Tellus conventional load balancing architecture with STAF Cron.

## Figure 2. Tellus Conventional Load Balancing Architecture With Chronos

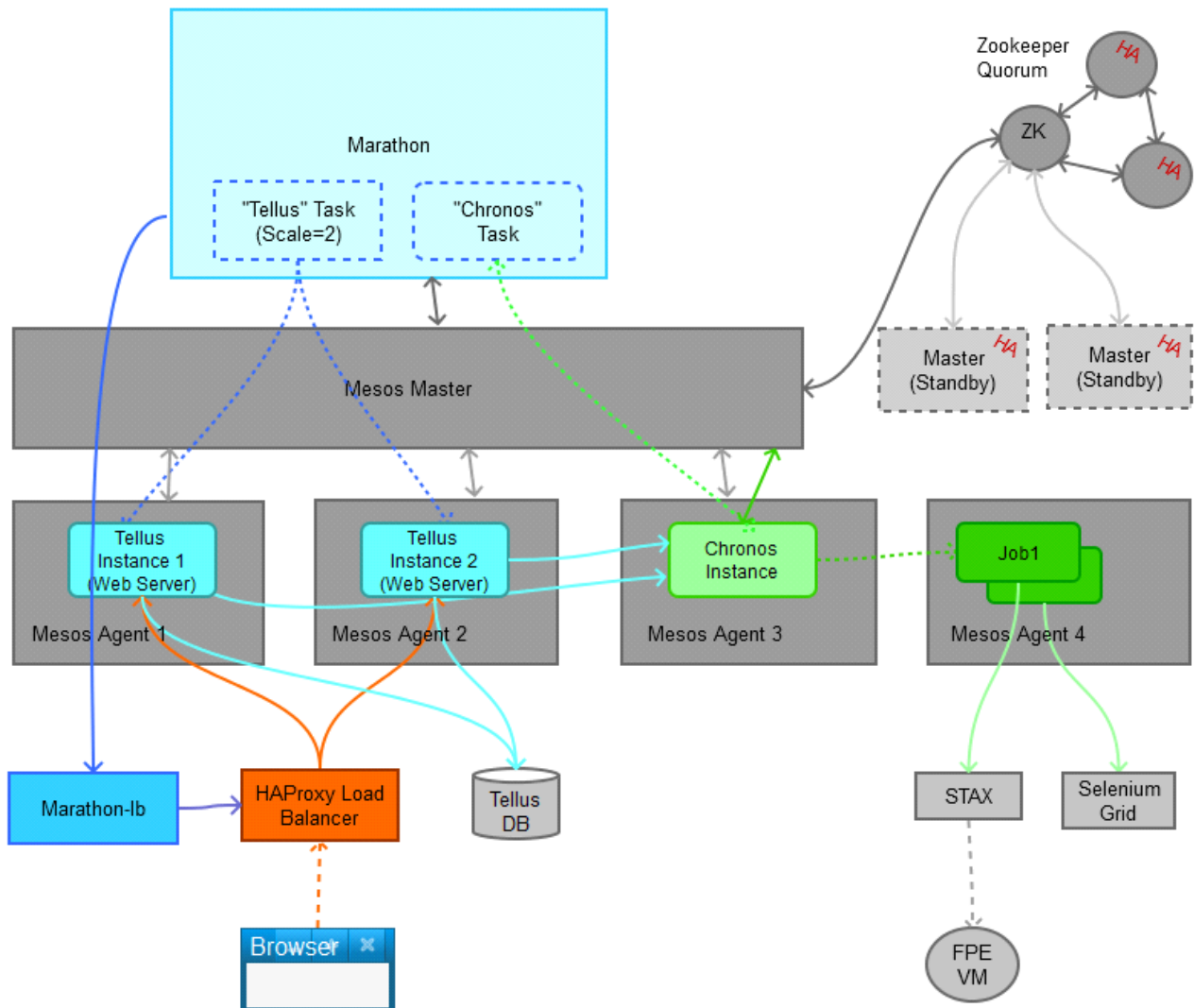
Friday, November 10, 2017 7:49 AM



**Figure 2.** Tellus conventional load balancing architecture with Apache Chronos.

# Figure 3. Tellus Mesos Load Balancing Architecture With Chronos

Friday, November 10, 2017 7:50 AM



**Figure 3.** Tellus load balancing architecture with Apache Chronos and Mesos. The optional "HA" components are only for High Availability (HA).



# Tellus Stability Issues

Wednesday, January 17, 2018 3:18 PM

## Tellus Stability Issues

1. **High CPU and memory utilization** on Tellus server
  - a. **Possible Solutions:**
    - i. [RPD:36380734](#) Tellus scaling (see below)
    - ii. [QAF-523](#) download test suite directly to VM for FPE jobs -- KrishnaTeja
2. **NPM install takes several minutes (> 7)**
  - a. **Possible Solutions:**
    - i. [RPD:34728138](#) Use yarn to download node\_modules on Tellus -- KrishnaTeja
3. [RPD:36380734](#) **Master Tellus Application Scale -- Avanthi, KrishnaTeja, Joseph**
  - a. **Phase 1:** Replace tellusb01 by tellusb03 (with more CPU cores and memory)
    - i. **Status:** In Test. Will release soon.
  - 1) **Phase 2:** [QAF-524](#) download test suite directly to a Client server for SGE jobs -- KrishnaTeja
    - i. **Status:** In Test. Will release soon.
  - 2) **Phase 3:** Load Balancer
    - i. **Status:** POC
4. [RPD:30045825](#) **Tellus FocalPoint error: ran out of VMs -- Avanthi**
  - a. **Main Cause:** Tellus Scheduler design issue: *dependence on duration estimates*  
Tellus Scheduler uses estimated job durations to layout job schedules. So, if a job runs longer than expected, the next job might content with that resource that is still held (e.g. FPE VM). This makes the Scheduler problem more obvious: [RPD:31219100](#) jobs running much longer (see below)
    - i. **Solution:** [RPD:34796867](#) replace job scheduler (see below)
  - b. Another cause is [RPD:33185848](#) hung jobs (see below)
  - c. Another cause: STAF "dispatcher" (job script) could not catch all exceptions, so didn't free the allocated VM
    - i. **Solution:** [RPD:25471809](#) Tellus STAF dispatcher catch all exceptions
  - d. Another possible cause is VM allocation race condition (involving ip.txt)? -- Avanthi  
(Tellus thinks there is VM allocation timed out, because it missed the allocation response, due to race condition?)
5. [QAF-465](#)/[RPD:34796867](#) **Tellus switch to new job scheduler -- Joseph**
  - 1) **Status:** Waiting on Infrastructure Team (Kyong Kim) to provide Kubernetes.
  - 2) **Description:** Replace the Tellus Scheduler and STAF Cron to solve Tellus Scheduler design issue. Prospective new job scheduler: Apache Chronos or Kubernetes. Infrastructure Team (Kyong Kim) prefers Kubernetes. QA Automation is okay with either Kubernetes or Apache Chronos.
  - 3) **Required Features:** Priority, Job Queue, Custom Resource Sharing
6. [RPD:34666075](#) **Tellus FocalPoint VM timeout -- Avanthi**
  - 1) **Status:** Needs investigation -- Avanthi will take a look when she's done with scaling [RPD:36380734](#)
7. [RPD:33185848](#) **Tellus jobs are hung -- was Vasu, now Avanthi**
  - 1) **Status:** Needs more investigation. Though this RPD is resolved, it might still re-occur.
  - a. **Possible causes:**
    - i. Unexpected pop-up dialog (e.g. login prompt, error message)
      - 1) Possible Solution: [RPD:31133602](#) Tellus hung job detection utility
    - ii. Test script waiting for an element for a very long time
      - 1) Possible Solution: Need to adjust timeout values?

8. [RPD:31219100](#) Tellus jobs running much longer than before -- was Vasu, now Avanthi
- a. **Status:** Need more investigation.
  - b. e.g. BVT ran for 1-3 days
  - c. Possible scripting issue (e.g. script not exiting properly)?
  - d. Some cases might be normal

# Tellus 2.0 Backend features list

Wednesday, July 26, 2017 3:19 PM

API : Improvements requested from last 1 year.

Marked in **green** are the fields we care about

Row Labels	Count of RPD Link
Ability to Schedule job from API	1
Consolidation of API	1
Environment line up - NA	1
Error Handling	2
External Application issue NA	1
Manual Code/file/dependencies moves	1
NA	23
New API - NA	1
New API NA	1
New Feature	2
New Feature - Auth	1
New Feature - Get Status	2
New Field	4
New Fields	7
New Test Environment type	1
Office version config file different from Tellus UI	2
SIT NA	1
Slowness	1
Update duration to a some value when sent 0	1
Use code in Dev and Prod and diff in a new file	1
(blank)	1
<b>Grand Total</b>	<b>56</b>