How to Test the Performance of Exalate for Jira Data Center

Last Modified on 10/17/2023 9:36 am EDT

This article describes the performance testing of the Exalate App in the Jira Data Center.

We've tested the performance and scalability according to testing requirements for Data Center apps in Atlassian Marketplace.

The results provide an understanding of how the Exalate app impacts a Data Center cluster deployment depending on the configuration.

We also compared the performance of Jira with the app and without the app to define differences in system performance and performance degradation.

Testing stages:

- executed tests using Jira DC without Exalate;
- executed tests using Jira DC with Exalate using 1 node Data Center deployment (with warm cache) and Jira Server with Exalate.
- executed tests using Jira DC with Exalate using 2 nodes Data Center deployment (with warm cache) and Jira Server with Exalate.
- executed tests using Jira DC with Exalate using 4 nodes Data Center deployment (with warm cache) and Jira Server with Exalate.

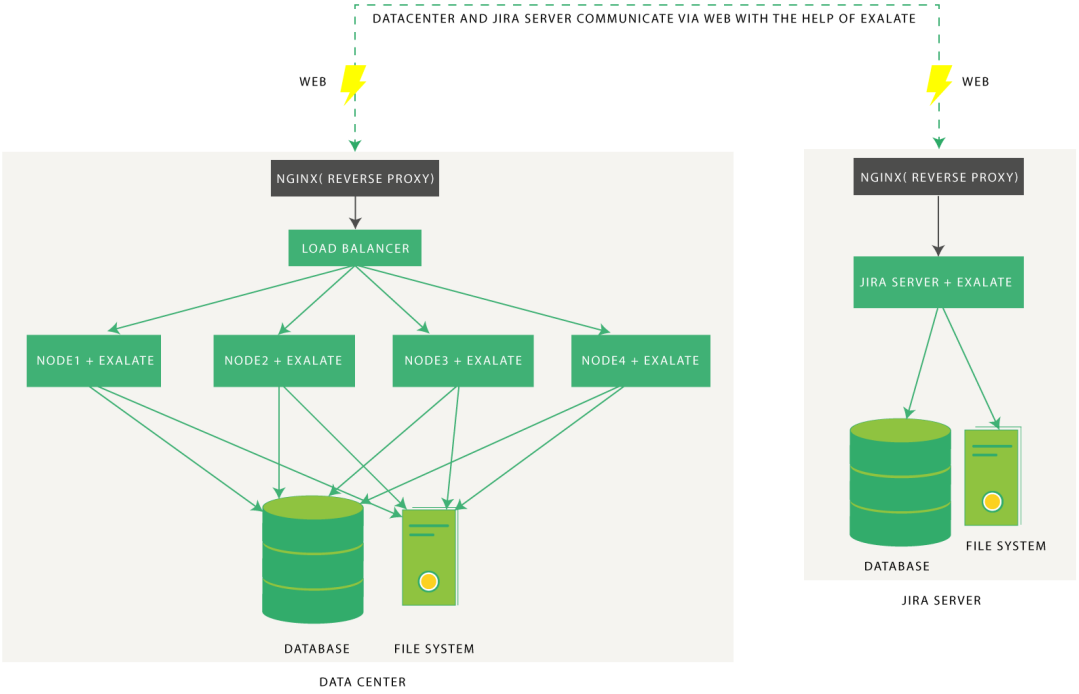
We've used the following methods for performance testing:

- Browser testing (Chrome browser) for performance and scale testing of Jira UI (User Actions) and Exalate UI (User Actions).
  User actions were executed manually and the response time of each request was manually measured via dev tools(Network), collected, and added to the table of results.
- Automated testing via Postman (Desktop version) for Jira and Exalate API performance and scale testing.
  Created collections of API actions. Each action consists of a request with configs and params required for the execution. The collections were executed via automatic collection run.

Environment

	Client-side environment:	**Server-side environment (application)**
Processor	2.5 GHz Intel Core i7	Intel(R) Xeon(R) CPU E3-1231 v3 @ 3.40GHz
Installed memory (RAM)	16 GB	32Gb
System type	macOS High Sierra Version 10.13.6 (17G65)	Ubuntu 16.04.5 LTS - 4.9.124-xxxx-std-ipv6-64

- The Jira Data Center consists of 4 Jira instances with the Exalate app on each node + 1 Postgres 9.4 (all in one local network #1). Apache2 is used as a load balancer.
- The Jira Server consists of 1 Jira instance with the Exalate app + 1 Postgres 9.4 (all in one local network #2).
  Jira Data Center and Jira Server communicate via the web with the help of the Exalate app. We've used Nginx as a reverse proxy on top of both the Data Center and the Jira Server. All communication requests between the Data Center and Jira Server went through Ngnix.
- The Jira Data Center database is deployed on the same physical hosts (machines) as the nodes themselves.
- The Jira Server database is deployed on the same physical hosts (machines) as the nodes themselves.



Dataset

We've populated the Jira Data Center with the following set of data

	Data entity type	quantity
1	Users	4902
2	Projects	1501
3	Issues	1,000,021
4	Custom fields	1408
5	Workflows	452
6	Attachments	659,994
7	Comments	9,903,367

User Actions with the Exalate App for Jira Data Center

We scripted the postman to perform a user action from a list of available actions and immediately move on the to next action (i.e. zero think time).

Each test was a dry run for 20 minutes, after which statistics were collected.

	Jira UI User Actions that result in Exalate code being triggered	Exalate API Actions

	Jira UI User Actions that result in Exalate code being triggered	**Exalate API Actions**
1	Add Worklog	Create Sync Request
2	Edit Work Log	Create Sync Response
3	Delete Work Log	Create Blob Request
4	Attach File	Create Blob Response
5	Delete Attached File	Create Error Response
6	Vote Via Shortcut	Create Blob Error Response
7	Vote By Clicking 'vote'	Get Node Info
8	Vote By Clicking 'vote' On The 'View Votes' Page	Download blob
9	Remove Vote Via 'Remove vote for this issue'	Poll - Get Sync Requests
10	Watch Issue Via 'Watch issue'	Poll - Get Sync Response
11	Watch issue Via 'start watching this issue'	Poll - Get Blob Requests
12	Delete Watcher	Poll - Get Blob Responses
13	Create Sub-task	Poll - Get Error Responses

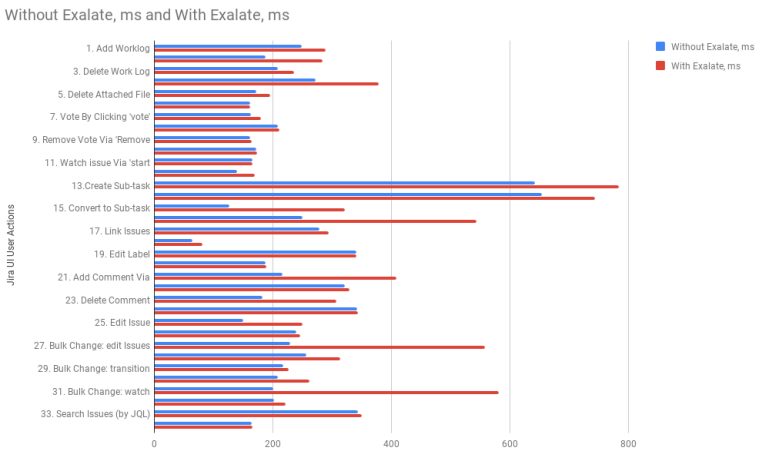
	Jira UI User Actions that result in Exalate code being triggered	**Exalate API Actions**
14	Create Issue	Poll - Get Blob Error Responses
15	Convert to Sub-task	Poll - Received Sync Request
16	Move issue	Poll - Received Sync Response
17	Link Issues	Poll - Received Blob Request
18	Clone Issue	Poll - Received Blob Response
19	Edit Label	
20	Delete Issue	
21	Add Comment Via Comment Button	
22	Edit Comment	
23	Delete Comment	
24	Inline Edit Issue	
25	Edit Issue	
26	Assign via Assign Button	

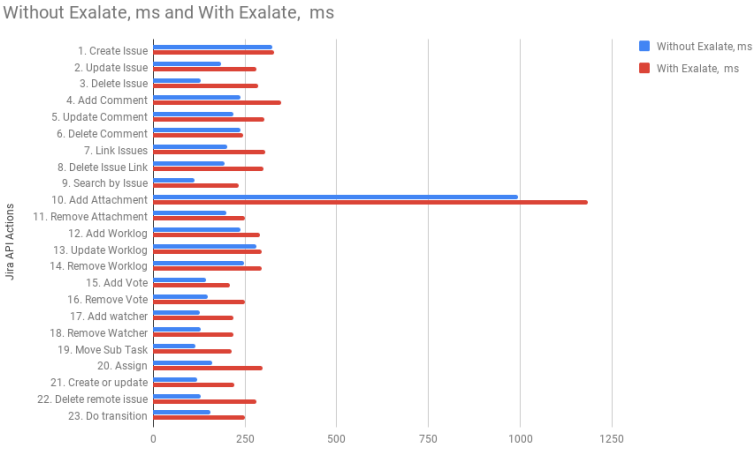
	Jira UI User Actions that result in Exalate code being triggered	Exalate API Actions
27	Bulk Change: edit Issues	
28	Bulk Change: move Issues	
29	Bulk Change: transition issues	
30	Bulk Change: delete issues	
31	Bulk Change: watch Issues	
32	Bulk Change: stop watching Issues	
33	Search Issues (by JQL)	
34	View Issue	

Test Results from Sept 2019

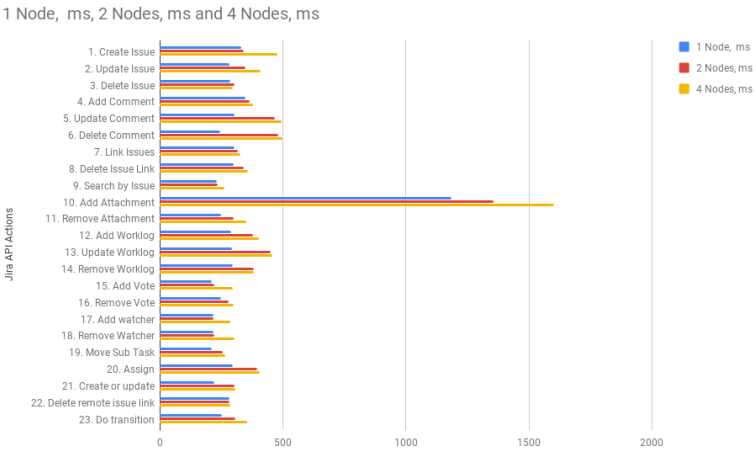
Check the diagrams below to see how the Exalate app affects the performance of the Jira Data Center environment.

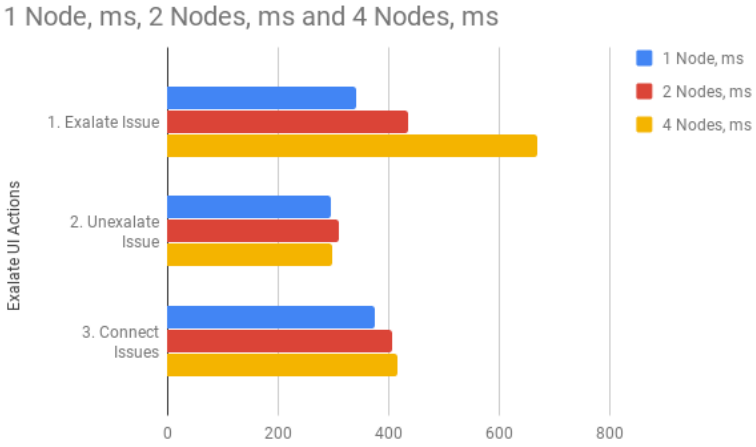
Endpoint testing

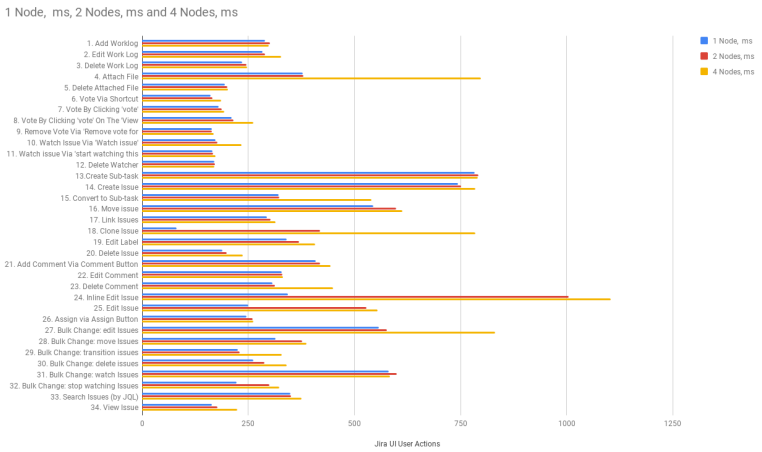


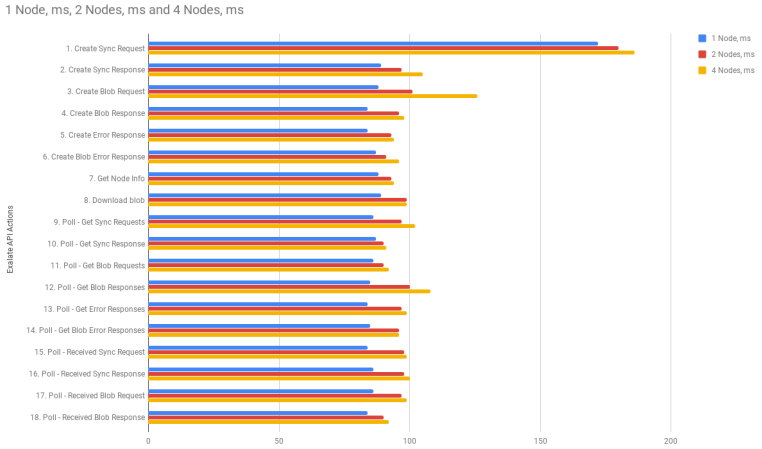


Scale testing









ON THIS PAGE

Resources

- Academy
- Blog
- YouTube Channel
- Ebooks

Still need help?

- Join our Community
- Visit our Service Desk
- Find a Partner