

TDS Collections

This project aims to improve the scalability of the THREDDDS Data Server through the implementation of a cluster of TDS instances that are deployed in the backend and that are only visible from the outside through a reverse proxy in the frontend. This project also aims to improve the management of the TDS catalogs by partitioning the hierarchy of catalogs into the multiple TDS instances that are deployed in the backend, allowing high availability of the datasets and tackling the current problem of large waiting times after performing a THREDDDS reinit when publishing new catalogs.

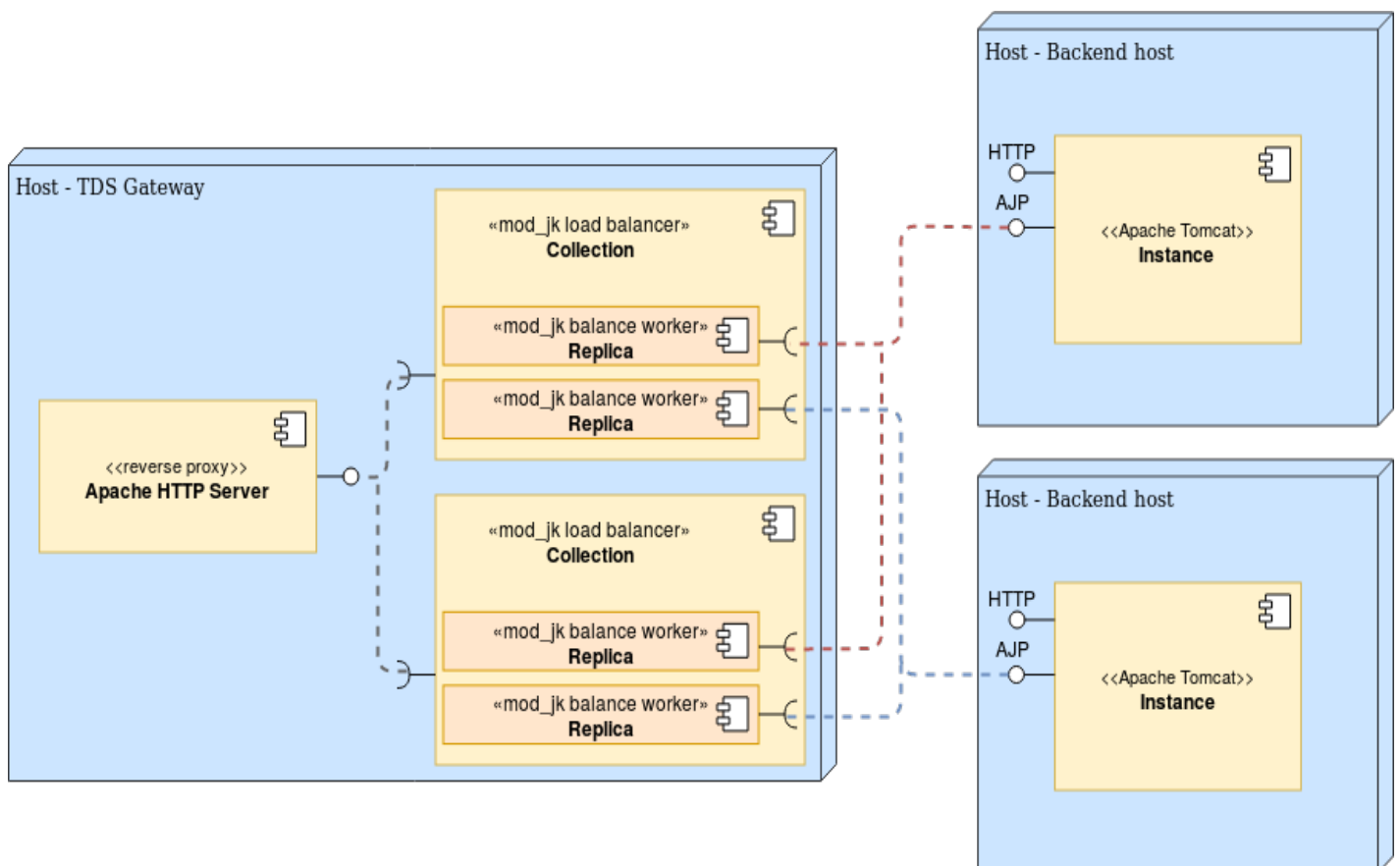
Keywords

tds gateway, reverse proxy, frontend, backend, THREDDDS, TDS, collection, mod_jk, worker, connector, load balancing, instance, replica, urimaps

Entities and relationships

Frontend – Thredds Data Server Gateway and it's components.

Backend – Tomcat servers not visible from the outside and their components. They are accessed via the frontend.



TDS Gateway

The Thredds Data Server Gateway is an Apache web server that works as a reverse proxy, forwarding requests to the backend TDS instances deployed in Tomcat application servers.

Collections

From an user perspective, *collections* are an aggrupation of catalogs and datasets. For example, the collection *Interim* would contain all data that belongs to ERA-Interim. Collections have no more meaning to users. However, collections allow system administrators to partition and replicate the catalog and dataset infrastructure into *replicas*, isolating datasets from the *Interim* collection from datasets of other collections. Thus, changes to the *Interim* collection don't have side effects on the datasets of other collections.

From an administrator perspective, collections are mod_jk workers of type *load balancing* compounded of *replicas*, that is, mod_jk *balance workers*. Every replica points to a tomcat instance deployed in the backend hosts. Each request is routed, through the use of *urimaps*, to the TDS instances that support the collection in which the requested data resides.

Urimaps

Urimaps are rules defined in the configuration of mod_jk that map URLs of requests made by the user to the corresponding collection, in order to be dispatched by one of the instances that belongs to the collection.

TDS Instances

Each TDS instance, or simply *instance*, corresponds to a Tomcat server proccess running in the backend hosts. Each host can run one or more instances. Each instance can contain one or more *connectors* (Connector xml elements in Tomcat's server.xml). Connectors of type AJP are used to connect to the TDS Gateway and they are the bridge between the replicas and the instances. Usually, you only need one Connector element per TDS Gateway/reverse proxy in the deployment, since Connector elements must reference their proxy.

Each instance has the option to support any number of collections and each instance will only respond to requests targeting data available in the supported collections. This is done through the TDS Gateway urimaps.