Concurrency control in WhereMyTube on Google App Engine

One of the core requirements for WMT is to make a fewer network calls as possible to the TFL Web site to obtain Arrival Board data for trains, and no calls should be made to the TFL site if no users are currently on the site (we’re not expecting a huge number of users at 5am). This requirement has lead to the design decision of retrieving for AB data on-demand when a user opens the map page, and caching that data for subsequent requests until the data becomes state and the AB data must be re-loaded. AB parsing essentially happens on-demand.

If a process reads cached AB data an realises it is out-of-data it initiates an AB parse.

This raises potential concurrency issues because there should never be separate requests parsing the same line at the same time some sort of synchronisation must occour between processes to coordinate which process will actually do the AB parsing and which will wait for the result. This is achived by one process setting a flag in the DS indicating it is parsing the line, if two prcesses come along at the same time and try to set the flag one write will result in an optimistic lock check error and will know another process has started to process the line.

While the algorithms developed for WMT substantially cut down on TFL calls so that the whole application only gets the data for one line every minute

Requirements:

* As few as possible calls to the TFL site should be made and no calls should be made to the TFL site if no users are currently on the site

Requests should be as short as possible to comply with GAE rules and general Web application good practice – append-only locking

Why not always poll? May make sense if there are always users on the site