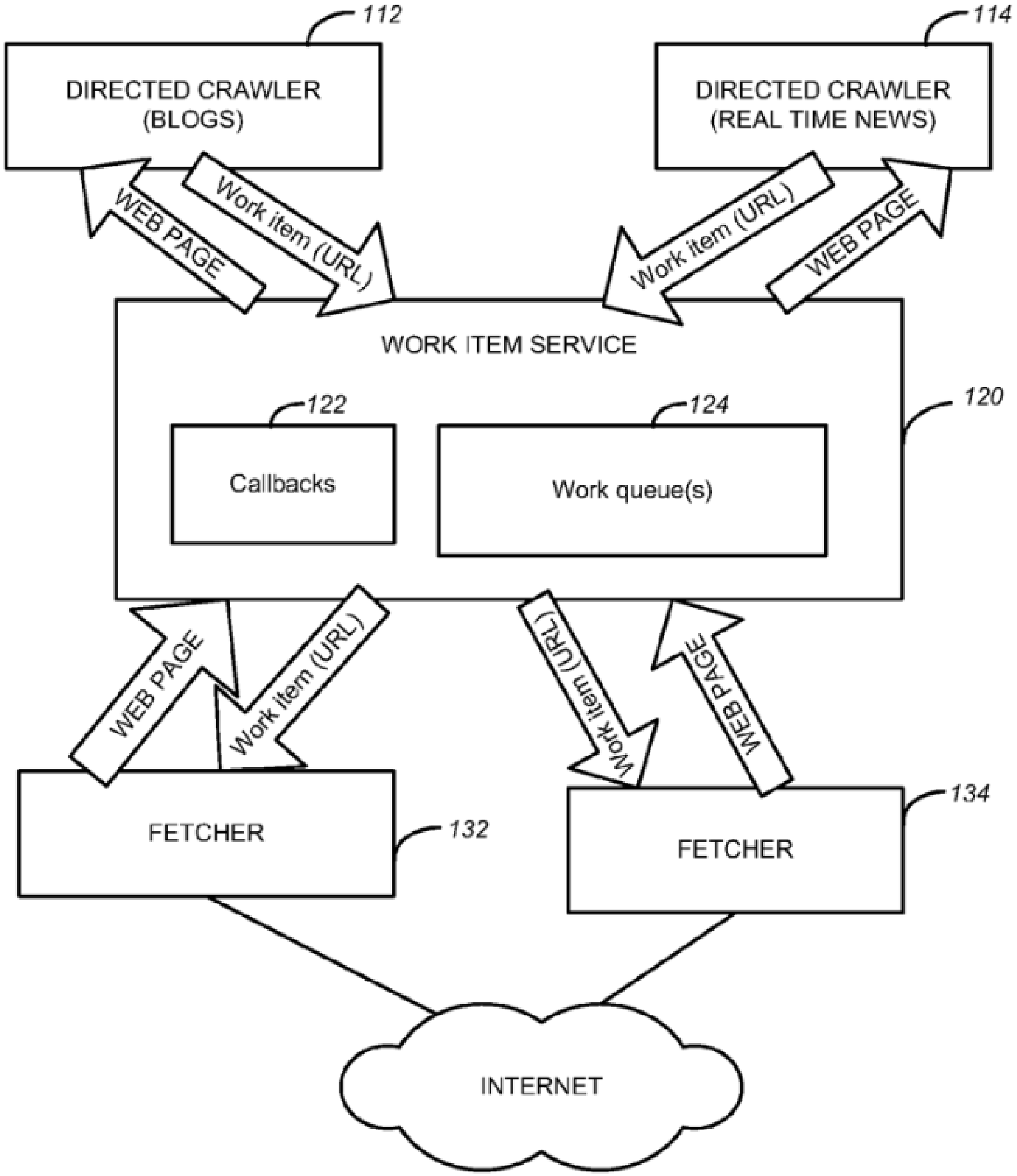
Distributed Web Crawler Architecture

Introduction

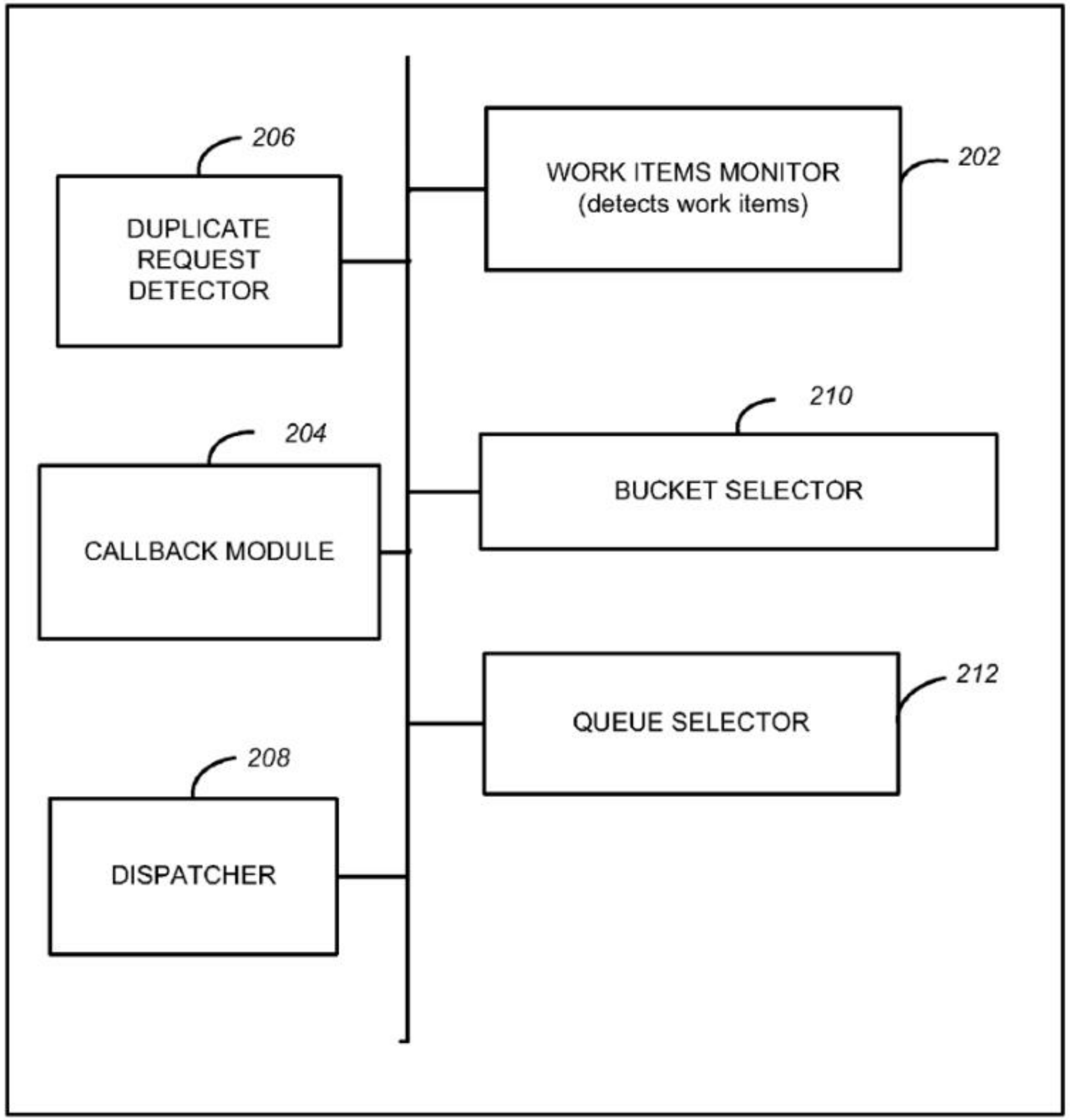
- Distributed web crawler architecture is made up of work items (URLs to be accessed), a duplicate request detector, and a callback module
- Web crawlers are divided into crawler (generates work items) and fetcher modules (fetches work items)
- “Directed” crawlers: each crawler targets a specific type of web pages (eg. blog pages, real time news)

Work Item Service

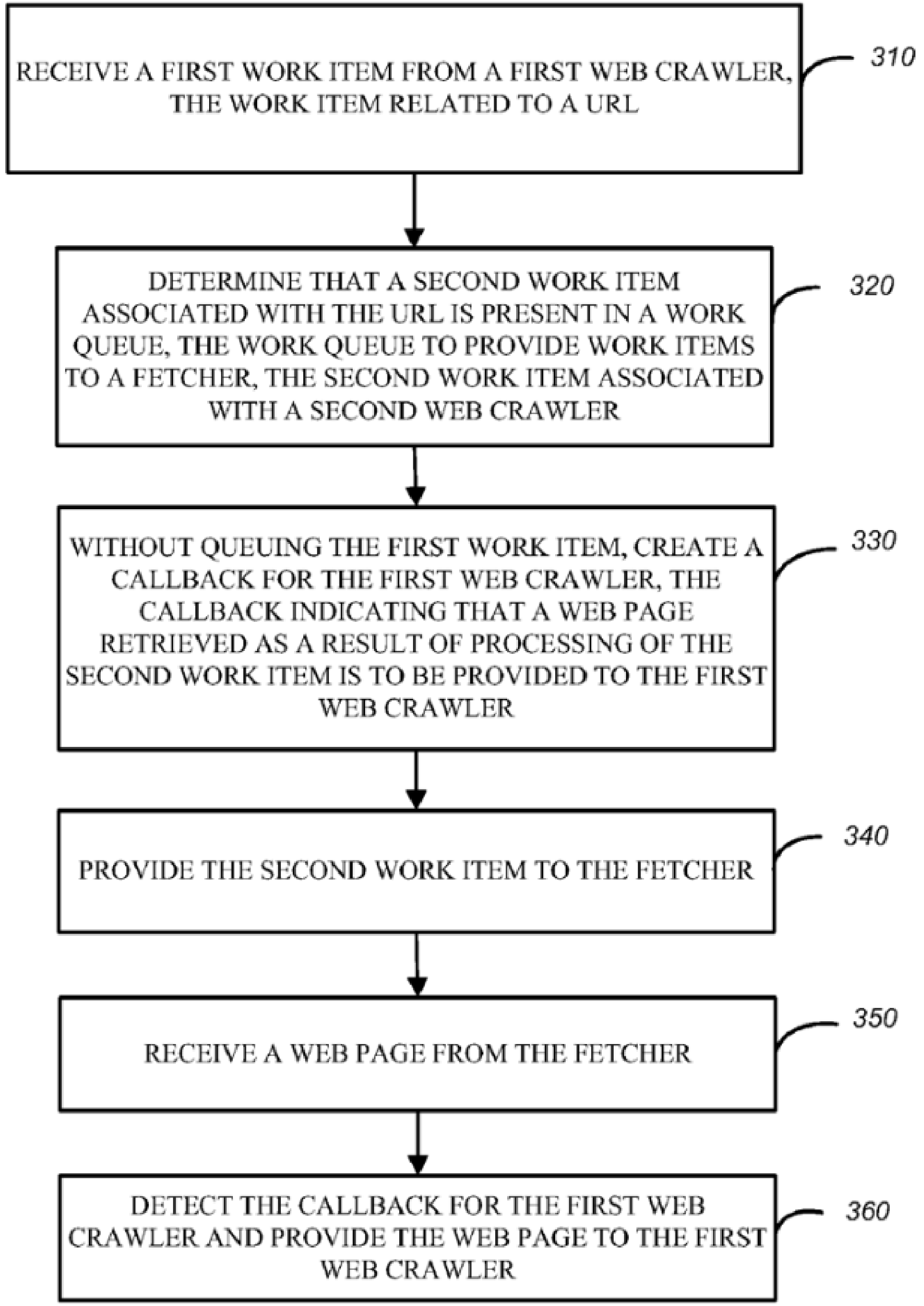
- Intermediary work item service
  - receives work items from crawlers
  - queues them in work queues
  - each sends work items to fetchers periodically
  - fetcher sends fetched web page back to work item service
  - these web pages are sent back to crawlers present in callbacks list (URL list where each URL is mapped to target web crawlers)
  - if the URL associated with a work item (from crawler A) is already present in a work queue, callback is created with address of A so that when the web page is fetched, data is returned to A as well



- Work item service comprises of:
  - work items monitor: detects incoming work items
  - callback module: holds information about source web crawler for each work item (URL: address pairs)



- duplicate requests detector: doesn’t queue repeat requests, callback is updated instead



RECEIVE A FIRST WORK ITEM FROM A FIRST WEB CRAWLER, THE WORK ITEM RELATED TO A URL

DETERMINE THAT A SECOND WORK ITEM ASSOCIATED WITH THE URL IS PRESENT IN A WORK QUEUE, THE WORK QUEUE TO PROVIDE WORK ITEMS TO A FETCHER, THE SECOND WORK ITEM ASSOCIATED WITH A SECOND WEB CRAWLER

WITHOUT QUEUING THE FIRST WORK ITEM, CREATE A CALLBACK FOR THE FIRST WEB CRAWLER, THE CALLBACK INDICATING THAT A WEB PAGE RETRIEVED AS A RESULT OF PROCESSING OF THE SECOND WORK ITEM IS TO BE PROVIDED TO THE FIRST WEB CRAWLER

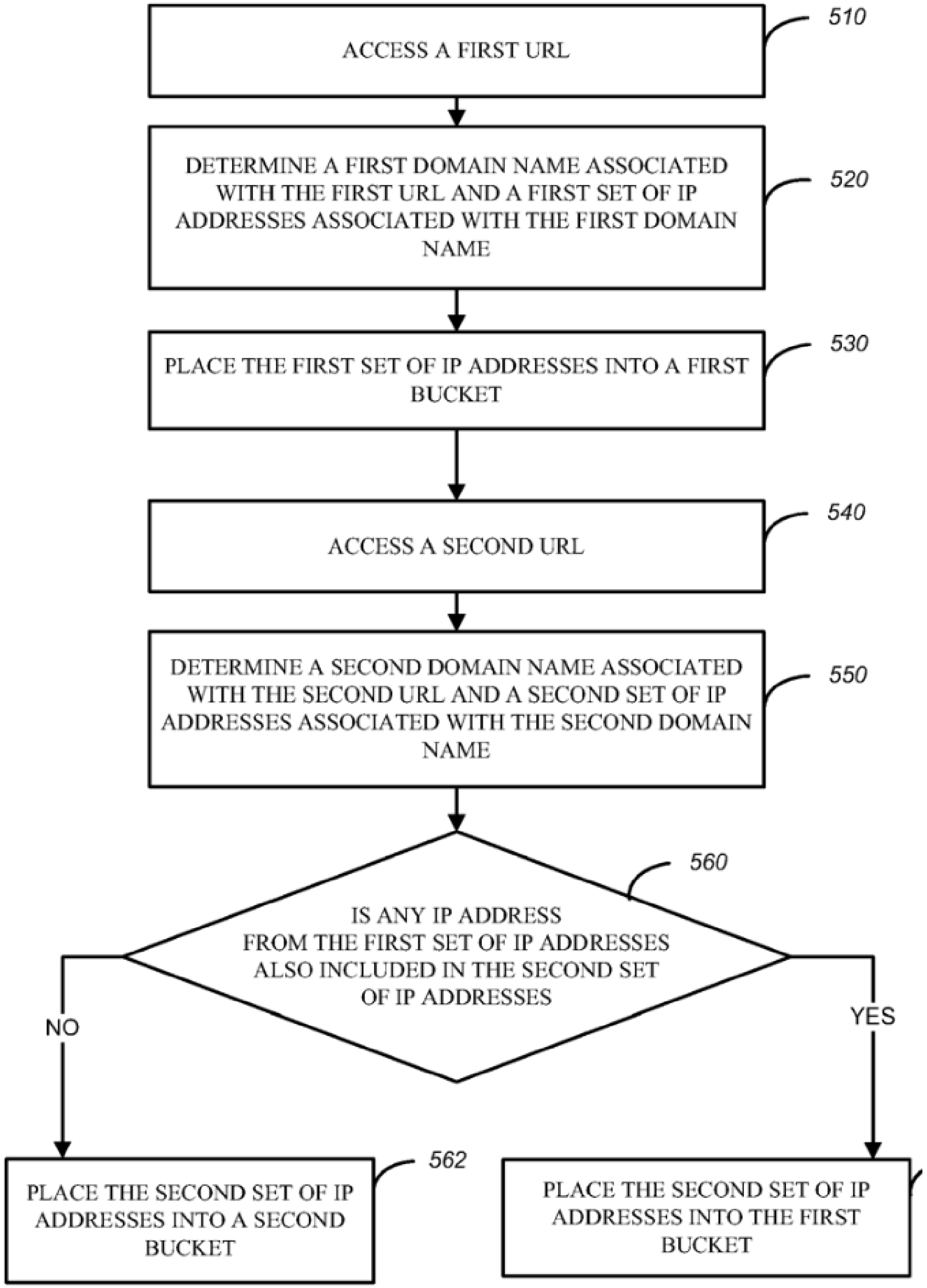
PROVIDE THE SECOND WORK ITEM TO THE FETCHER

RECEIVE A WEB PAGE FROM THE FETCHER

DETECT THE CALLBACK FOR THE FIRST WEB CRAWLER AND PROVIDE THE WEB PAGE TO THE FIRST WEB CRAWLER

Bucket Service

- work items mapped to buckets based on URL/IP
- overcomes issue of overloading single web server (IP) by avoiding repetitive requests by different fetchers (in case two distinct domains have overlapping IPs)
- fetchers poll buckets to pick up work items; queues (associated with buckets) release work items at predetermined frequencies
- each bucket is associated with its own work queue



510
ACCESS A FIRST URL
520
DETERMINE A FIRST DOMAIN NAME ASSOCIATED WITH THE FIRST URL AND A FIRST SET OF IP ADDRESSES ASSOCIATED WITH THE FIRST DOMAIN NAME
530
PLACE THE FIRST SET OF IP ADDRESSES INTO A FIRST BUCKET
540
ACCESS A SECOND URL
550
DETERMINE A SECOND DOMAIN NAME ASSOCIATED WITH THE SECOND URL AND A SECOND SET OF IP ADDRESSES ASSOCIATED WITH THE SECOND DOMAIN NAME
560
IS ANY IP ADDRESS FROM THE FIRST SET OF IP ADDRESSES ALSO INCLUDED IN THE SECOND SET OF IP ADDRESSES
NO
YES
562
PLACE THE SECOND SET OF IP ADDRESSES INTO A SECOND BUCKET
PLACE THE SECOND SET OF IP ADDRESSES INTO THE FIRST BUCKET

Reference

Distributed web crawler architecture, by S. Severance. (2011, Dec. 15). US20110307467A1 [Online]. Available: https://patents.google.com/patent/US20110307467A1