Distributed Crawler Application

Main requirements specification, phase A, 2014-01-26

Characterization

The Distributed Crawler Application (DC) application is common purpose service class software. The main functional goal is to provide possibilities for users to fetch data from data sources and to create huge collections of resources. Basic features include the web-crawling algorithms implementation with standard (web http-based) or custom resource fetcher. The application based on Distributed Tasks Manager (DTM) and implements applied crawling algorithms and architecture as distributed tasks. Additionally provides interface for extended post crawling content processing like scrapping, NLP, statistical and so on.

The main features list:

- http links collections,
- resources collections and distributed storages,
- distributed data fetcher IPs,
- planned and managed http requests frequency,
- custom data fetchers, request protocols, resources formats,
- free scalability and hot changes of productivity,
- different model of resources sharding,
- parameters templates and inheritance,
- customizable target data storages, including native file system, key-value DB, SQL BD and other,
- durability and failsafe principles

Conditions

Standalone boxed application that works as a client-side top user level of DTM application server.

Usage Domain

Applied by developers and system integrators in target projects as server-side software and tools set. Most common purposes:

- Distributed asynchronous parallel web sites crawling.
- Distributed raw web resources data storages management.
- Distributed URLs database storages management.
- Distributed raw web resources post crawling processing as chained tasks.

Technology
Single process, multithreaded, Python language application implementation, integrated with DTM application server by HCE Python API bindings.

Object model
Threaded control objects, MOM-based networking, ZMQ tcp and inproc sockets, messages communications, messages queues, json serialization protocol.

Functionality
Process client requests, managed automated web-sites crawling, resource management.

Philosophy
Asynchronous multithreaded mutexless messages handling, parallel computations, asynchronous distributed remote tasks processing, automated managed batch processing, distributed resources management.