Course Knowledge Graph from USC, UCLA, UCB

Zi Gu, zigu@usc.edu

Haili Wang, hailiwan@usc.edu

Outline

1. Building a Distributed web crawler

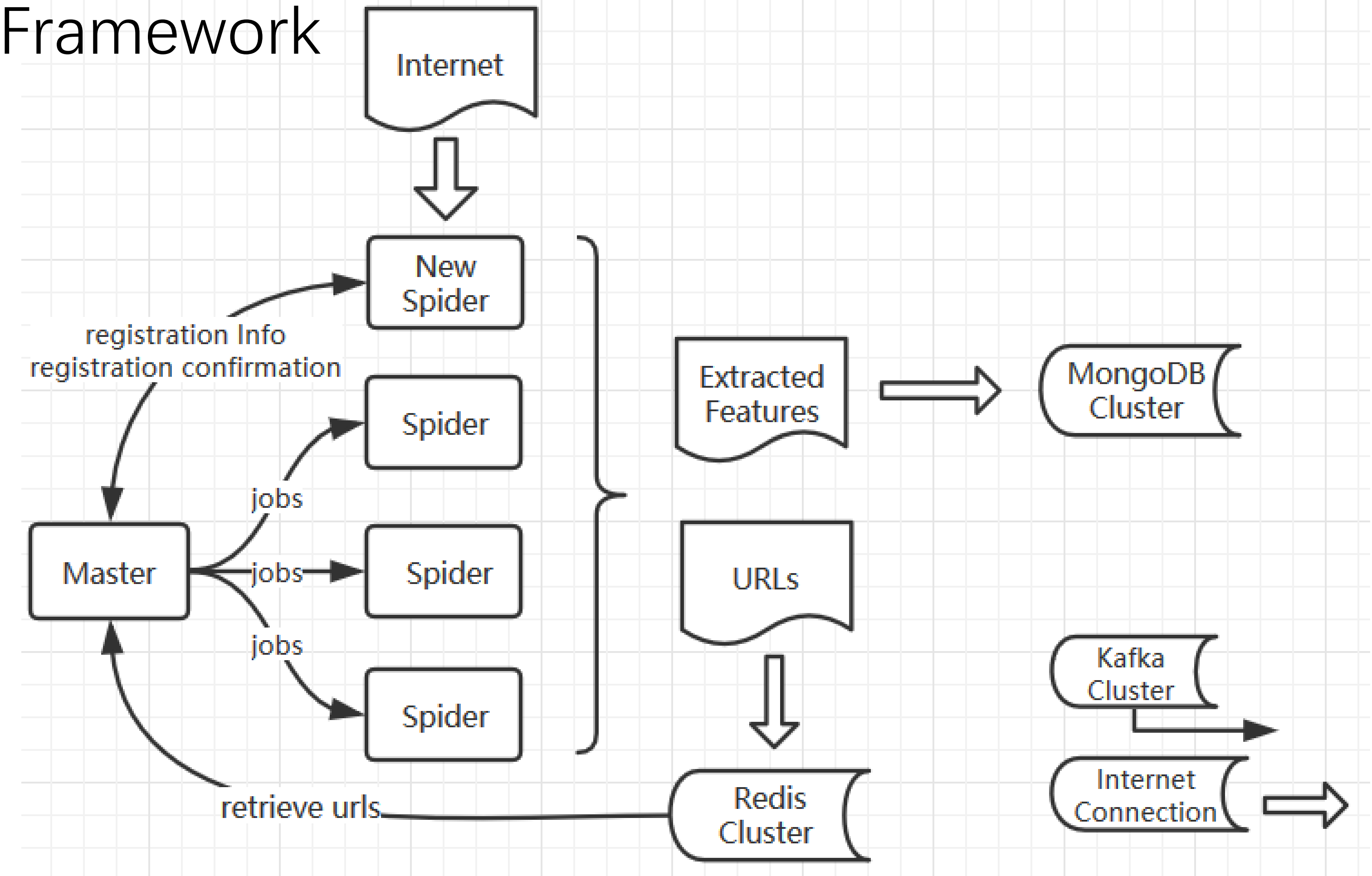
2. Building a knowledge Graph

Outline

1. Building a Distributed web crawler
    - Master & spiders structure
    - Guaranty politeness and priority
    - Base on Apache Kafka, Redis, MongoDB and scrapy

Building a Distributed web crawler

- General Framework



Building a Distributed web crawler

- Priority & Politeness - Master
  - Master configuration: domain priority, num pages/second
  - Randomly distribute jobs with care of priority & politeness

Building a Distributed web crawler

- Internal service reasoning
  - Distributed cluster service

  - Apache Kafka – Robust message send/receive platform
  - Redis – Memory based DB, fast for short message exchange
  - MongoDB – NoSQL persistence

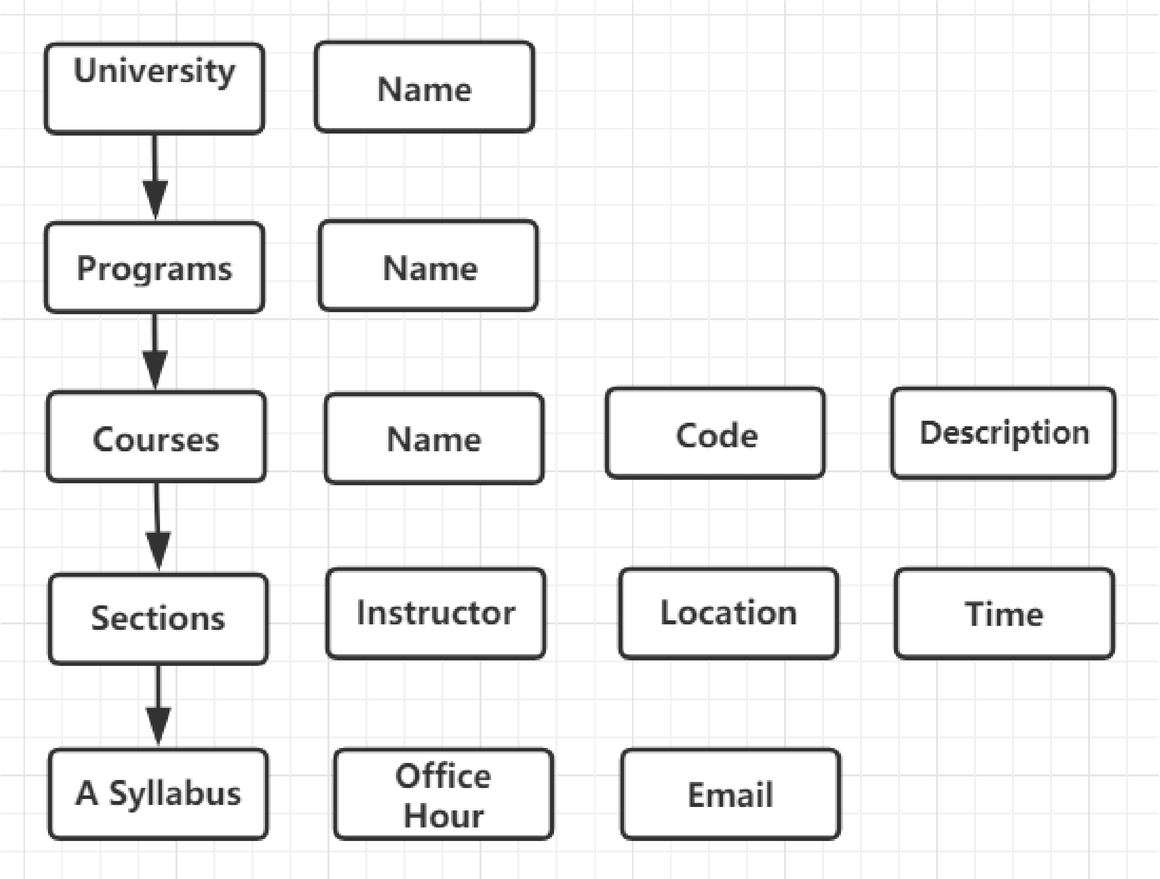
Outline

1. Building a Distributed web crawler

2. Building a knowledge Graph
   - Ontology
   - PDF extraction
   - Instructor entity matching
   - Elasticsearch & queries

Building a knowledge graph

- Real world entities



University Name

Programs Name

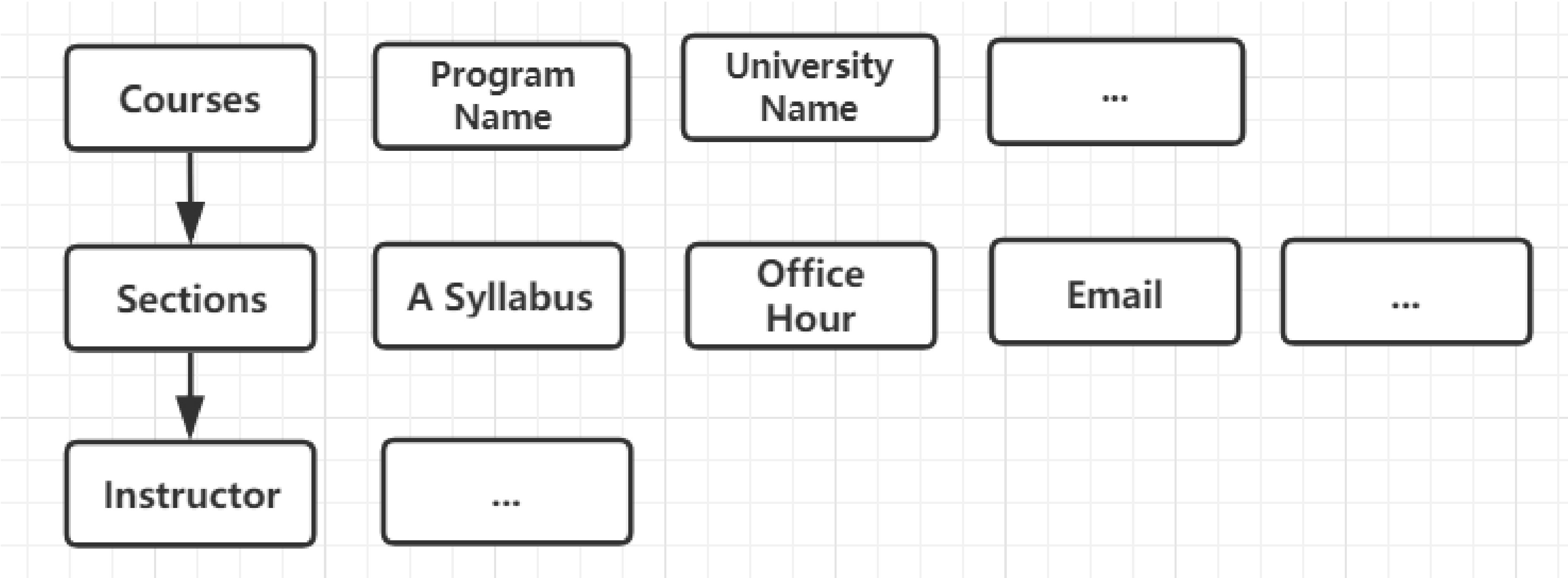
Courses Name Code Description

Sections Instructor Location Time

A Syllabus Office Hour Email

Building a knowledge graph

- Implementation



Building a knowledge graph

- Course
  - Course Abbreviation, Course Name, Description
  - Program Name, University Name
  - Section IDs
- Section
  - Days, Time, Location, Syllabus Link, Email, Session Type, Office Hour
  - Instructor ID
- Instructor
  - Instructor Name, Instructor Link

Building a knowledge graph

- PDF syllabus extraction
  - Instructor email, Section office hour
  - PDF => text
  - Regular expression

- Instructor entity matching
  - Similarity of instructor name
  - Edit distance

Building a knowledge graph

- Application
  - Elasticsearch
  - Tkinter Python GUI, JSON Viewer

  - Breadth query – Same type ranked by similarity score
  - Depth query – Linked objects

Course Knowledge Graph from USC, UCLA, UCB

1. Building a Distributed web crawler
   - Master & spiders structure
   - Guaranty politeness and priority
   - Base on Apache Kafka, Redis, MongoDB and scrapy

2. Building a knowledge Graph
   - Ontology
   - PDF extraction
   - Instructor entity matching
   - Elasticsearch & queries