

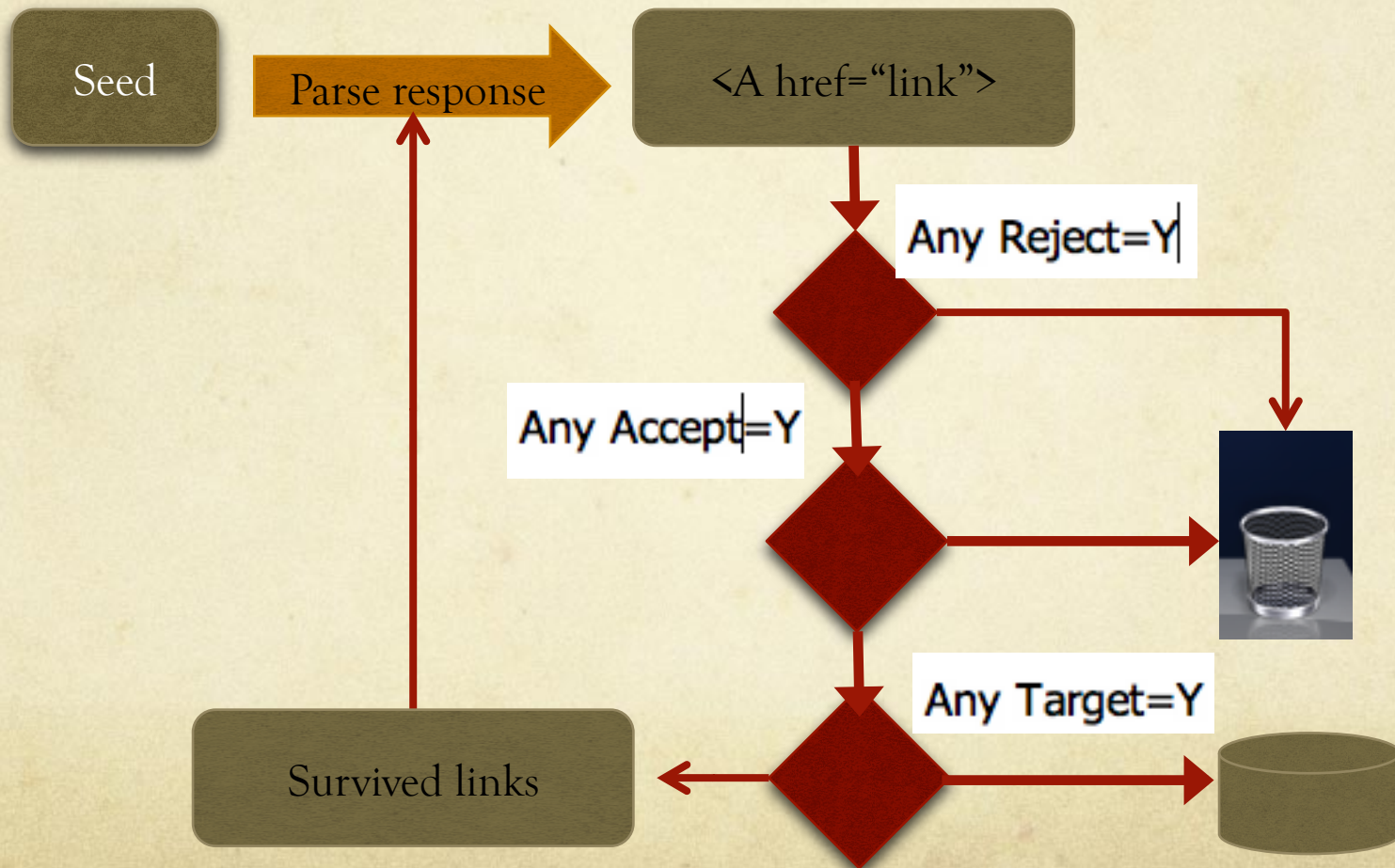
# crawlman

Usage&Design

By jingjing.zhijj

# Work Flow

<http://www.1688.com>





# Configuration

- 1) configure reject rule[s]
- 2) configure accept rule[s]
- 3) configure target rule[s]

# Sample code

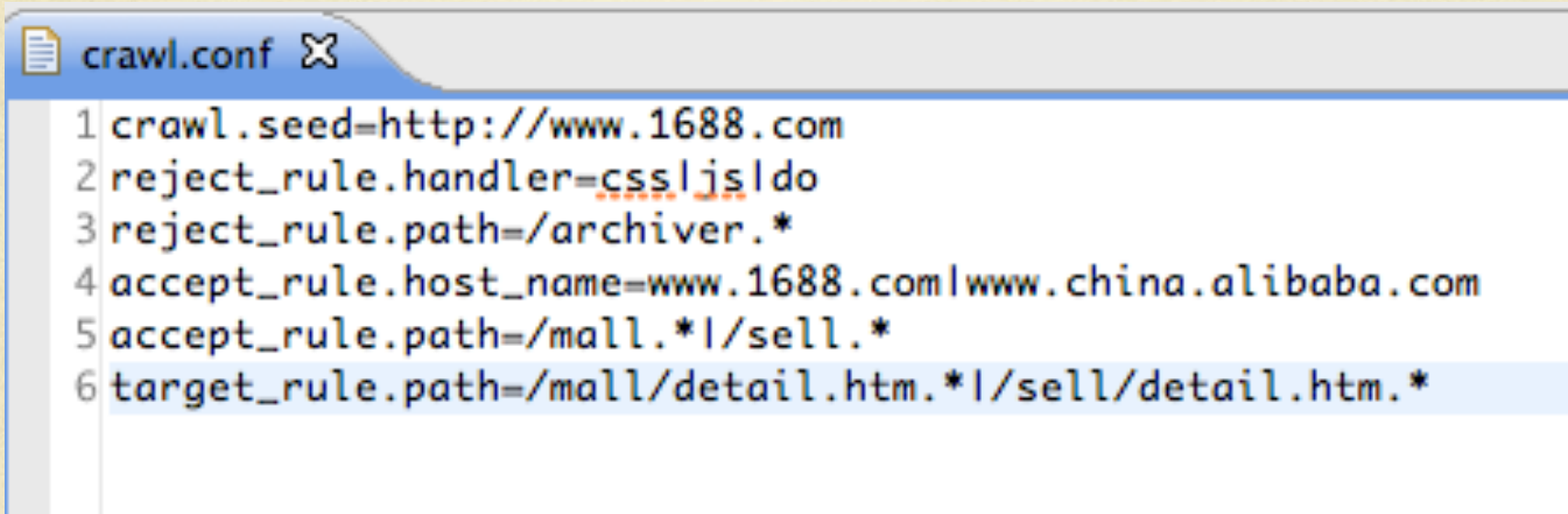
```
public void testSample() throws CrawlException {
    ApplicationContext appContext = new ClassPathXmlApplicationContext(
        new String[] { "classpath:/crawl.xml" });
    CrawlJobRunner crawlJobRunner = (CrawlJobRunner) appContext
        .getBean("crawlJobRunner");
    Assert.assertNotNull(crawlJobRunner);

    CrawlJob job = new CrawlJob("http://www.1688.com");
    // accept multiple reject rules
    job.putConf(CrawlJob.HANDLER_REJECT_RULE_KEY, "css|js|do");
    job.putConf(CrawlJob.PATH_REJECT_RULE_KEY, "/archiver.*");
    job.putConf(CrawlJob.HOST_NAME_ACCEPT_RULE_KEY, "www.1688.com|www.china.alibaba.com");
    job.putConf(CrawlJob.PATH_ACCEPT_RULE_KEY, "/mall.*|/sell.*");

    job.putConf(CrawlJob.TARGET_RULE_KEY
        + "=/mall/detail.htm.*|/sell/detail.htm.*");

    crawlJobRunner.setJob(job);
    crawlJobRunner.instantiate();
    crawlJobRunner.startJob();
}
```

# Sample configuration



The image shows a screenshot of a text editor window with a tab labeled 'crawl.conf'. The editor contains six lines of configuration text. The sixth line is highlighted with a light blue background. The text is as follows:

```
1 crawl.seed=http://www.1688.com
2 reject_rule.handler=cssljsldo
3 reject_rule.path=/archiver.*
4 accept_rule.host_name=www.1688.com|www.china.alibaba.com
5 accept_rule.path=/mall.*|/sell.*
6 target_rule.path=/mall/detail.htm.*|/sell/detail.htm.*
```

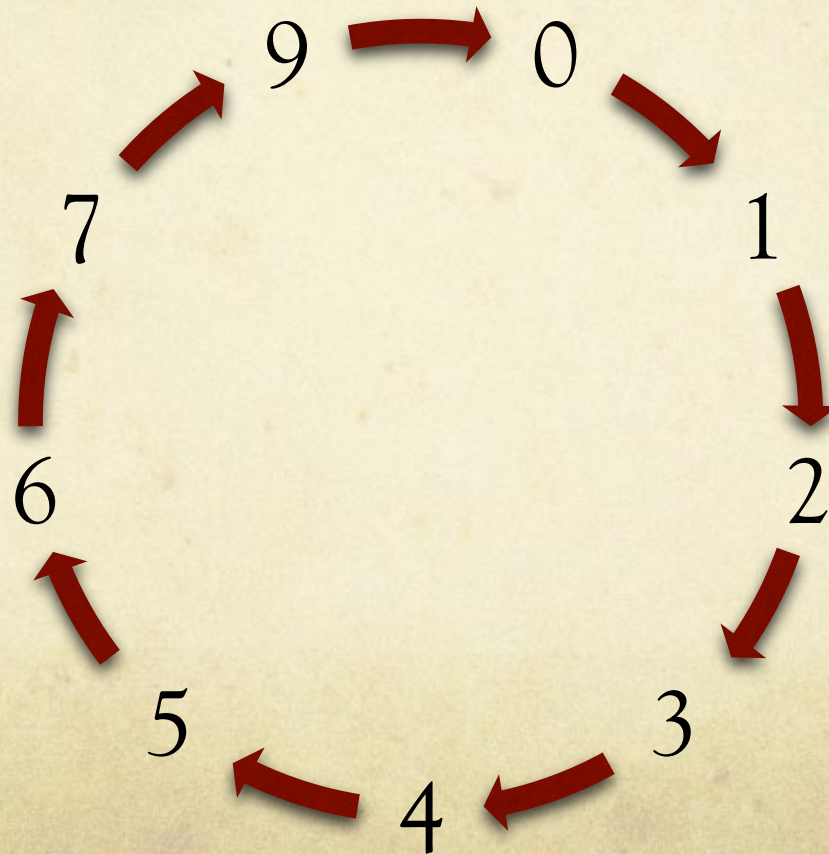


# Design issues

- How to determine the request flow?
- How to control request flow?
- How to determine the job progress?
- How to get an efficient way to avoid endless recursive requests?
- How to get the job resume working from the state interrupted?

# Flow Bucket

- Every timed bucket record the requests number, like the first one record 0~100 millisecond.





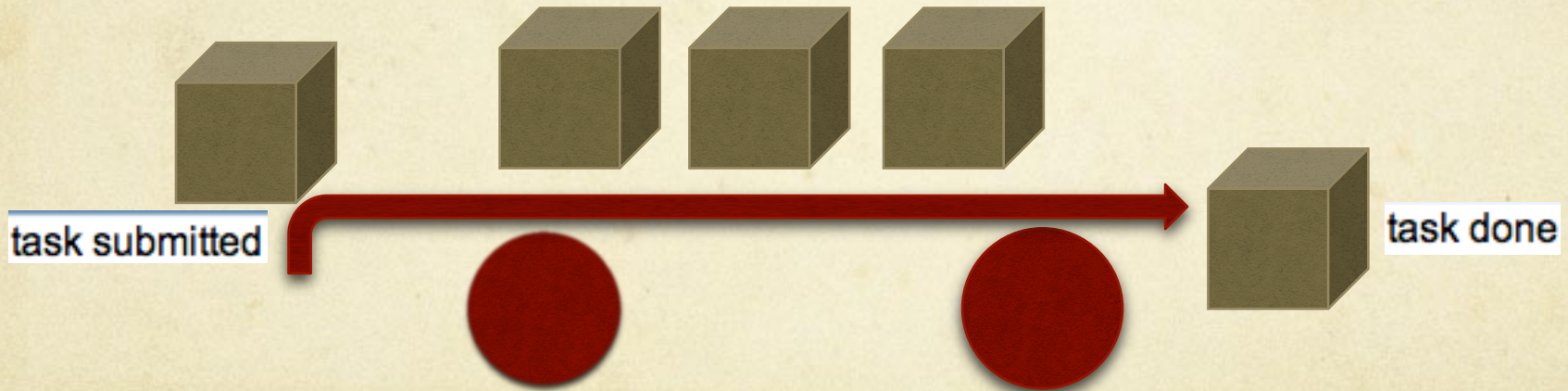
# Request flow in the last second

```
/**
 * get flow in the latest one second
 *
 * @return
 */
public long getFlow() {
    long sum = 0;
    for (long bucketFlow : this.bucketFlows) {
        sum += bucketFlow;
    }
    return sum;
}
```



# Job progress

- Request Bus— transfer tasks
- Event Bus—tell if task is submitted or done
- Task status -task submitted or done



# Job Progress Checker

```
private void check() {
    if(log.isInfoEnabled()){
        log.info("JobProgressChecker start to check...");
    }
    while (!stop) {

        long finished = this.jobState.getFinished();
        long submitted = this.jobState.getSubmitted();
        long remaining = submitted - finished;

        if (log.isDebugEnabled()) {
            log.debug("finished: " + finished + ", submitted:" + submitted
                + ", remaining: " + remaining);
        }

        if (remaining== 0 && this.requestBus.isEmpty()) {
            if (log.isInfoEnabled()) {
                log.info("Job finished");
            }

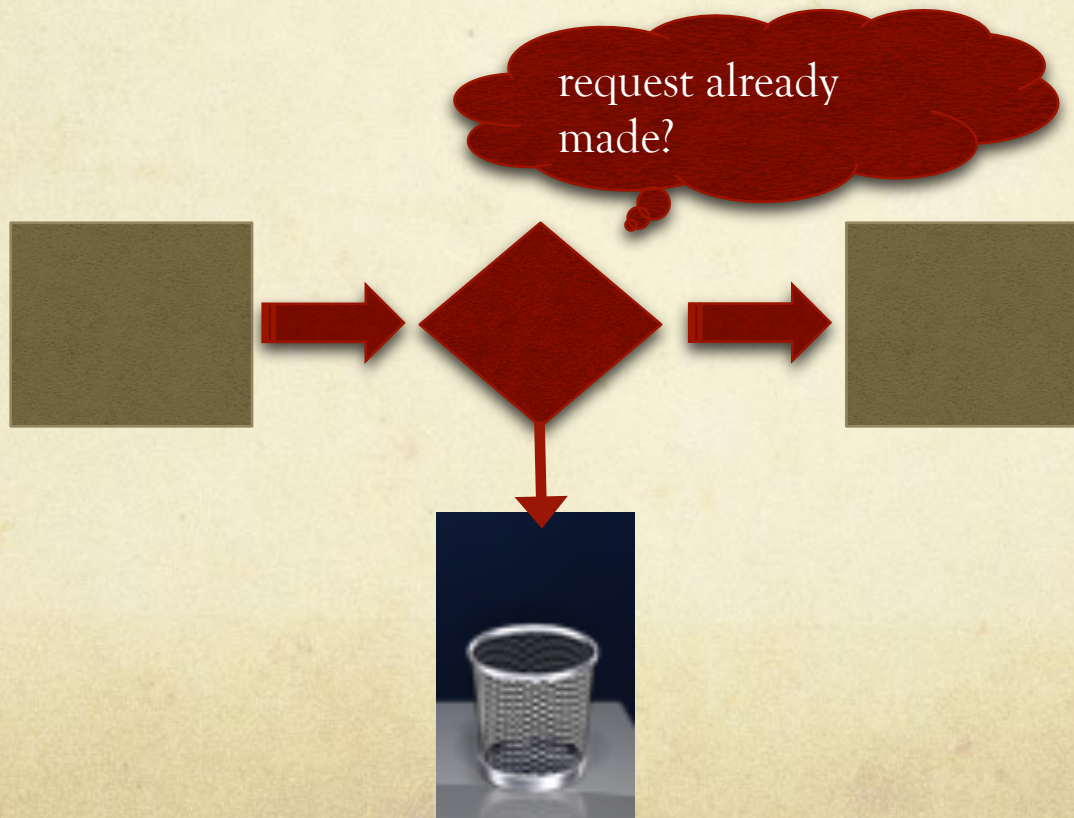
            eventBus.publishEvent(new CrawlJobEvent(
                CrawlJobEvent.CRAWL_JOB_FINISHED, "Job finished."));
            break;
        }

        sleep();
    }
}
```



# Avoid recursive requests

- Record issued requests
- Request Objects maybe too large to save in memory



# Job & Request state

- Certain workers have to know Job & Tasks' status





# Log

```
(0 ms) [main] INFO : webcrawl.run.CrawlJobRunner#startRunnerThread : event bus now start to service.
(68 ms) [main] INFO : webcrawl.run.CrawlJobRunner#startJob : start crawling with seed: http://www.***.net
(72 ms) [main] INFO : webcrawl.run.RequestBus#submit : http://www.***.net added to request queue.
(77 ms) [Thread-3] INFO : webcrawl.run.JobProgressChecker#check : JobProgressChecker start to check...
(77 ms) [Thread-3] INFO : webcrawl.run.JobProgressChecker#check : JobProgressChecker start to check...
(78 ms) [Thread-3] DEBUG: webcrawl.run.JobProgressChecker#check : finished: 0, submitted:1, remaining: 1
(78 ms) [Thread-3] DEBUG: webcrawl.run.JobProgressChecker#check : finished: 0, submitted:1, remaining: 1
(79 ms) [Thread-4] INFO : webcrawl.run.FlowMonitor#check : FlowMonitor start to check...
(1080 ms) [Thread-4] DEBUG: webcrawl.run.FlowMonitor#check : flow is :1
(1080 ms) [Thread-4] DEBUG: webcrawl.run.FlowMonitor#check : speeding up job runner, lower limit:20
(1080 ms) [Thread-4] INFO : webcrawl.run.HttpRequestTaskExecutorEngine#setCorePoolSize : corepool size resize to 12, minPoolSize: 8, maxPoolSize:50
(1951 ms) [pool-2-thread-1] DEBUG: webcrawl.run.HttpRequestTask#run : http://www.***.net requested.
(2081 ms) [Thread-4] DEBUG: webcrawl.run.FlowMonitor#check : flow is :1
(2081 ms) [Thread-4] DEBUG: webcrawl.run.FlowMonitor#check : speeding up job runner, lower limit:20
(2082 ms) [Thread-4] INFO : webcrawl.run.HttpRequestTaskExecutorEngine#setCorePoolSize : corepool size resize to 14, minPoolSize: 8, maxPoolSize:50
(2155 ms) [pool-2-thread-1] DEBUG: webcrawl.run.AcceptRequestMatcher#match : not match[accept], rule:/mall.*!sell.*, url:http://www.***.net/announce/1.htm
(2156 ms) [pool-2-thread-1] DEBUG: webcrawl.run.AcceptRequestMatcher#match : match[accept], rule:/mall.*!sell.*, url:http://www.***.net/sell/list.php?ca

(180083 ms) [Thread-3] DEBUG: webcrawl.run.JobProgressChecker#check : finished: 1000, submitted:1000, remaining: 0
(180084 ms) [Thread-3] INFO : webcrawl.run.JobProgressChecker#check : Job finished
(180084 ms) [Thread-3] INFO : webcrawl.run.JobProgressChecker#check : Job finished
(180084 ms) [Thread-2] INFO : webcrawl.run.JobProgressChecker#onJobFinished : stop checking on job finished.
(180084 ms) [Thread-2] INFO : webcrawl.run.JobProgressChecker#onJobFinished : stop checking on job finished.
(180084 ms) [Thread-2] INFO : webcrawl.request.MemoryRequestGuardar#onJobFinished : submit:0, finished:0
(180084 ms) [Thread-2] INFO : webcrawl.request.MemoryRequestGuardar#onJobFinished : memory clear on job finished.
(180084 ms) [Thread-2] INFO : webcrawl.run.HttpRequestTaskExecutorEngine#onJobFinished : executor engine was shut down
(180085 ms) [Thread-2] INFO : webcrawl.run.HttpRequestTaskExecutorEngine#onJobFinished : executor engine was shut down
(180085 ms) [Thread-2] INFO : webcrawl.run.CrawlJobRunner#pollEvent : event bus now stop to service.
(180295 ms) [Thread-4] DEBUG: webcrawl.run.FlowMonitor#check : flow is :38
(180295 ms) [Thread-4] INFO : webcrawl.run.FlowMonitor#check : stop flow monitoring on job finished.
```

# Improvements?

- Start multiple crawler jobs in one java progress?
- Make it a cluster crawler?
- Follow redirect response?
- Follow custom query target & parameters?
- More like a real browser when encounter js & ajax response?