Scalable Parallel Spin-image Generation Using Dynamic Loop Scheduling

Ahmed Eleliemy, Ali Mohammed, and Florina M. Ciorba
Department of Mathematics and Computer Science
University of Basel, Switzerland
Email: {ahmed.eleliemy, ali.mohammed, florina.ciorba}@unibas.ch

June 16, 2017

Algorithm 1: The proposed APSIA master perspective

```
generatingSpinImages (OF, W, B, S, N, DM)
Inputs : OF: location of the input data, W: image width, B: bin size,
         S: support angle, N: number of generated spin-images,
         DM: DLS technique
Output: spinImages: list of generated spin images
OP = read3DPoints(OF)
scheduledTasks = 0
schedulingStep = 0
receivedResults = 0
startEnd[2]
workersCount = getCountOfWorkers()
sendToWorkers(OP)
while $scheduledTasks < N$ do
    requestWork = receiveRequestAnyWorker()
    worker = getSourceOfRequest(requestWork)
    chunk = getChunk(DM, schedulingStep, N, workersCount)
    startEnd[0] = scheduledTasks
    startEnd[1] = scheduledTasks + chunk
    sendResponse(worker, startEnd, assignWork)
    scheduledTasks = scheduledTasks + chunk
end
while $receivedResults < workersCount$ do
    request = receiveRequestFromAnyWorker()
    requestType = getRequestType(request)
    worker = getSourceOfRequest(request)
    if $requestType = assignWork$ then
        sendResponseToWorker(worker, NULL, terminate)
    else
        receiveDataFromWorker(worker, tempSpinImages)
        add(spinImages, tempSpinImages)
        receivedResults++
    end
end
```

Algorithm 2: The proposed APSIA worker perspective

```
generatingSpinImages (OF, W, B, S, DM)
Inputs : OF: location of the input data, W: image width, B: bin size,
         S: support angle, DM: DLS technique
Output: spinImages: list of generated spin images
receiveFromMaster(OP)
M = getLength(OP)
startEnd[2]
sendRequest(assignWork)
response = receiveResponseFromMaster()
spinImages = createSpinImagesList(M)
while $response = assignWork$ do
    startEnd = getResponseData(response)
    /* as shown in Algorithm 2 */
    adCalculateSpinImages(W, B, S, OP, M, spinImages, startEnd[0],
    startEnd[1])
end
sendDataToMaster(spinImages)
```