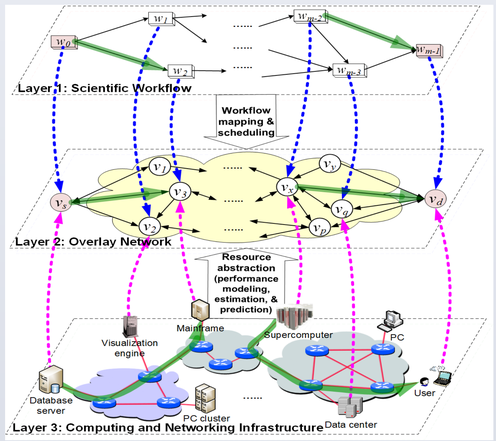
Resource and Performance Optimization of Big Data Scientific Workflows in Distributed Network Environments

Yi Gu, Dept. of Computer Science, Middle Tennessee State University

INTRODUCTION

- Supercomputing facilitates and accelerates scientific discovery
- BIG DATA: from T to P, to E, to Z, to Y, and beyond…
  - ❖ Simulation, Experimental, Observational
- Collaborative research and discovery
  - ❖ A team of scientists with complementary expertise working jointly
- Massively distributed resources
  - ❖ Hardware - Computing facilities, Storage system, i.e., Cloud, Special rendering engines, Display devices (tiled display, powerwall, etc.)
  - ❖ Software - Domain-specific analysis/processing tools/programs...
  - ❖ Data - Data canter in cloud
- Data-, computation-, and network-intensive scientific applications pose new challenges
- An increasing trend in many big data scientific applications to move computing workflow executions to a cloud environment
- A three-layer architecture for performance modeling and optimization of big data scientific workflows in distributed environments



PROBLEMS AND OBJECTIVES

- Problem 1
  - ❖ Given a cloud environment modeled as a faulty heterogeneous computer network $G_{cn} = (V_{cn}, E_{cn})$, where each node or link is associated with an independent failure rate, and a bound $\boldsymbol{F}$ for the overall failure rate (OFR);
  - ❖ Each computer node is DVFS enabled;
  - ❖ A user submits a request of a streaming application modeled as a DAG-structured computing workflow $G_w = (V_w, E_w)$, and a bound $\boldsymbol{FR}$ on the frame rate (FR) or throughput.
  - ❖ Objective: To find a workflow mapping schedule such that the total energy consumption (TEC) is minimized and overall performance satisfies the $\boldsymbol{F}$ and $\boldsymbol{FR}$ constraints.

$$F_{OFR} \le \boldsymbol{F}$$

$$\min_{\text{all possible mappings}} (EC), \text{ Where } \frac{1}{BT} \ge \boldsymbol{FR}$$

- Problem 2
  - ❖ Given a computing workflow, a cloud model with different types of VMs, and a cost objective function.
    - processing power (instructions/second)
    - storage access bandwidth (bytes/second)
    - charge rate ($/second)

VM Type	MIPS	Disk BW	Cost/hr
T1	1000	10	$0.10
T2	2000	100	$0.30
T3	3000	10	$0.30
T4	4000	100	$0.60

  - ❖ Objective: To assign workflow tasks to VM instances such that the total execution time and total cost are minimized.

- Problem 1
  - ❖ Module execution time

$$T_{exec}(m_i, v_k) = S_{v_k} * \left(\frac{C_{m_i}(\cdot) * \sum_{m_q \in pre(m_i)} z_{qi}}{p_k}\right)$$

  - ❖ Data transfer time

$$T_{trans}(z_{ij}, e_{ij}, l_{hk}) = S_{l_{hk}} * \frac{z_{ij}}{b_{hk}} + d_{hk}$$

  - ❖ Reliability and overall failure rate (OFR)

$$R_{f_h}(t) = e^{-\lambda_{f_h} \cdot t_{f_h}} \qquad R_{sd} = \prod_{h=0}^{N+|E_{cn}|-1} R_{f_h}(t) \qquad F_{OFR} = 1 - R_{sd}$$

  - ❖ Power consumption/cost model (DVFS)
    - Three power modes

$$P_j(t) = \begin{cases} \delta_j + \alpha_j f_j^3 & \text{active mode} \\ \delta_j & \text{idle mode} \\ << \delta_j & \text{sleep mode} \end{cases}$$

    - EC on a server

$$E_{v_h,Active} = \sum_{i=1}^{k} P_h * RT_{exec}(m_i, v_h)$$

    - TEC

$$TEC = \sum_{h=0}^{N-1} E_{v_h,Active}$$

  - ❖ Execution time, CPU frequency and reliability
    - Execution time at frequency f

$$T_{exec}(m_i, v_h, f) = T_{exec}(m_i, v_h, f_{h,max}) * \left(\beta_{cpu}\left(\frac{f_{h,max}}{f} - 1\right) + 1\right)$$

    - Execution time at frequency f under failure rate

$$RT_{exec}(m_i, v_h, f) = \frac{T_{exec}(m_i, v_h, f)}{R_{f_h}(t)}$$

    - Transfer time under failure rate

$$RT_{trans}(e_{ij}, l_{hk}) = \frac{T_{trans}(e_{ij}, l_{hk})}{R_{l_{hk}}(t)}$$

- Problem 2
  - ❖ Run time of each task (seconds)

$$cruntime_i = \frac{c_i}{p_{I(M(i))}}$$

  - ❖ I/O time of each task (seconds)

$$ciotime_i = \frac{\sum_k f_{ki} + \sum_j f_{i,j}}{b_{I(M(i))}}$$

  - ❖ Makespan Cost ($)

$$makespanCost = makespan \cdot costOfTime$$

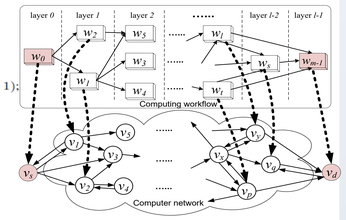
  - ❖ The total cost ($)

$$ccost = makespanCost + \sum_{V_n \in V} r_{I(n)} \cdot vmtime_n$$

TECHNICAL SOLUTIONS

Algorithm 1 LEES algorithm (G_w, G_{cn}, v_s, v_d)

```
pre(v_j): the set of preceding nodes of node v_j;
v(m_i): the node to which module m_i is mapped;
{f_s}: Discrete CPU frequency levels;
freqStep = 0.2;
counter = 0;
minTEC = 0.0;
for all node v_i ∈ G_cn do
    f_j,opt = ∛(δ_j / 2α_j);
Topologically sort workflow G_w into l layers: lay_k, k = 0, 1, ..., l − 1;
for all lay_k, k = 1, 2, .., l − 1 do
    for all unmapped module m_i ∈ lay_k do
        Select m_i with the highest computational requirements;
        Find pre(m_i) and V_mapping(pre(m_i))
        Find V_candidate(m_i);
        EC = MaxValue;
        for all (f_chosen ∈ {f_s})&(EC = MaxValue) do
            for all node v_j ∈ V_candidate(m_i) do
                f_chosen = f_j,opt + (freqStep * counter);
                T_j'(f_chosen) = T_j'(f_j,max) * (β_cpu (f_j,max / f_chosen − 1) + 1);
                if ParOFR_j ≤ OFR bound F then
                    if 1/BT_j ≥ FR bound FR then
                        E_vj = P_j * T_j'(f_chosen);
                        if E_vj ≤ EC then
                            EC = E_vj;
                            v(m_i) = j;
        if EC ≠ MaxValue then
            minTEC = minTEC + EC;
            counter + +;
return minTEC satisfying F_OFR ≤ F and 1/BT ≤ FR.
```



Algorithm 1 Breeze1 (workflow G, cloud C, cost CoT)

```
Sort tasks topologically
for all i in G.T do
    band ← set of tasks at same depth as T_i
    parallels ← cardinality of band
    timeCost ← CoT/parallels
    M_i, T_i.tasktime ← BestVMType(T_i, C, timeCost)
AssignTaskTimes(G.T)
cost ← CalcTotalCost(T, C, CoT)
return M, cost
```

Algorithm 2 Breeze2 (workflow G, cloud C, cost CoT)

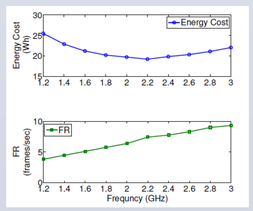
```
for all i in G.T do
    M_i, T_i.tasktime ← bestVMType(T_i, C, CoT)
Sort tasks topologically
Compute critical path using task times
AmortizeCost(G.T, CoT)
AssignTaskTimes(G.T)
cost ← CalcTotalCost(T, C, CoT)
return M, cost
```

Algorithm 3 Breeze3 (workflow G, cloud C, cost CoT)

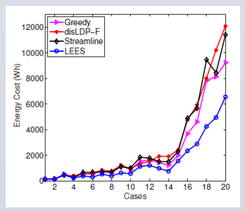
```
for all i in G.T do
    M_i, T_i.tasktime ← bestVMType(T_i, C, CoT)
Sort tasks topologically
Compute critical path using task times
AmortizeCost(G.T, CoT)
FindAssignments(G.T, 0, G.T.length)
AssignTaskTimes(G.T)
cost ← CalcTotalCost(T, C, CoT)
return M, cost
```

PERFORMANCE EVALUATION AND COMPARISON

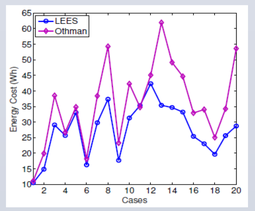
- Problem 1

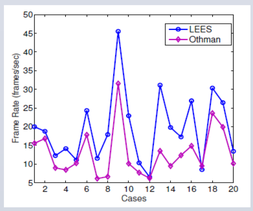


Energy and FR with the optimal frequency = 2.2 GHz



EC comparison of four algorithms

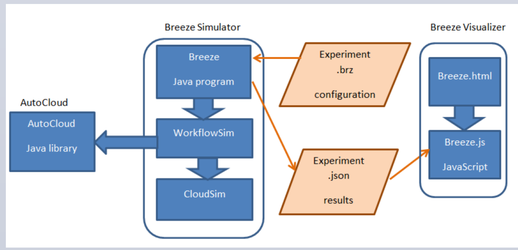


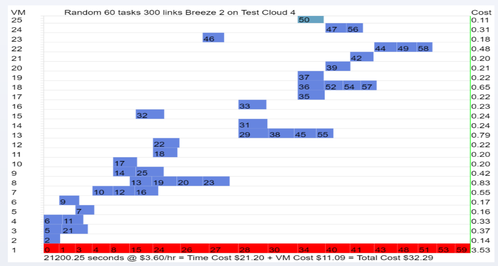


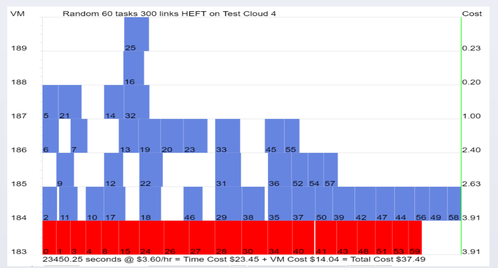
LEES vs Othman

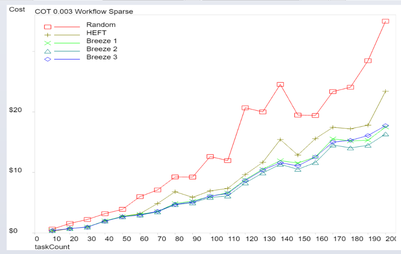
- Problem 2
  - ❖ Breeze logo

  - ❖ Breeze framework
    - AutoCloud Java library
      - ✓ decides a virtual machine instance type to run each task
    - Breeze Simulator utilizes two libraries
      - ✓ WorkflowSim
      - ✓ CloudSim
    - Breeze Visualizer
      - ✓ runs in the user's web browser
      - ✓ depicts the workflow structure statically and dynamically









CONCLUSION AND FUTURE WORK

- Tackled a tri-objective optimization problem
- Developed a Breeze simulator to illustrate the workflow execution using different scheduling algorithms
- To conduct more experiments using real-life large-scale application workflows in a real cloud environment
- To define more realistic cloud model and pricing structure
  - ❖ multi-core VMs
  - ❖ modeling local storage performance
  - ❖ allotting some time to provision a VM

SELECTED REFERENCES

- Othman, J.Nicod, L. Philippe, and V. Sonigo, "Optimal energy consumption and throughput for workflow applications on distributed architectures," J. of Sustainable Computing - Informatics and Systems, vol. 4, no. 1, pp. 55–74, 2014.
- G. Aupy, A. Benoit, and Y. Robert, "Energy-aware scheduling under reliability and makespan constraints," in Proc. of the 19th Int. Conf. on High Performance Computing, December 18-22 2012.
- Agarwalla, N. Ahmed, D. Hilley, and U. Ramachandran, "Streamline: a scheduling heuristic for streaming application on the grid," in Proc. of the 13th Multimedia Comp. and Net. Conf., San Jose, CA, 2006.
- Y. Gu, Q. Wu, X. Liu, and D. Yu, "Distributed throughput optimization for large-scale scientific workflows under fault-tolerance constraint," J. of Grid Computing, vol. 11, no. 3, pp. 361–379, 2013.
- A. K. Gupta, S. Saxena, and S. Khakhil, "A Journey towards Workflow Scheduling of Cloud Computing," International Journal of Computer Applications, vol. 123, no. 4, Aug 2015.
- E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: The montage example," in the proceedings of Int. Conf. for High Performance Computing, Networking, Storage and Analysis, SC 2008, NJ, 2008.

ACKNOWLEGEMENT

Most of the implementations are done by my Ph.D. students – Huda Khalid Al Rammah and Hugh Matlock

This research is sponsored by the National Science Foundation under Grant No. 1525537 with Middle Tennessee State University.