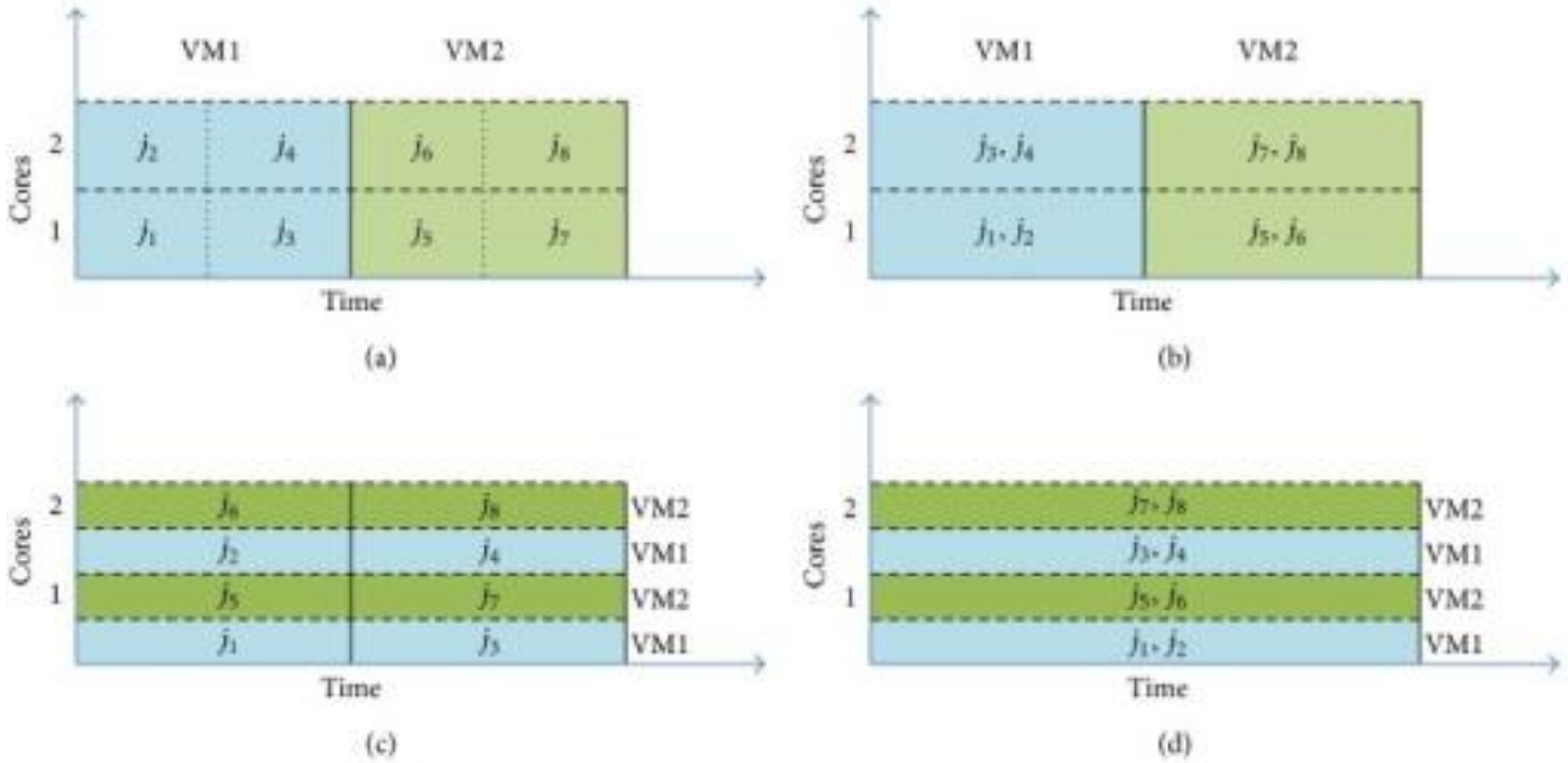
Task Scheduling and VM Provisioning

Resource (VM) Provisioning



(a) Space-share for VMs and jobs, (b) space-share for VMs and time-share for jobs,
(c) time-share for VMs and space-share for jobs, and (d) time-share for VMs and jobs.

Time Shared and Space Shared

- Space-sharing: The machine may be partitioned into sets of processors/cores (clusters). Each cluster is allocated to a single **job** that is allowed to Run To Completion (RTC).

- Time-sharing: More than one job may be allocated to a cluster where each **job** runs for some quantum of time before being preempted to allow other jobs to run.

Resource Provisioning Challenges

- Efficient resource management
- Dynamic allocation of VMs
- QoS meeting
- Minimize makespan (time required to complete group of jobs)
- Minimize energy consumption
- Minimize cost
- Maximize CSP profit etc.

- Performance of VMs- Not stable, e.g., 24 % variability on Amazon’s EC2 cloud.

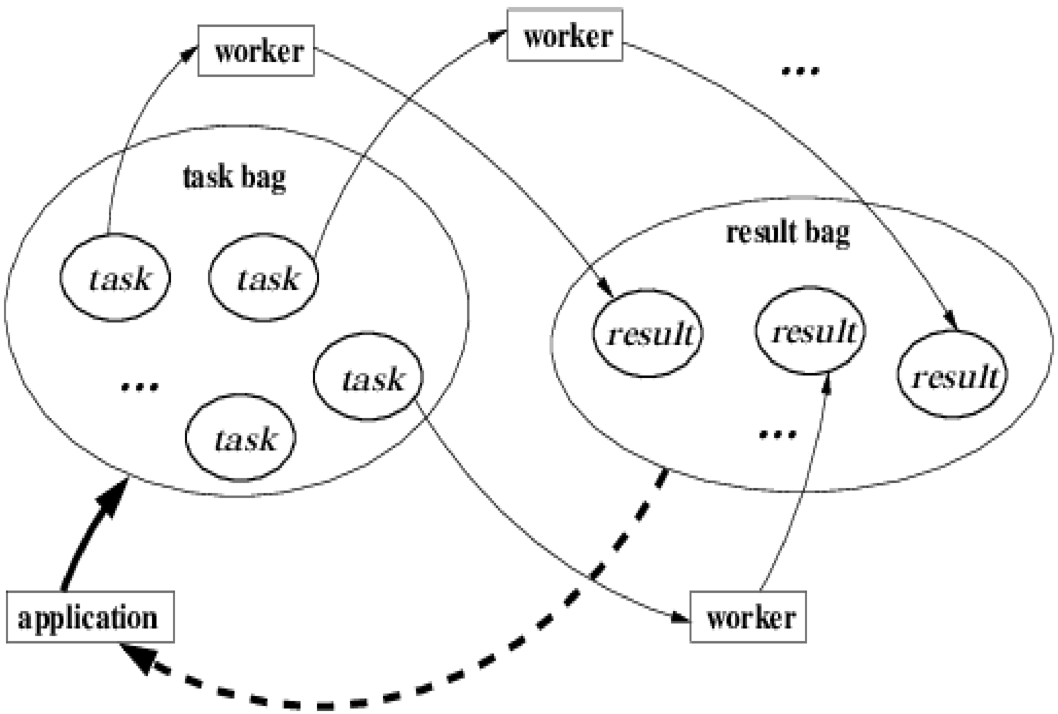
How to allocate task/jobs to VMs?

- VMs can be added/deleted/migrated to other machines
- VMs can be dynamically allocated
- Tasks are allocated to the VMs to run
- Tasks are run on VMs
- But how to allocate and in which order the tasks are executed on the VMs are the main concerns of cloud computing.

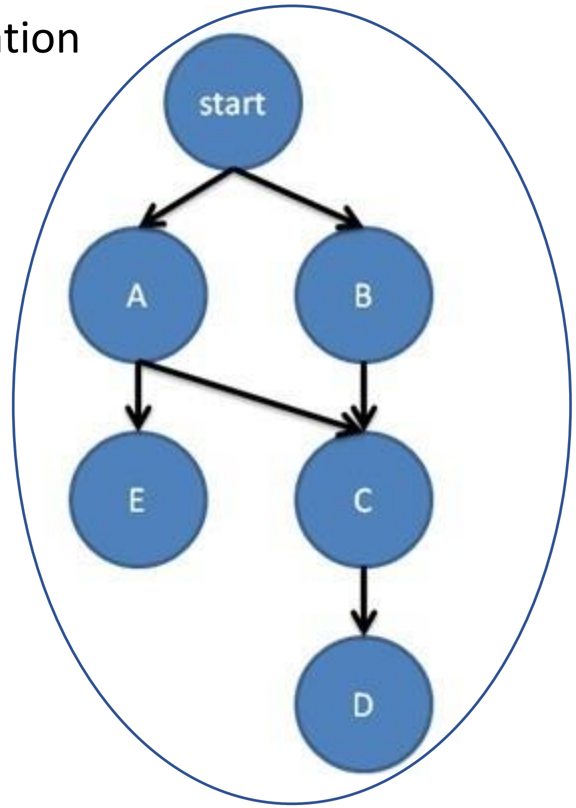
What is task/jobs

- Task is an atomic unit that is a part of an application.
- To run one application, we need to execute N number of tasks.

Task types: based on dependencies



Application



Independent task
(Bag of Task (BoT))

Dependent task or DAG or
workflow

Task types: based on leasing priorities

- Advance Reservation (AR)
- Best Effort (BE)
- Immediate Lease
- Best Effort with Deadline
- Negotiated Lease

Ref: http://haizea.cs.uchicago.edu/manual/node9.html

Lease assignment policies provided by various companies

Company name	Allocation Policy
AWS EC2	Best Effort
Nimbus	Immediate Lease
Eucalyptus	Immediate Lease
Open Nebula	Best Effort
Haizea	AR, BE, Immediate

Task Scheduling

- Applications are submitted to clouds
- An application is basically a workflow
- It consists of independent and dependent tasks
- An application can be represented as a DAG

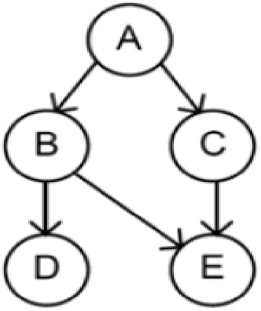


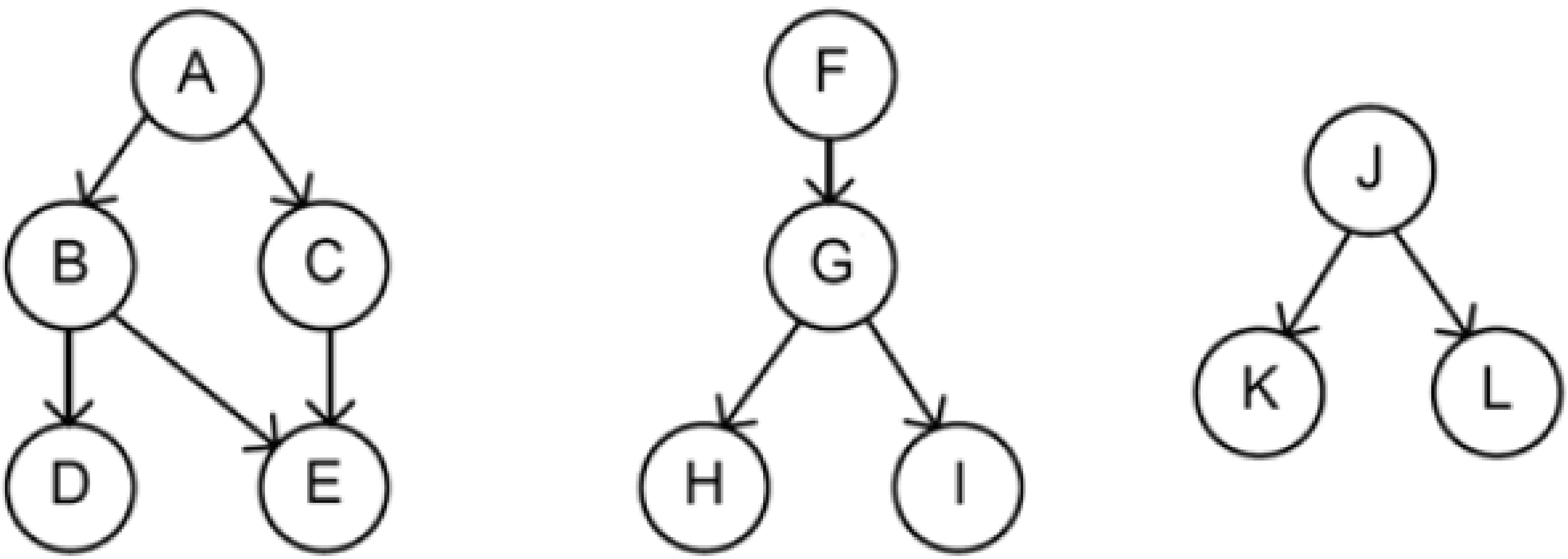
Fig. A Workflow. 1) A is the Parent task 2) B & C are child tasks 3) An arrow shows dependency

Task Scheduling

- Two main stages while planning execution of a workflow
  - Resource Provisioning: Computing resources (VMs) are selected to run the tasks
  - Scheduling: A schedule is generated by mapping the tasks to the best suited resources (VMs)

- Note: The selection of the resources and mapping of the tasks are done so that user defined QoS are met.

Application or Workflow

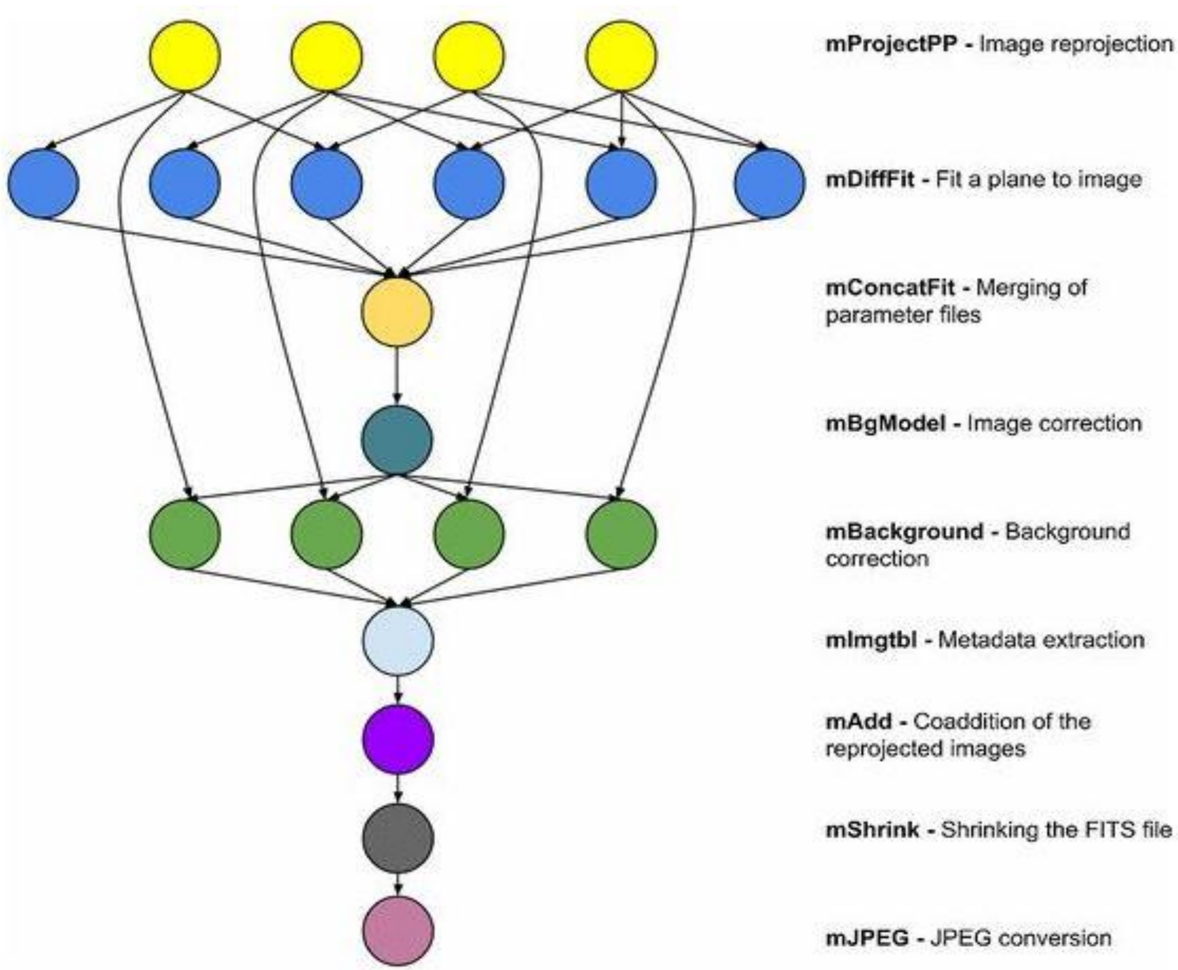


Application 1,
Arrival time: 0

Application 2,
Arrival time: 3

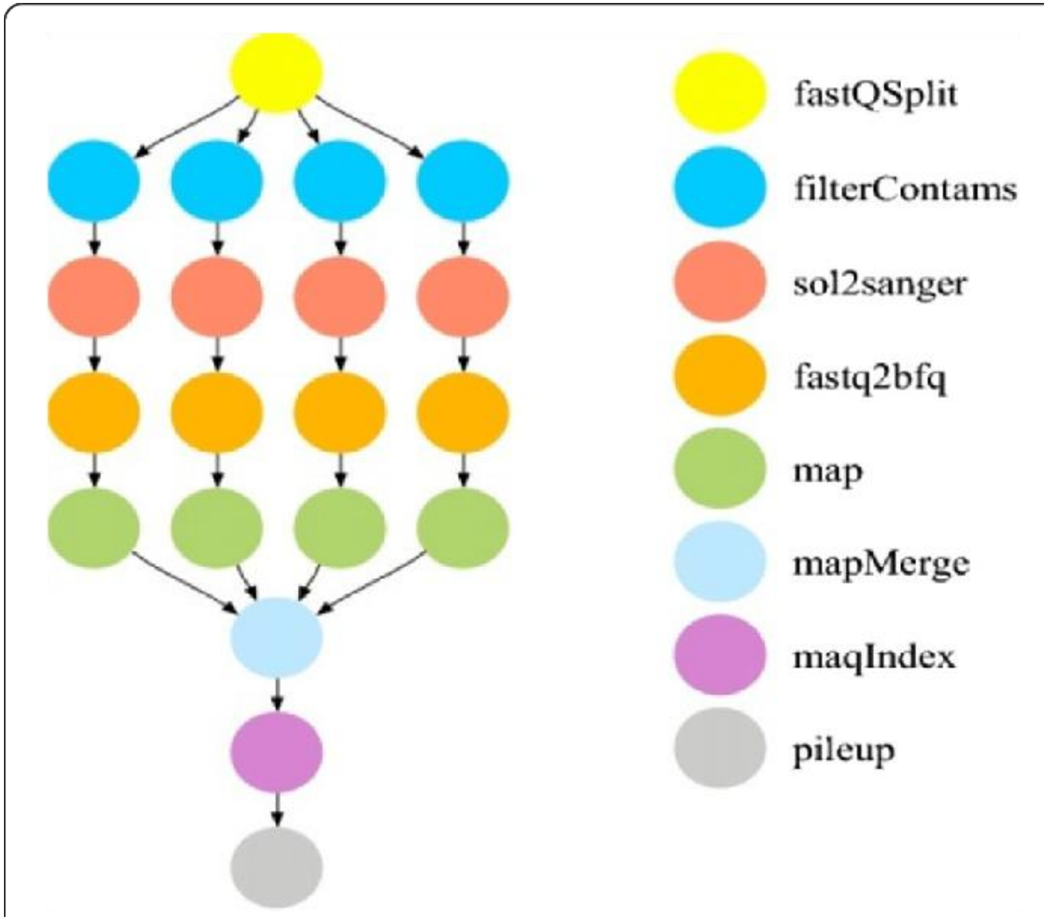
Application 3,
Arrival time: 9

Benchmark Workflow or Application



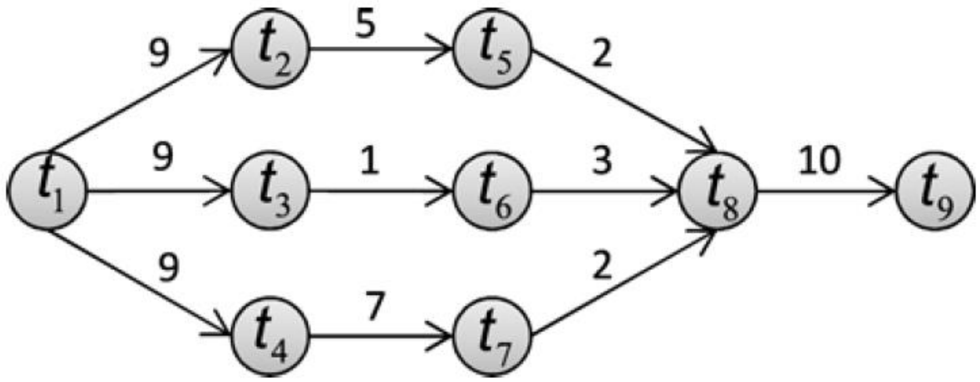
Montage Workflow: Astronomy Project

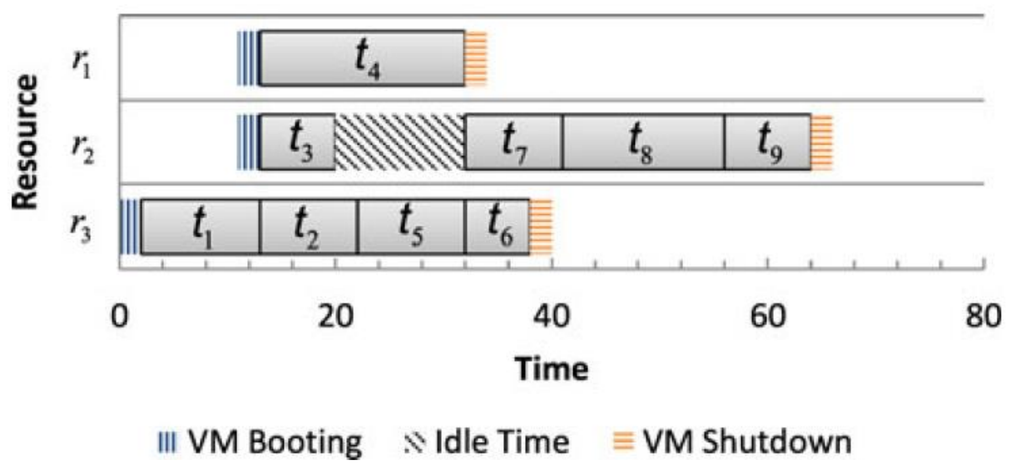
Benchmark Workflow or Application



Epigenomics workflow : Biology Project

Task Scheduling on a single cloud



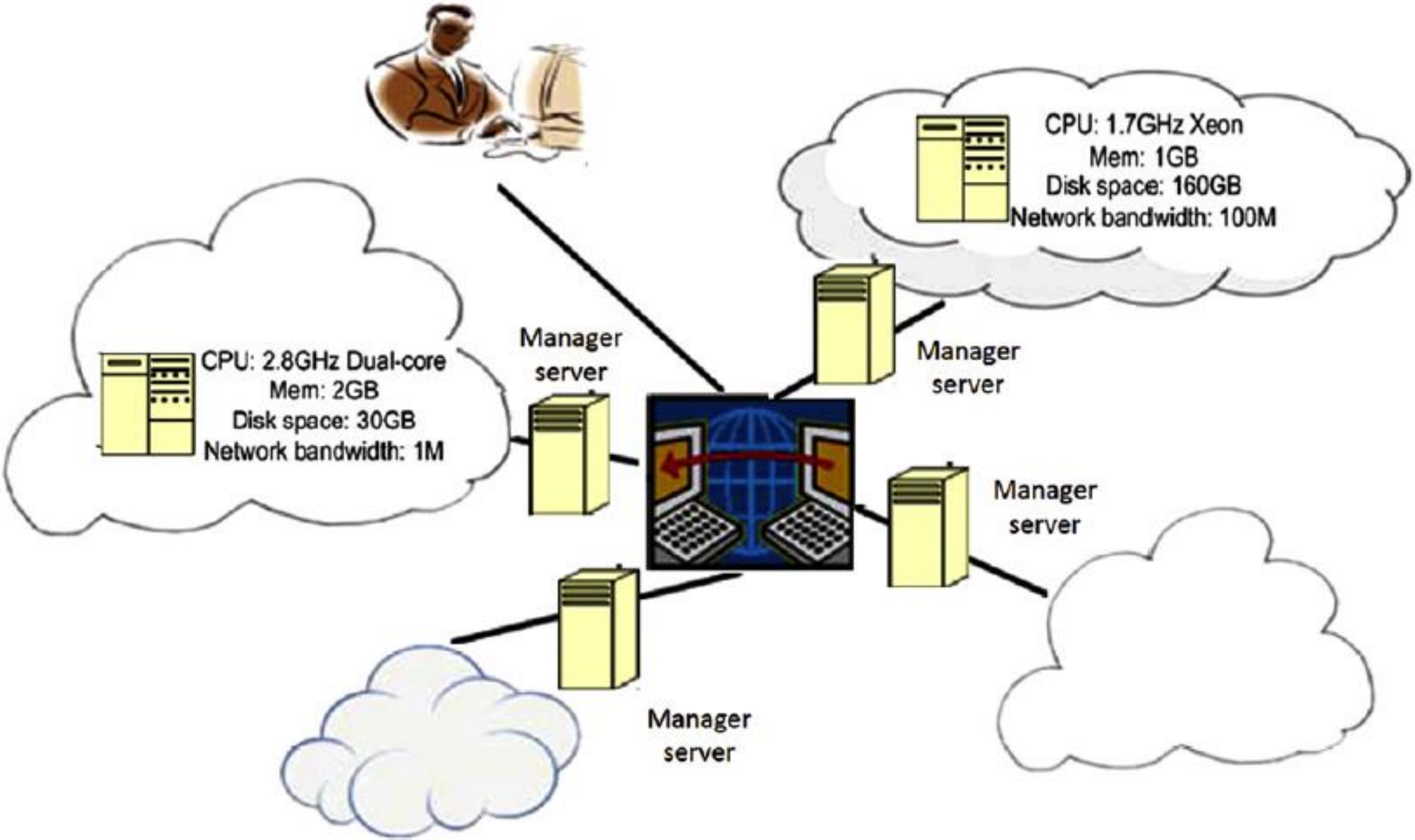


Need of Heterogeneous Multi-cloud

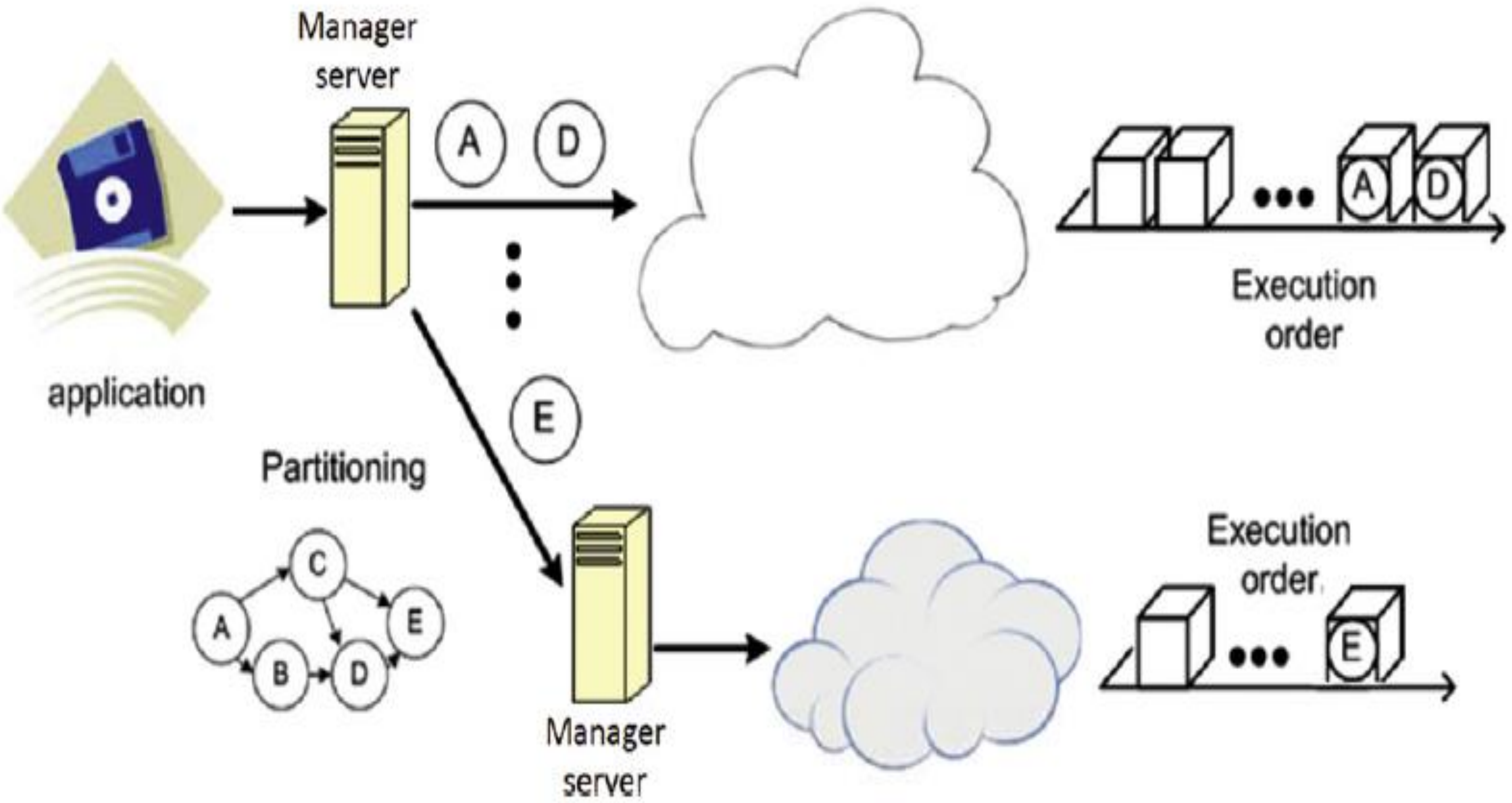
- It consists of multiple clouds with their own data centers.

- **Motivation 1:** No data center with unlimited resource capacity
- **Motivation 2:** In peak demand, some data centers may be overloaded
- **Motivation 3:** Workload can be shared among different data centers

Centralize Task Scheduling in Multi-cloud

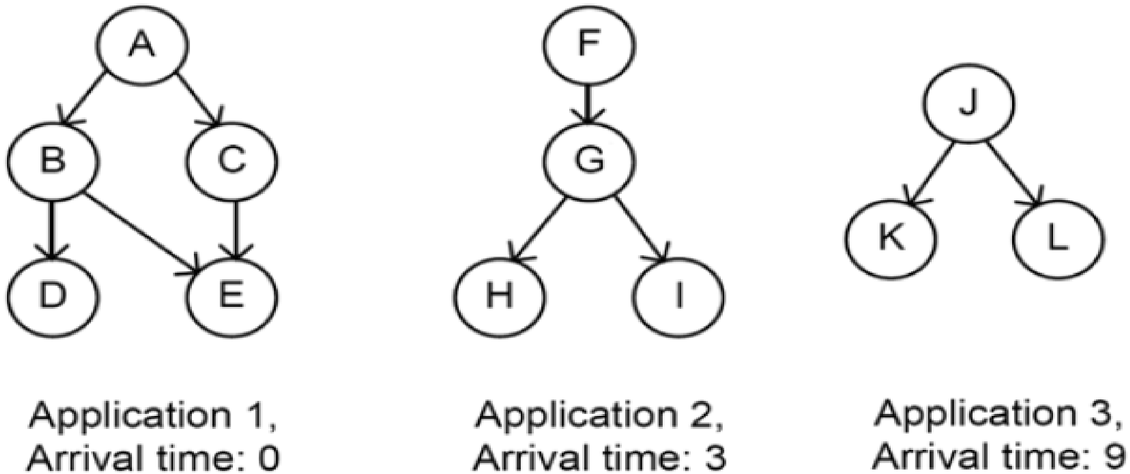


Distributed Task Scheduling in Multi-cloud



Task Scheduling Problem

- A Cloud Server $C = \{VM_1, VM_2, VM_3, \ldots, VM_m\}$ and a set of applications $A = \{A_1, A_2, A_3, \ldots, A_n\}$.
- Each A_i a DAG = (T_i, E_i) where $T_i = \{T_{i1}, T_{i2}, T_{i3}, \ldots, T_{ipi}\}$ is a set of p_i tasks and E_i denotes a set of links.
- An edge $E_{ijk} = (T_{ij} \rightarrow T_{ik}) \in E_i$ represents precedence such that task T_{ij} cannot start until T_{ik} is completed.



Task Scheduling Problem Cont…

- Note that $|A_i|$ may not be equal to $|A_j|$ and $\{A_i\} \cap \{A_j\} = \emptyset$ for $i \neq j$.

- Each application A_i has different arrival time and has some mode of execution AR or BE.

- No AR task can be preempted. However, Advance Reservation (AR) tasks can preempt any Best Efforts (BE) task.

- **The problem is to assign the tasks to cloud resources (VMs) to be executed by fulfilling some criteria e.g., minimum makespan, total cost and maximum cloud resource utilization etc.**

Expected Time to Compute (ETC) and Data Transfer Time (DT)

- Let $T = \{T_1, T_2, \ldots, T_n\}$ set of tasks.

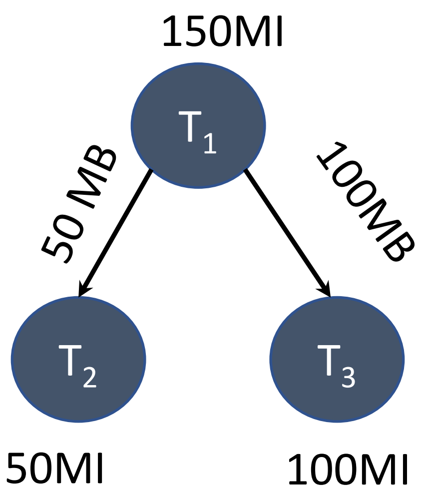
- Let $VM_j = \{P, B\}$ where, P is processing speed in **MIPS** and B is bandwidth in **MBPS**.

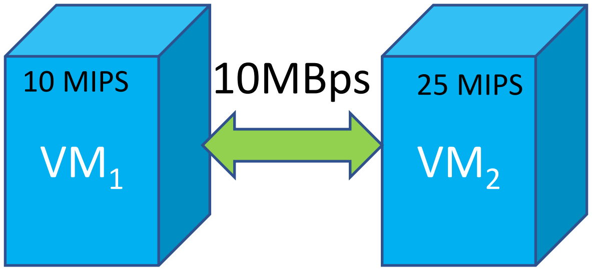
- Let $T_i = \{I, D\}$ be a task having set of instruction I in **MI** and data D in **MB**. Then the execution time of T_i on VM_j is expressed as follows:

$$ETC_{ij} = \frac{\boldsymbol{I}}{\boldsymbol{P}} \qquad \mathrm{DT} = \frac{\boldsymbol{MB}}{\boldsymbol{MBps}}$$

*** If T_i and T_j are allocate on same VM then data transfer time will be zero*

ETC and DT numerical problem





** If T_i and T_j *are allocate on same VM then data transfer time will be zero*

ETC Matrix

$$
\begin{array}{c}
 \\ T1 \\ T2 \\ T3
\end{array}
\begin{array}{c}
\begin{array}{cc} VM_1 & VM_2 \end{array} \\
\begin{pmatrix} 15 & 6 \\ 5 & 2 \\ 10 & 4 \end{pmatrix}
\end{array}
$$

DT Matrix

$$
\begin{array}{c}
 \\ T1 \\ T2 \\ T3
\end{array}
\begin{array}{c}
\begin{array}{ccc} T1 & T2 & T3 \end{array} \\
\begin{pmatrix} 0 & 5 & 10 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}
\end{array}
$$

Types of ETC matrix

Consistent

		R1	R2	R3	R4	R5
	U1	5	15	65	90	125
	U2	25	30	80	110	135
ETC	U3	35	45	90	135	160
	U4	65	75	125	155	180
	U5	78	89	130	160	200

Inconsistent

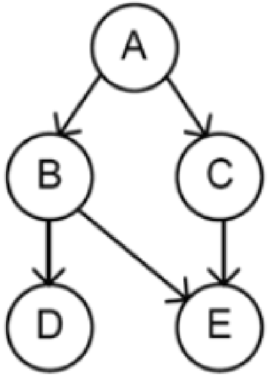
		R1	R2	R3	R4	R5
	U1	5	15	65	90	125
	U2	25	56	35	110	15
ETC	U3	60	35	90	135	160
	U4	65	45	35	155	25
	U5	78	48	130	160	200

Semi consistent

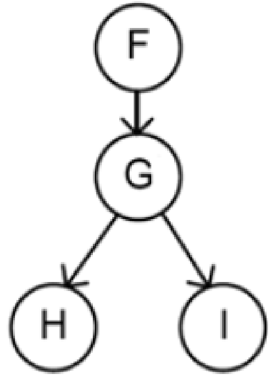
		R1	R2	R3	R4	R5
	U1	5	15	65	90	125
	U2	25	30	80	110	135
ETC	U3	35	45	90	135	160
	U4	65	35	43	45	322
	U5	78	65	30	14	32

ETC for the application provided in the DAG

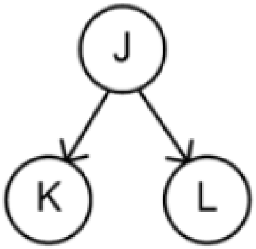
	A	B	C	D	E	F	G	H	I	J	K	L
VM_1	2	6	5	7	5	4	8	2	5	8	9	2
VM_2	3	8	3	10	9	2	8	3	4	4	3	3
VM_3	5	4	8	5	2	3	4	6	7	6	7	4



Application 1,
Arrival time: 0

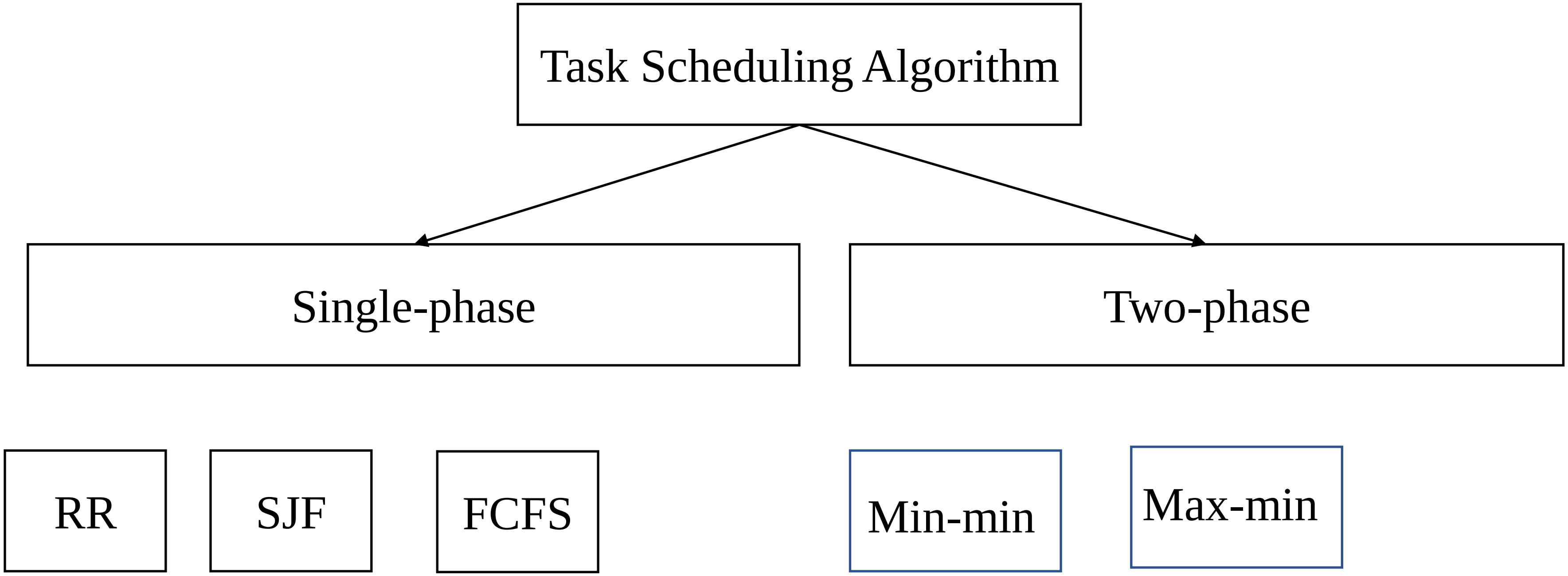


Application 2,
Arrival time: 3



Application 3,
Arrival time: 9

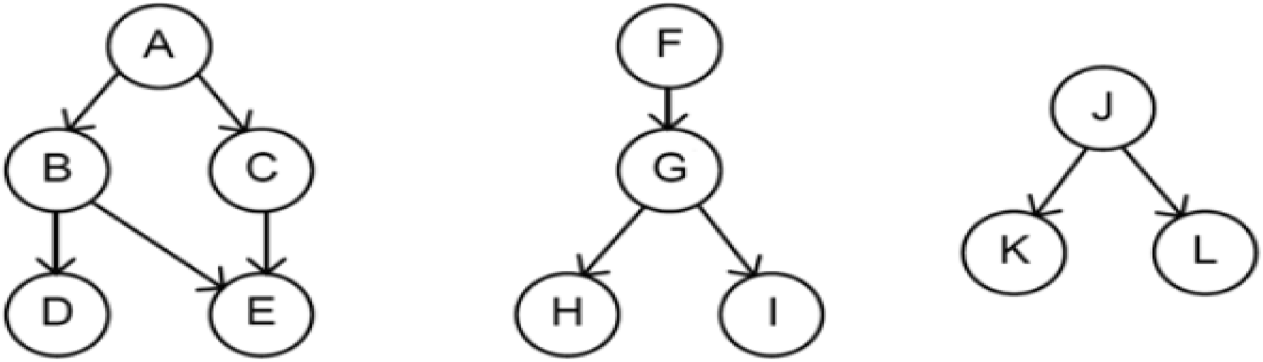
Basic Task Scheduling Algorithm



Note that: There may be other than these techniques

RR and SJF

Principle: Tasks are assigned to the VMs evenly across VMs.



Application 1, Arrival time: 0 — Application 2, Arrival time: 3 — Application 3, Arrival time: 9

BE — *AR* — *BE*

	A	B	C	D	E	F	G	H	I	J	K	L
VM_1	2	6	5	7	5	4	8	2	5	8	9	2
VM_2	3	8	3	10	9	2	8	3	4	4	3	3
VM_3	5	4	8	5	2	3	4	6	7	6	7	4

	A	B	C	D	E	F	G	H	I	J	K	L
RR	1	2	3	1	2	3	1	2	3	1	2	3
SJF	1	3	2	3	3	2	3	1	2	2	2	1

Min-Min and Max-Min

Min-Min

Task/VM	VM_1	VM_2
T_1	150	18
T_2	32	7
T_3	20	3
T_4	50	15

Task/VM	
T_1	18
T_2	7
T_3	3
T_4	15

Order of execution
T3 -> T2 -> T4 ->T1

Task/VM	VM_2
T_3	3

Max-Min

Task/VM	VM_1	VM_2
T_1	150	18
T_2	32	7
T_3	20	3
T_4	50	15

Task/VM	
T_1	18
T_2	7
T_3	3
T_4	15

Order of execution
T1 -> T4 -> T2 ->T3

Task/VM	VM_2
T_1	18

Priority assignment in DAG scheduling

- ❖ DAG scheduling follows two phases.

- ❖ First phase is known as task Prioritization phase. In which priority of each task is calculated.

  - ❖ There are various schemes for calculation the priority of tasks. e.g., T-level, B-level, ALAP, CP, FCFS, RR and so on.

- ❖ We will discuss ***B-level* and *T-level*** priority schemes only.

- ❖ **In the second phase**, the scheduling (task to VM mapping) takes place by calculating earliest start time (EST) and earliest finish time (EFT) of each prioritized task.

B-level Calculation and Formula

$$B-level(t_i) = ACC\ (t_i) + \max_{t_s \in\ set\ of\ immediate\ \ succ.(t_i)} \{TT\ (t_i, t_s) + B-level(t_s)\}$$

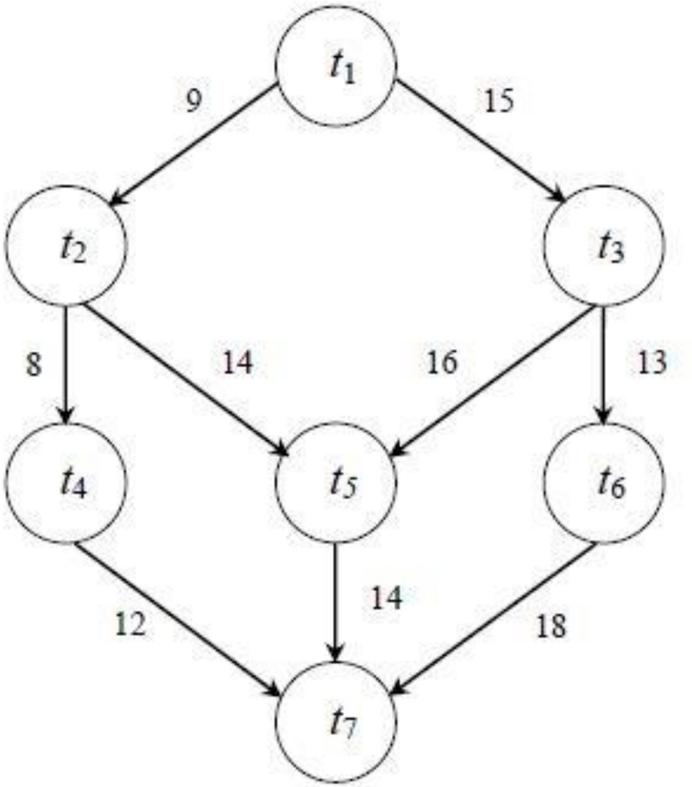
- Where, ACC(t_i) is the *average computation cost* or *average execution time* of task t_i as per the given *ETC*
- *TT(t_i, t_s) is the transfer time from task t_i to task t_s.*

- Do a bottom-up traversal of the DAG and find *B-level* priority to each task (t_{i})

- Arrange the list in non-increasing order based on B-level priority that will be the priority list for the scheduling.

B-level Example

Let consider the DAG and corresponding ETC matrix,



DAG of 7 tasks

ETC matrix

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
VM_1	10	11	5	11	9	5	6
VM_2	8	11	11	6	11	11	8
VM_3	10	5	7	11	10	11	9

Phase I

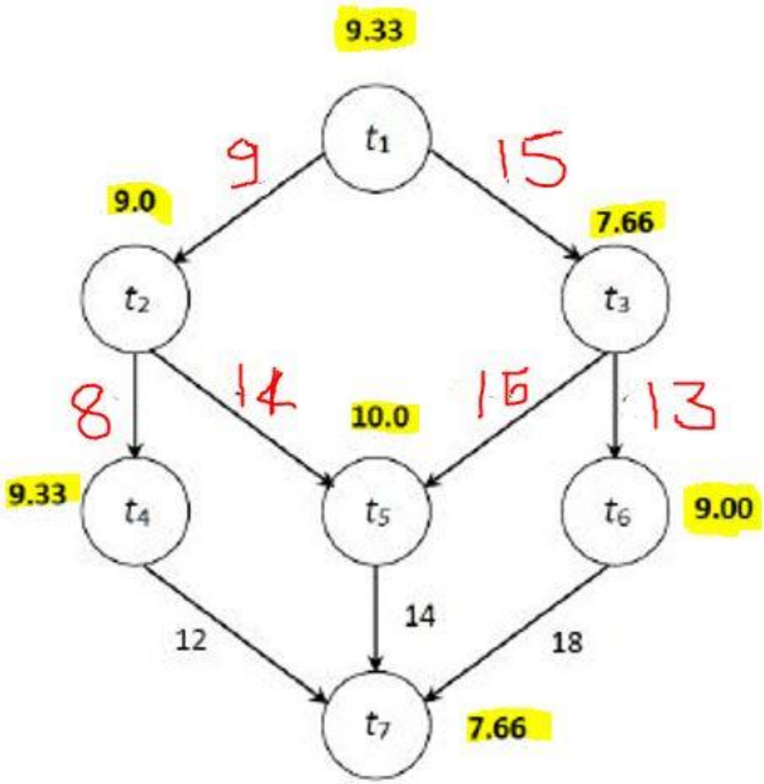
1. Calculate the ACC of each tasks on available 3 VMs as:
   (This step is compulsory for both B-level and T-level calculation)

Task(*i*)	**t1**	**t2**	**t3**	**t4**	**t5**	**t6**	**t7**
ACC (*ti*)	9.33	9.00	7.66	9.33	10	9.00	7.66

- Then traverse the DAG as per the priority scheme (i.e, Top down, bottom up)

- **See the Illustration**

Example



By traversing DAG upwards from exit task node i.e., t_7

Therefore,

B-level (*t7*) = *ACC(t7)=7.66*

B-level (*t6*) = *(9+18+7.66)=34.66*

B-level (*t5*) = *(10+14+7.66)=31.66*

B-level (*t4*) = *(9.33+12+7.66)=28.99*

B-level (*t3*) = *(7.66+16+blevel of t5)= 23.66+31.66 =55.32*

= (7.66+13+blevel of t6)= 20.66+34.66 =55.32

(both the values are equal so 55.32 will be the value of B-level (*t3*) *)*

B-level (*t2*) = *(9+8+blevel of t4)= 17+28.99 =45.99*

= (9+14+blevel of t5)= 23+31.66 =54.66

(it is maximum so 54.66 will be the value of B-level (*t2*) *)*

B-level (*t1*) = *(9.33+9+blevel of t2)= 18.33+54.66 =72.99*

= (9.33+15+blevel of t3)= 24.33+55.32 =79.65

(it is maximum so 79.65 will be the value of B-level (*t1*) *)*

Therefore the ***B-level*** of each task will be as:

Task(i)	t1	t2	t3	t4	t5	t6	t7
B-level (*ti*)	79.65	54.66	55.32	28.99	31.66	34.66	7.66

Cont...

Task(i)	t1	t2	t3	t4	t5	t6	t7
B-level (*ti*)	79.65	54.66	55.32	28.99	31.66	34.66	7.66

- By arranging the B-level of all tasks in non-increasing order then the priority list will be

t1	t3	t2	t6	t5	t4	t7

- After that task-VM mapping will be done in second phase by calculating EST and EFT of each task node of the given DAG

T-level

- Do a Top-down traversal of the DAG and assign *T-level* priority to each task (t_i) by following the given formula as given below.

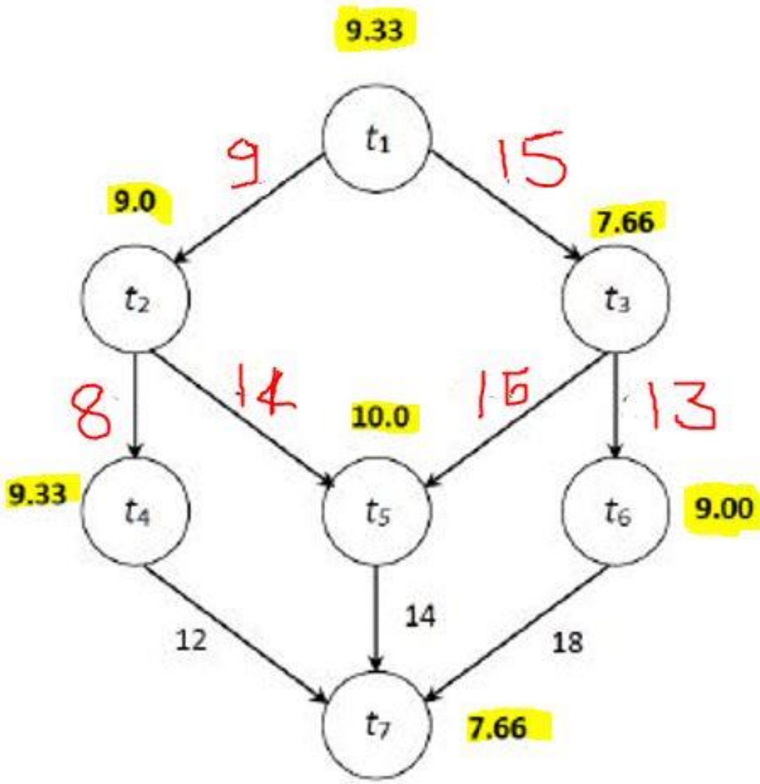
$$T - level(t_i) = Max\{ACC\ (t_p) + TT\ (t_p, t_i) + T - level(t_p)\}$$

Where, t_p is the set of immediate predecessors of t_i.

- Where, ACC(t_i) is the *average computation cost* or *average execution time* of task t_i as per the given *ETC*.
- *TT(t_p, t_i) is the transfer time from task t_p to task t_i. .*

- Arrange a list in increasing order of T-level priority for the scheduling

T-level Calculation example



By traversing DAG downwards from entry to exit task node i.e., t_1

Therefore,

T-level (*t1*) = *0+0 (because it has no predecessor task)*

T-level (*t2*) = *(9.33+9+0)=18.33*

T-level (*t3*) = *(9.33+15+0)=24.33*

T-level (*t4*) = *(**9**+8+18.33)=35.33*

T-level (*t5*) = *(**9**+14+18.33)= 41.33*

*=(**7.66**+16+24.33) = 47.99*

(because 47.99 is the maximum)

T-level (*t6*) = *(**7.66**+13+24.33)= 44.99*

T-level (*t7*) = *(9.33+12+T-level of t4)= 21.33+35.33 =56.66*

= (10+14+T-level of t5)= 14 +47.99 =61.99

= (9+18+T-level of t6)= 27+44.99 =71.99

Therefore the ***T-level*** of each task will be as in table.

Task(i)	t1	t2	t3	t4	t5	t6	t7
T-level (*ti*)	0	18.33	24.33	35.33	47.99	44.99	71.99

- **The task prioritization list will be according to increasing order of T-level**

Phase 2 Task-VM mapping

- Suppose the task priority ordering (by using any priority scheme) is given as:
  t1, t3, t2, t6,t5, t4, and t7
- Task –VM mapping using the following formulas:

I. $EST(t_i, VM_j)$=0 (Only for the entry task)

II. $EST(t_i, VM_j) = \max\{ \underset{j\in m\ (VM\ type\ )}{avail\ [j]}\ ,\ \max_{t_p\in pred(t_i)}(AFT(t_p) + TT_{p,i})\}$

Where, EST is the earliest start time of task t_i , t_p is the predecessor tasks set of t_i and $TT_{p,i}$ is the transfer time from t_p to t_i .

III. EFT is the earliest finish time which is defined as:

$$EFT(t_i, VM_j) = ETC(t_i, VMj) + EST(t_i, VM_j)$$

- **Note: here, TT(t_i,t_j) is represented by $C_{p,i}$**
- *TT* and *C* can be used interchangeably i.e., transfer or communication time.

Example

First task t_1 is selected so $EST(t_1)$ will be zero for all the 3 VMs i.e. $EST(t_1, VM_1)$ =0, $EST(t_1, VM_2)$=0 and $EST(t_1, VM_3)$=0.

Now we find $EFT(t_1)$ on all the VMs as follows.
$EFT(t_1, VM_1)$ = 0+10 = 10
$EFT(t_1, VM_2)$= 0+8 = 8 (it is best suited VM)
$EFT(t_1, VM_3)$= 0+10 =10

Next, the task t_3 is selected as per the task ordering list in phase 1. Therefore,
$EST(t_3, VM_1)$= max{0, 8+15 }=23
$EST(t_3, VM_2)$ =max{8, 8+0 }=8 (because t_1 is running on same VM so communication time will be zero)
$EST(t_3, VM_3)$= max{0, 8+15 }=23

Solved Example

Now, we calculate the EFT for t_3 corresponding to EST
$EFT(t_3, VM_1)$= 23+5=28
$EFT(t_3, VM_2)$ = 8+11=19 (again it is best suited VM for t_3)
$EFT(t_3, VM_3)$= 23+7=30

Finally, the task t_3 schedules on VM_2
In similar manner, we will find the best suited VM for the given tasks by considering task dependencies.

Example

	VM1		VM2		VM3		
	EST	EFT	EST	EFT	EST	EFT	VMID
T1	0	10	**0**	**8**	0	10	2
T2	17	28	19	30	**17**	**22**	3
T3	23	28	8	19	23	30	2
T4	45	56	30	36	22	33	3
T5	36	45	36	47	35	45	1
T6	32	37	19	30	32	43	2
T7	48	54	59	67	59	68	1

Makespan and Resource utilization formula

- The overall workflow processing time known as the ***makespan*** which is described as follow:

$$makespan = \min\{EFT(t_{exit})\}$$

- The utilization of a VM is the ratio of actual working time of that VM and the overall makespan of the cloud server where the VM is deployed. Mathematically,

$$U(VM_j) = \frac{working\ time\ (VM_j)}{Makespan} * 100$$

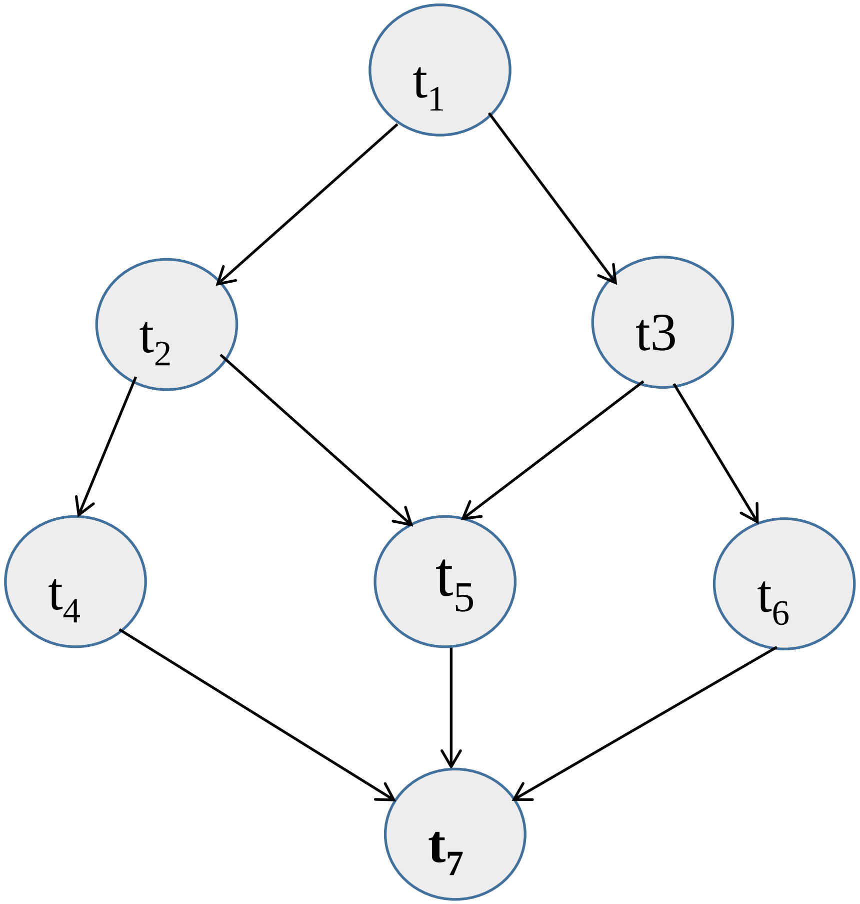
It is VM Utilization

- *Average Utilization = (Sum of all VM Utilization/ total number of VM)*

- *For the given example, U(VM1)= (9+6)/54 = 0.277*100=27.7%*
- *U(VM2)= (8+11+11)/54=0.555*100=55.5%*
- *U(VM2)= (5+11)/54=0.2962*100=29.62%*

- *Average Utilization = (27.7+55.5+29.62)/3 =37.606%*

Example



Consider the DAG having 7 tasks and cloud server of 3 VMs as per the given table.
Apply FCFS, RR, SJF, scheduling techniques and compare the makespan and **average resource utilization** of each scheduling scheme.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7
VM_1	4	3	10	8	12	6	7
VM_2	6	9	8	8	10	8	5
VM_3	9	8	9	7	9	9	9

Solution:

1. Note:
In the given DAG/Workflow no data transfer time is given so we need to consider only task dependency (i.e., child-parent dependency of tasks)

2. Here, we are following minimum execution time strategy at each level of DAG as the ordering or prioritization phase.

Hence, as per the minimum execution time (MET or SJF) following will be the task-VM Mapping.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7
VM1	**4**	**3**				**6**	
VM2			**8**				**5**
VM3				**7**	**9**		

Mapping & Scheduling

For

$T_1 \rightarrow VM_1$

$EST(T_1, VM_1) = 0$

$EFT(T_1, VM_1) = 0 + 4 = 4$

Note: No Data transfer time is considered in the given DAG. Therefore, EST and EFT will be calculated simply.

$T_2 \rightarrow VM_1$

$EST(T_2, VM_1) = 0+4 = 4$

$EFT(T_2, VM_1) = 4+3 = 7$

$T_3 \rightarrow VM_2$

$EST(T_3, VM_2) = 0+4 = 4$

$EFT(T_3, VM_2) = 4+8 = 12$

$T_4 \rightarrow VM_3$

$EST = 0+7 = 7$

$EFT = 7+7 = 14$

$T_5 \rightarrow VM_3$

$EST(T_5, VM_3) = 0+12$ ✗

$= 0+7+7 = 14$ ✓

$EFT(T_5, VM_3) = 14+9 = 23$

U ⇒ utilization

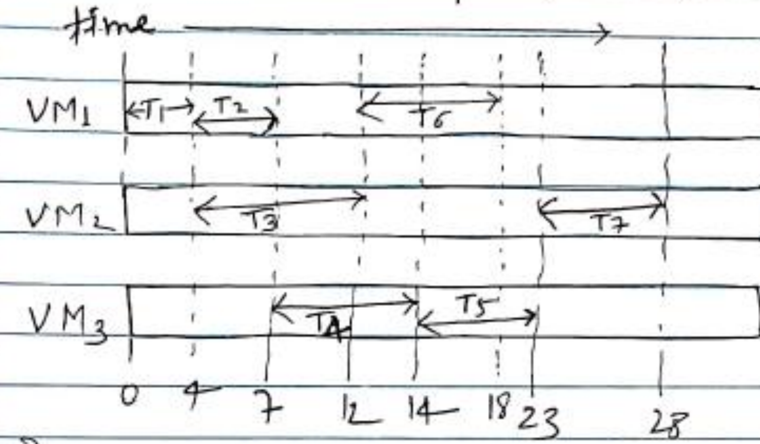
Makespan = 28

$U_{VM_1} = (7+6)/28$

$U_{VM_2} = (8+5)/28$

$U_{VM_3} = (7+9)/28$

Gantt chart



⇒

For $T_6 \rightarrow VM_1$

$EST(T_6, VM_1) = 0+12 = 12$

$EFT(T_6, VM_1) = 12+6 = 18$

For $T_7 \rightarrow VM_2$

$EST(T_7, VM_2) = 0+12 = 12$ ✗ discarded

$= 0+23 = 23$ ✓ Max

$= 0+18 = 18$ ✗ discarded

$EFT(T_7, VM_2) = 23+5 = 28$

which is makespan

Calculations:

Overall DAG processing time is called makespan which is calculated as **28** unit of time.

Now we will calculate the Utilization of each VM i.e.,

1. U(VM1)= (4+3+6)/28= 13/ 28 = **0.464**
2. U(VM2)= (8+5)/ 28 = 13/ 28 = **0.464**
3. U(VM3)= (7+9)/ 28 = 16/ 28 = **0.571**

Average cloud resource utilization = U(VM1+VM2+VM3)/3
= (0.464+ 0.464+ 0.571)/3
= 0.4996
Average cloud resource utilization in % = .4996*100= 49.96%.