FT-VMP: FAULT-TOLERANT VIRTUAL MACHINE PLACEMENT IN CLOUD DATA CENTERS

CHRISTOPHER GONZALEZ AND BIN TANG

DEPARTMENT OF COMPUTER SCIENCE

CALIFORNIA STATE UNIVERSITY DOMINGUEZ HILLS

OUTLINE

- Fault Tolerant VM Placement Problem (FT-VMP)
- Feasibility problem of FT-VMP using maximum flow
- Algorithms for FT-VMP
  - Optimal: integer linear programming (ILP)
  - Heuristics: VM replica placement and Migration
- Performance evaluation
- Conclusion and future work

CHALLENGES OF VM REPLICATION

- VM replication is an effective technique to achieve fault tolerance and reduce cloud user access latencies. A few challenges:
  - Fault tolerance constraint of VMs: multiple replica copies of the same VM application are placed into different physical machines (PMs).
  - Resource capacity constraint of PMs: each PM has limited cloud resources including CPUs, storages and memories.
  - Compatibility constraint of VMs to PMs: not all VM replica copies can be placed onto all PMs due to software/platform incompatibility

PROBLEM STATEMENT OF FT-VMP

- A set of VM applications, referred to as original VMs, have already been created and placed inside some PMs of the cloud data center.
- The fault-tolerance SLA requires that a number of replica copies (referred to as VM replicas) to be made and placed into the cloud data center.
- The goal of FT-VMP is to place VM replica copies into PMs while
  - satisfying constraints of fault-tolerance, resource capacity, and compatibility, and
  - minimizing number of active PMs.

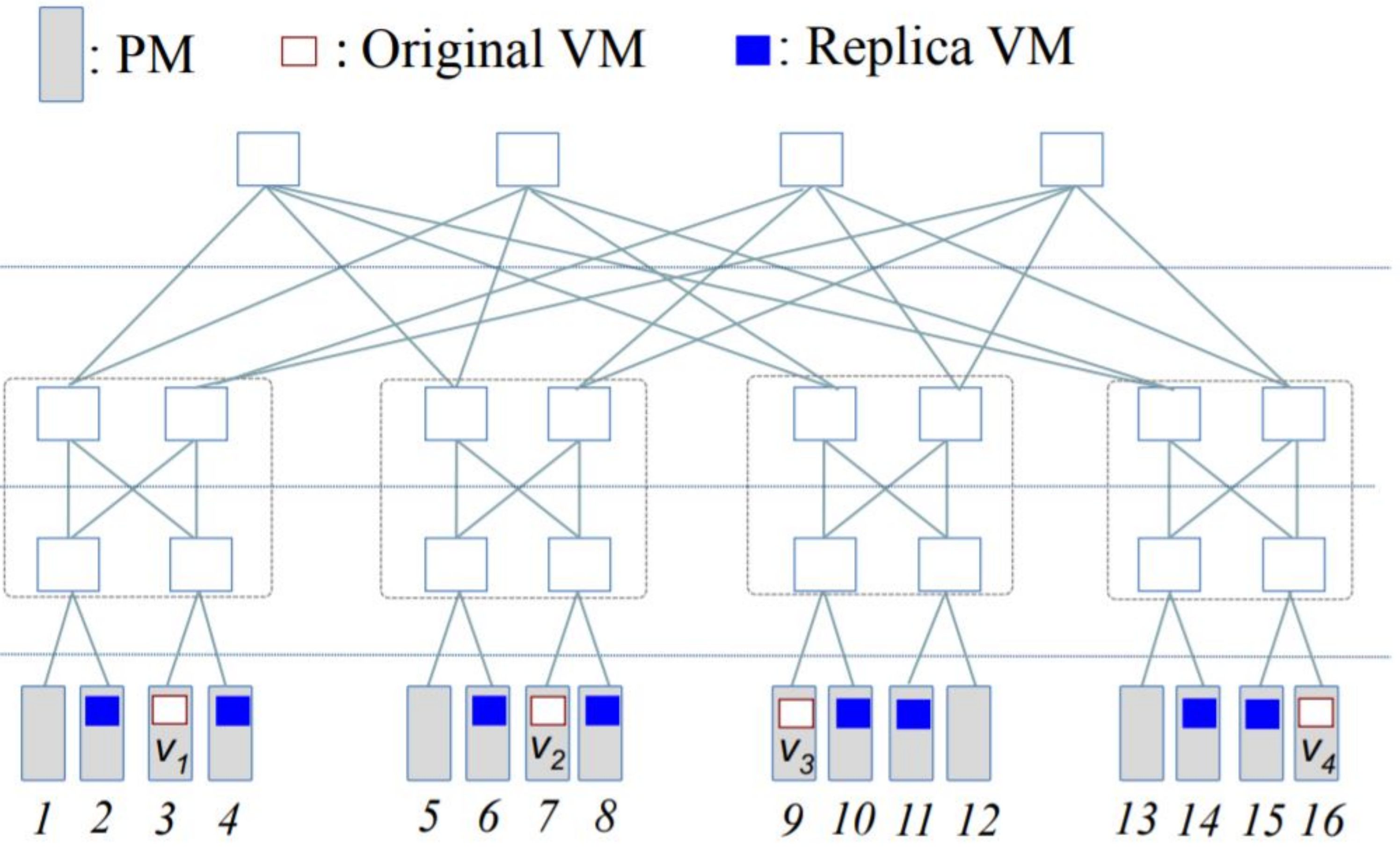


Fig. 1. A $k = 4$ fat tree topology. $l = 4$ original VMs: (v_1, v_2, ..., v_4) are located at PM 3, 7, 9, 16, respectively. $r_j = 2$.

THEOREM 1: FT-VMP IS NP-HARD

- Reduce vertex coloring (VC) problem, which is NP-hard, to a special case of FT-VMP.
- Cannot reduce from seemly similar bin packing problem:
  - All VMs and all their replicas are unit size

FEASIBILITY PROBLEM OF FT-VMP

- Question: Given any instance of FT-VMP, is it possible to place all the replica copies into the cloud data center to satisfy the fault-tolerance, resource capacity, and compatibility constraints?
- Solution: a maximum-flow based technique on a flow network transformed from the data center network.

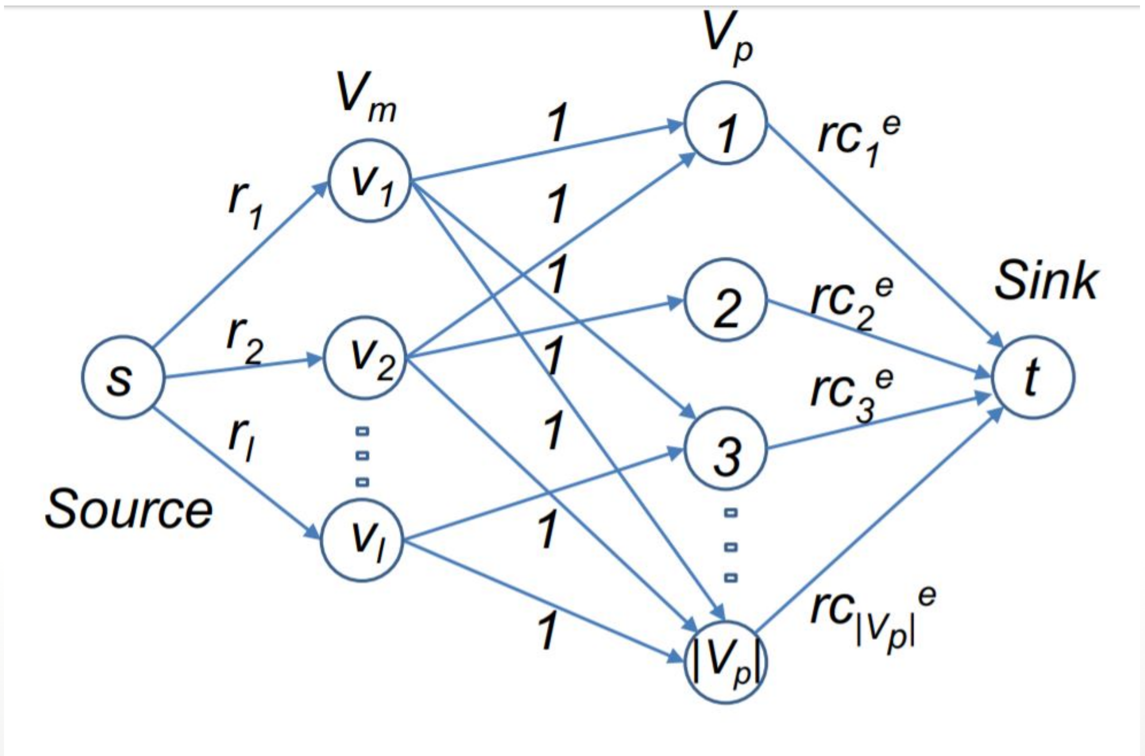


Fig. 3. Flow network $G'(V', E')$ to check the feasibility of FT-VMP. The value on each edge is its capacity. Note there is no edge between $v_j \in V_m$ and $i \in V_p - \mathcal{C}(j)$ if PM i is not in v_j's compatibility set $\mathcal{C}(j)$.

FEASIBILITY OF FT-VMP IS EQUIVALENT TO MAX FLOW PROBLEM

TIME COMPLEXITY: $O((L+|V_P|)^2 L |V_P|)$

LINEAR PROGRAMMING FORMULATION FOR FT-VMP

$$\min \sum_{i=1}^{|V_p|} y_i \tag{1}$$

s.t.

$$y_i = 0, 1 \quad \forall i \in V_p \tag{2}$$

$$x_{i,j,k} = 0, 1 \quad \forall i \in V_p, 1 \leq j \leq l, 1 \leq k \leq r_j \tag{3}$$

$$y_i = 1, \quad \forall i \in V_d \tag{4}$$

$$\sum_{i \in V_p} x_{i,j,k} = 1, \quad \forall 1 \leq j \leq l, 1 \leq k \leq r_j \tag{5}$$

$$y_i \geq x_{i,j,k}, \quad \forall i \in V_p - V_d, 1 \leq j \leq l, 1 \leq k \leq r_j \tag{6}$$

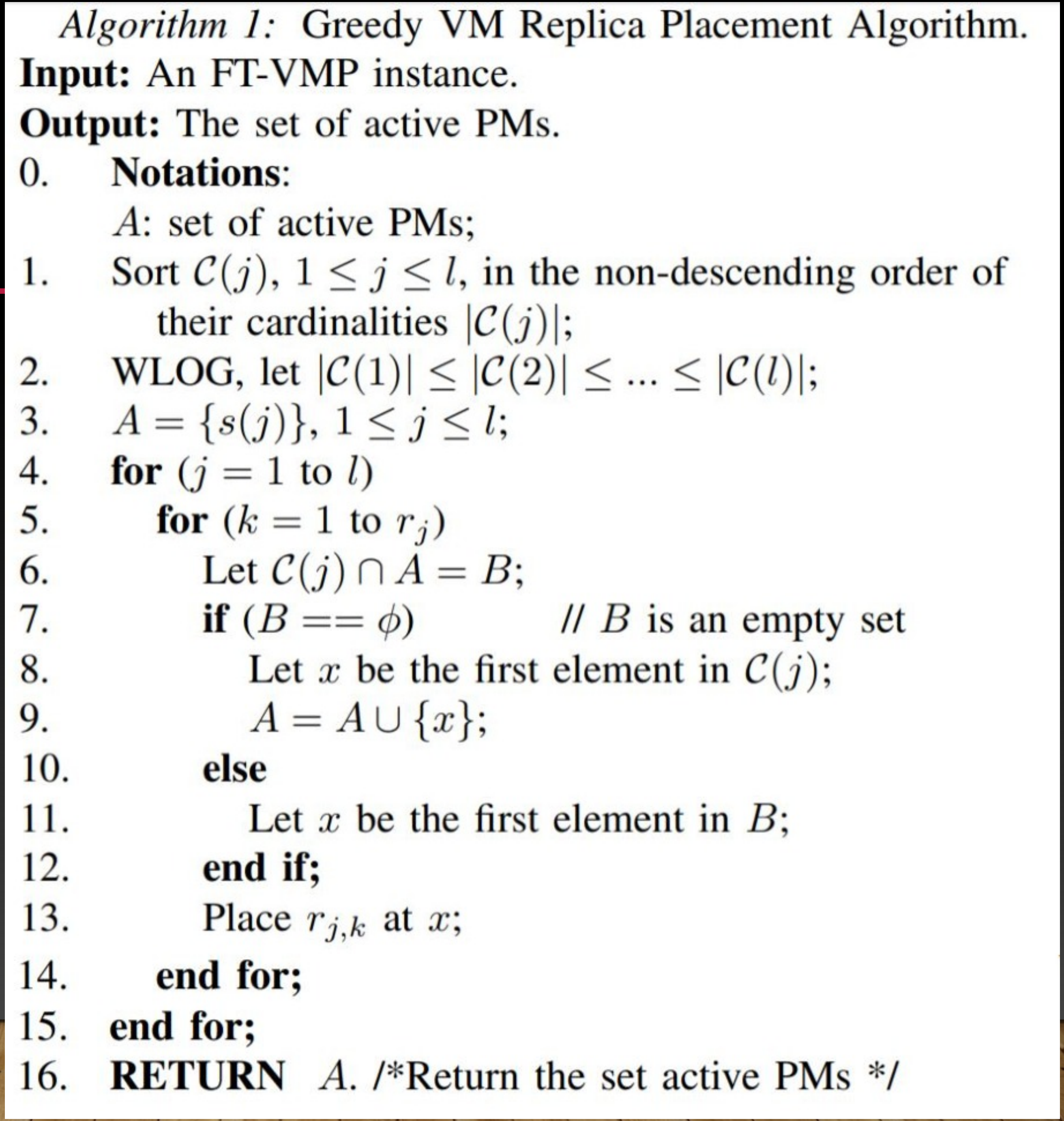
$$y_i * m_i^e \geq \sum_{j=1}^{l} \sum_{k=1}^{r_j} x_{i,j,k}, \quad \forall i \in V_p \tag{7}$$

$$\sum_{k=1}^{r_j} x_{i,j,k} \leq 1, \quad \forall i \in V_p, 1 \leq j \leq l \tag{8}$$

$$x_{i,j,k} = 0, \quad \forall 1 \leq j \leq l, 1 \leq k \leq r_j, i \in V_p - \mathcal{C}(j) \tag{9}$$

ALGORITHM 1:VM REPLICA PLACEMENT

TIME COMPLEXITY: O(L (LOG L + R))



Algorithm 1: Greedy VM Replica Placement Algorithm.

Input: An FT-VMP instance.

Output: The set of active PMs.

```
0.  Notations:
    $A$: set of active PMs;
1.  Sort $\mathcal{C}(j)$, $1 \leq j \leq l$, in the non-descending order of
      their cardinalities $|\mathcal{C}(j)|$;
2.  WLOG, let $|\mathcal{C}(1)| \leq |\mathcal{C}(2)| \leq ... \leq |\mathcal{C}(l)|$;
3.  $A = \{s(j)\}$, $1 \leq j \leq l$;
4.  for ($j = 1$ to $l$)
5.    for ($k = 1$ to $r_j$)
6.      Let $\mathcal{C}(j) \cap A = B$;
7.      if ($B == \phi$)          // $B$ is an empty set
8.        Let $x$ be the first element in $\mathcal{C}(j)$;
9.        $A = A \cup \{x\}$;
10.     else
11.       Let $x$ be the first element in $B$;
12.     end if;
13.     Place $r_{j,k}$ at $x$;
14.   end for;
15. end for;
16. RETURN  $A$. /*Return the set active PMs */
```

TWO DEFINITIONS IN VM REPLICA MIGRATION ALGORITHM

- Target Physical Machine (TPM) of replica VM $v_{j,k}$ is a PM that $v_{j,k}$ can be possibly moved to.
- Consolidating Physical Machines (CPMs) are PMs that can be potentially turned off and made inactive.

SERVER CONSOLIDATION ALGORITHM BY KHANI ET AL., ICC 2016

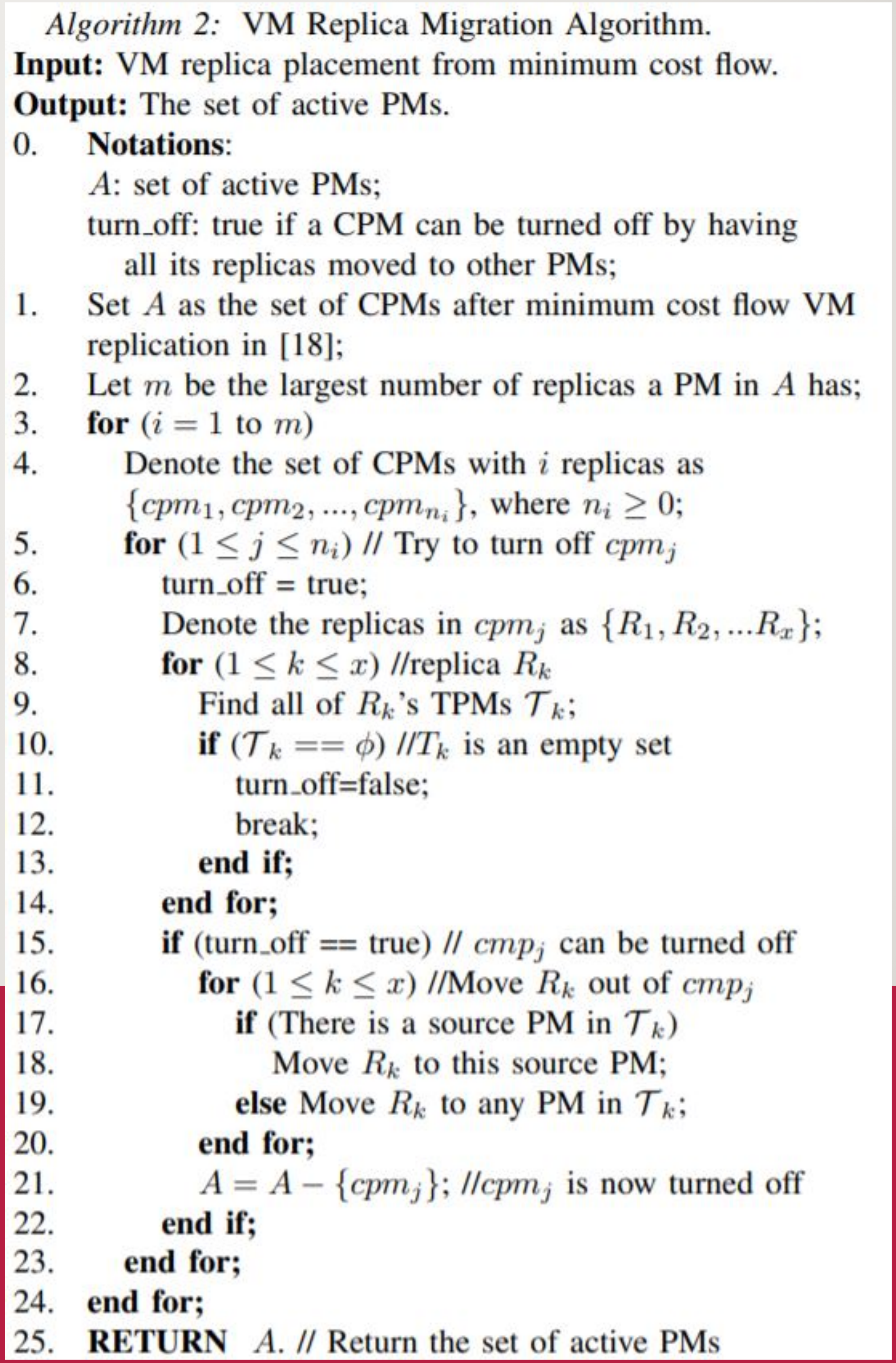
- Works for a minimum cost flow (MCF)-based replica placement.
- Finds TPM for each CPM with one replica. If successful, moves replica to this TPM and turns this CPM off.
- Checks CPMs with two replicas. If both can be moved, turns it off.
- Checks CPMs with more replicas until all the CPMs are checked.

ALGORITHM 2: VM REPLICA MIGRATION ALGORITHM

- Based on two observations of Khani et al.:
  - Replicas on a CPM are moved regardless of if rest of the replicas can be moved or not.
  - When there are multiple TPMs that a replica can be moved to, it randomly chooses one.

ALGORITHM 2: VM REPLICA MIGRATION ALGORITHM

TIME COMPLEXITY: $O(L^2 |V|^3)$



Algorithm 2: VM Replica Migration Algorithm.

Input: VM replica placement from minimum cost flow.

Output: The set of active PMs.

```
0.  **Notations**:
    $A$: set of active PMs;
    turn_off: true if a CPM can be turned off by having
      all its replicas moved to other PMs;
1.  Set $A$ as the set of CPMs after minimum cost flow VM
    replication in [18];
2.  Let $m$ be the largest number of replicas a PM in $A$ has;
3.  **for** $(i = 1$ to $m)$
4.      Denote the set of CPMs with $i$ replicas as
        $\{cpm_1, cpm_2, ..., cpm_{n_i}\}$, where $n_i \geq 0$;
5.      **for** $(1 \leq j \leq n_i)$ // Try to turn off $cpm_j$
6.          turn_off = true;
7.          Denote the replicas in $cpm_j$ as $\{R_1, R_2, ...R_x\}$;
8.          **for** $(1 \leq k \leq x)$ //replica $R_k$
9.              Find all of $R_k$'s TPMs $\mathcal{T}_k$;
10.             **if** $(\mathcal{T}_k == \phi)$ //$T_k$ is an empty set
11.                 turn_off=false;
12.                 break;
13.             **end if;**
14.         **end for;**
15.         **if** (turn_off == true) // $cmp_j$ can be turned off
16.             **for** $(1 \leq k \leq x)$ //Move $R_k$ out of $cmp_j$
17.                 **if** (There is a source PM in $\mathcal{T}_k$)
18.                     Move $R_k$ to this source PM;
19.                 **else** Move $R_k$ to any PM in $\mathcal{T}_k$;
20.             **end for;**
21.             $A = A - \{cpm_j\}$; //$cpm_j$ is now turned off
22.         **end if;**
23.     **end for;**
24. **end for;**
25. **RETURN** $A$. // Return the set of active PMs
```

PERFORMANCE EVALUATION

- Compare how many PMs can turn off by each algorithm.
- Compare inactive PMs (IPMs) resulted from each algorithm.
- A small k = 8 fat-tree data center with 128 PMs and a large k = 16 data center with 1024 PMs.
- VM replica migration are based on two replica placement scenarios:
  - MCF-based VM replica placement (Khani et al., ICC 2016).
  - Random.

- As l increases, the number of IPMs decreases, as more VM replicas are placed inside the cloud data center thus less number of PMs can be turned off.
- When l is small (20 and 40), the performance of all the four algorithms are similar, and each of them can further turn off 15- 25 PMs.
- However, when increasing l, ILP always outperforms the other three by turning off at least one more PM, as it is an optimal solution

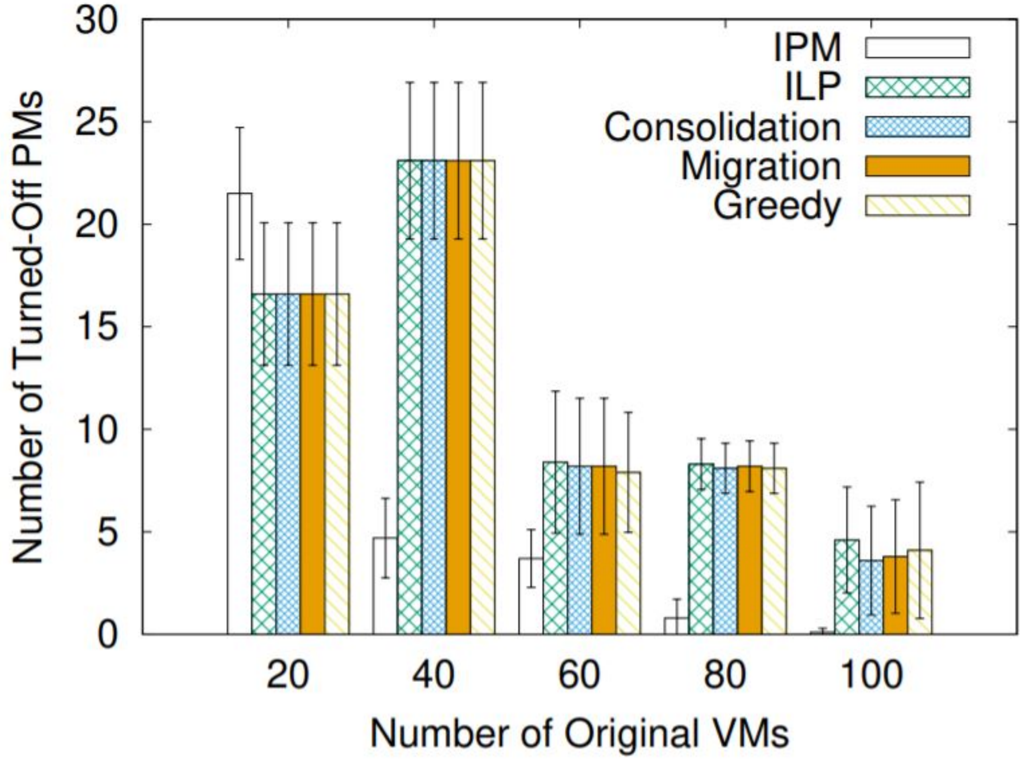


Fig. 4. Performance comparison by varying l, number of original VMs. Here, $k = 8$, $r_j = 10$, and $rc_i = 10$.

RANDOM VM REPLICA PLACEMENT

- Random initial VM replica placement that satisfies the fault-tolerance, compatibility, and resource capacity constraints, and execute these four algorithms on this placement.
- For compatibility constraint, we assume that each original VM and its replica copies cannot be placed to a randomly chosen specific PM due to software or platform incompatibility.

- while increasing number of original VMs l the number of turned off PMs increases.
- As l increases from 20 to 60 and to 100, the number of IPMs of the random placement decreases from 60 to 10 to 2, thus there are more PMs that are initially on to be turned off by the algorithms.
- Compared to placement from MCF-based VM replication (Fig. 4), which can turn off less than 25 PMs, in random VM replica placement, our algorithms are able to turn off around 100 PMs.

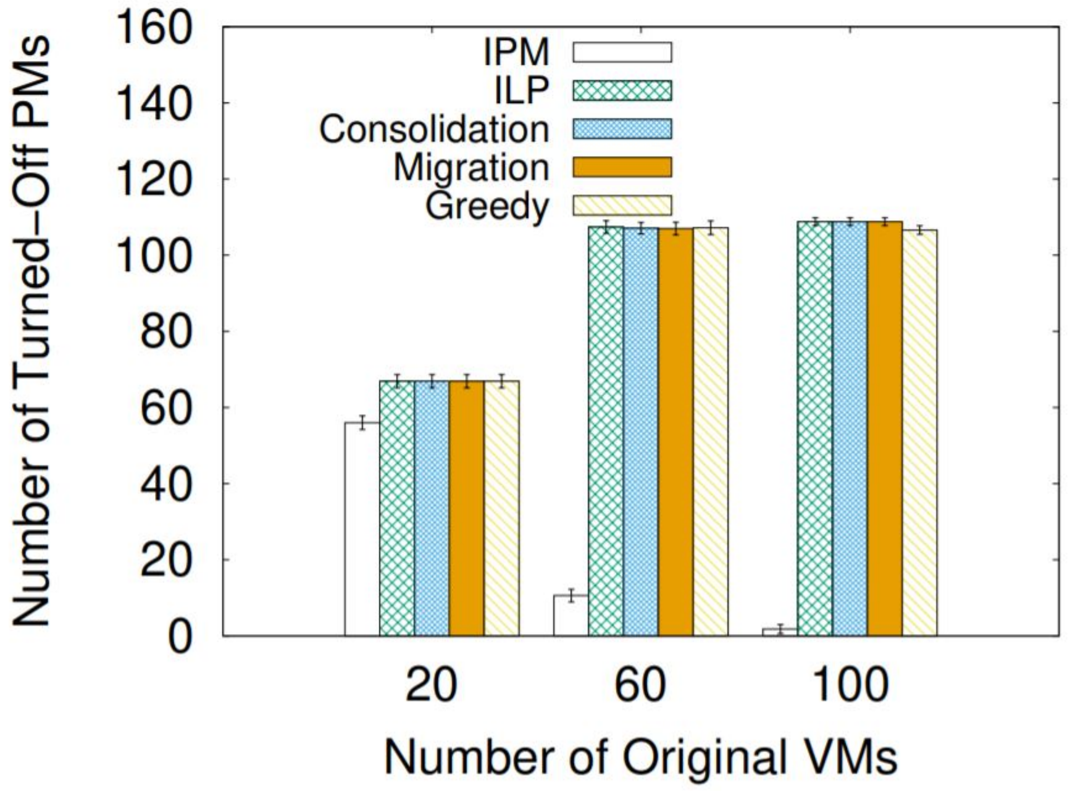


Fig. 7. Performance comparison in random placement by varying l, number of original VMs. Here, $k = 8$, $r_j = 10$, and $rc_i = 30$.

SCALABILITY STUDIES

- A large scale k = 16 data center of 1024 PMs. As ILP takes long time to compute, we only compare Greedy, Consolidation, and Migration.

- Number of IPMs and number of turned-off PMs decrease with the increase of r_j .
- As more copies of VM replicas are placed, more difficult to find an IPM or turn off any existing PMs.
- Turn off close to 700 PMs

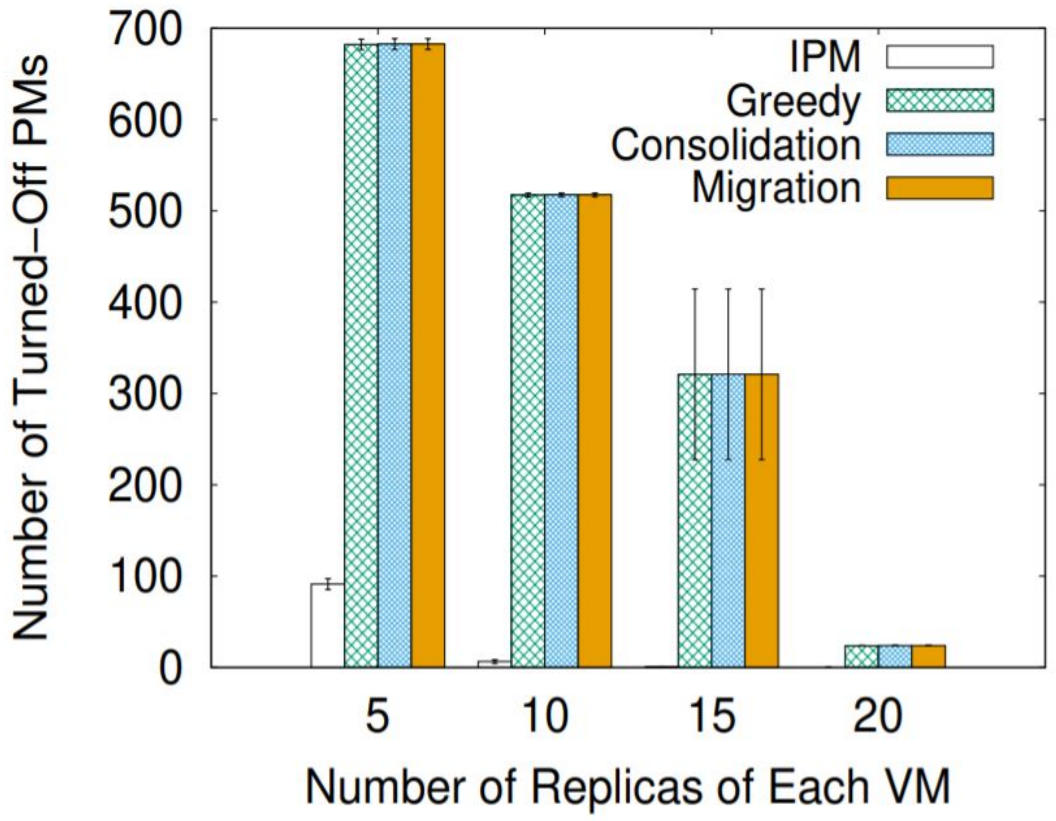


Fig. 8. Scalability study by varying r_j, number of replica copies of each VM. Here, $k = 16$, $l = 500$, and $rc_i = 10$.

- When increasing the resource capacity of each PM, all three algorithms are able to turn off around 700 PMs.
- Effectiveness of our algorithms in turning off PMs.

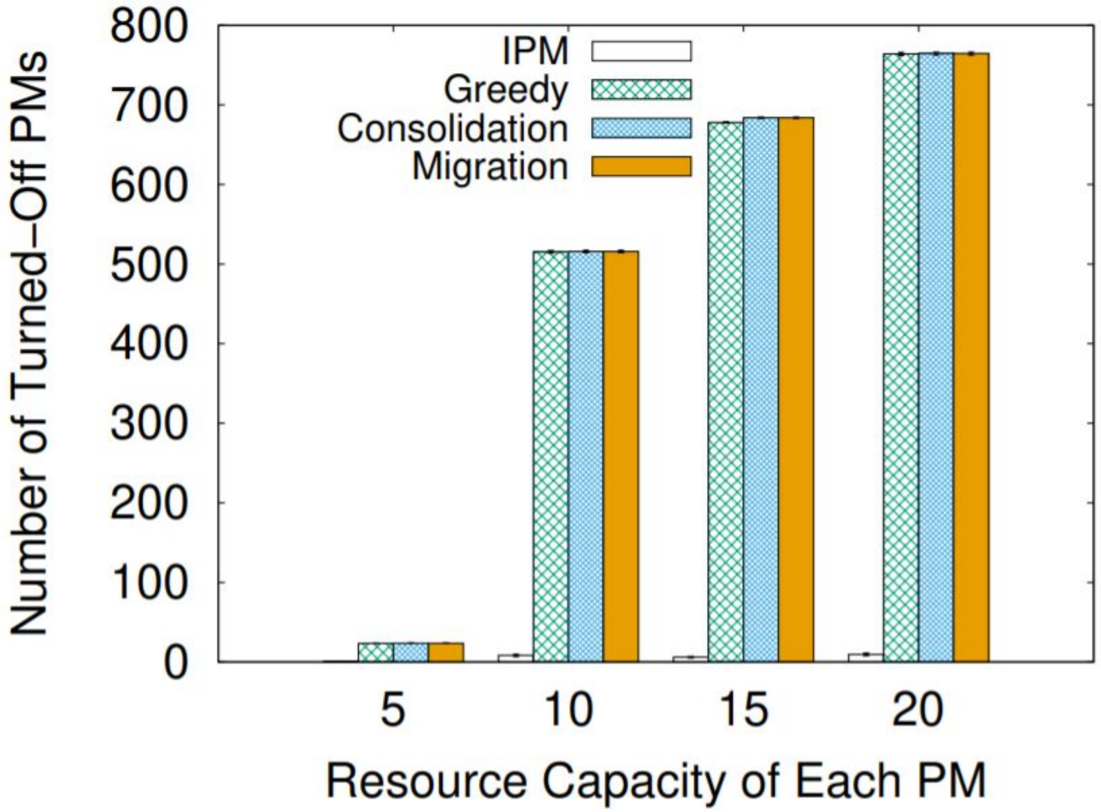


Fig. 9. Scalability study by varying rc_i, storage capacity of each PM. Here, $k = 16$, $l = 500$, and $r_j = 10$.

CONCLUSION AND FUTURE WORK

- Proposed FT-VMP: a new fault-tolerant VM placement problem.
  - Prove NP-hardness, and design an optimal ILP algorithm.
  - Feasibility of FT-VMP using maximum flow.
  - Time-efficient VM replica placement and migration algorithms.
  - Our algorithms are fault-tolerant and energy-efficient.
- Future work: consider energy consumption in both network and PMs and use multi-objective optimization techniques.

ACKNOWLEDGEMENT

- This work was supported in part by NSF Grant CNS1911191.