How to save more energy?

Group 7

LIU, Wei	Anthony Lo
WANG, Duo	WANG, Runqiu
ZHANG, Qian	ZHOU,Lin

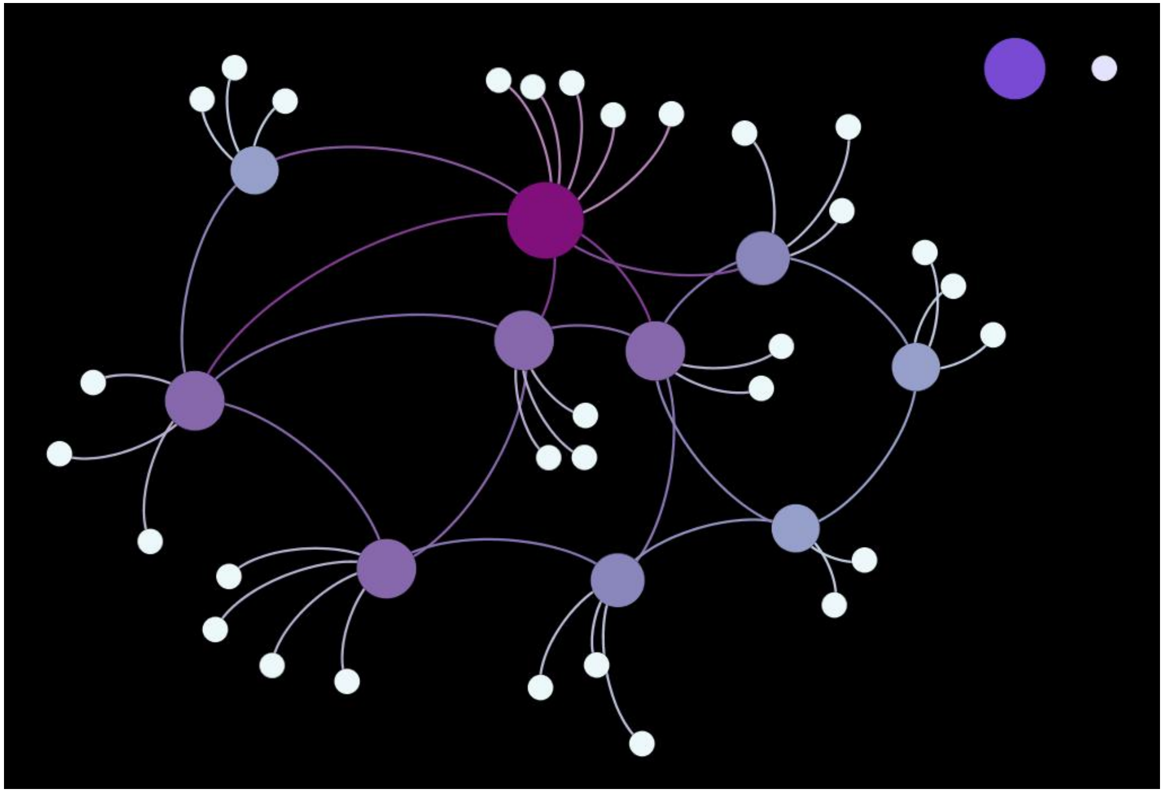
Background and Problem

- With the emerge of cloud computing paradigm, a large number of future-generation data center will use virtualization and cloud computing technology, high efficiency of energy management is a great challenge.

- What we want to solve is to find a way to the save energy of physical machine in the model of Virtualized Cloud Computing.

Our idea

Data Center Network Model



■Formally, we consider a data center network with dependency graph G(V,E);

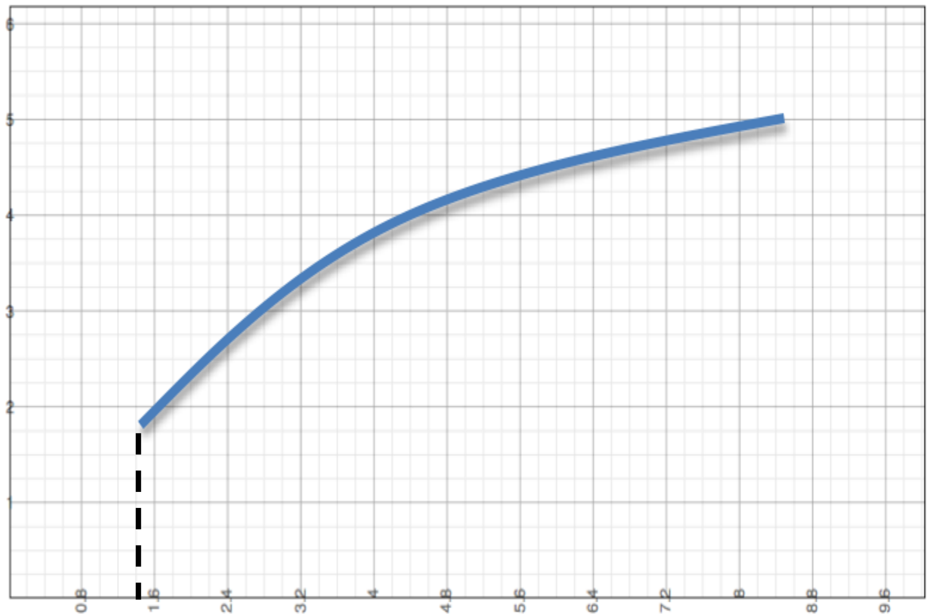
■where V = {v1, v2, ..., vn} is the set of VMs;

■E is the set of edges defined as E = (vi, vj) : vi, vj ∈ V, two VMs vi and vj are dependent with each other if any communication takes places between them;

■Let P = {p1, p2, ..., pm} denote set of physical machines;

■Then, the set of VMs hosted by physical machine p ∈ P can be denoted by V (p).

Assume we have a figure:

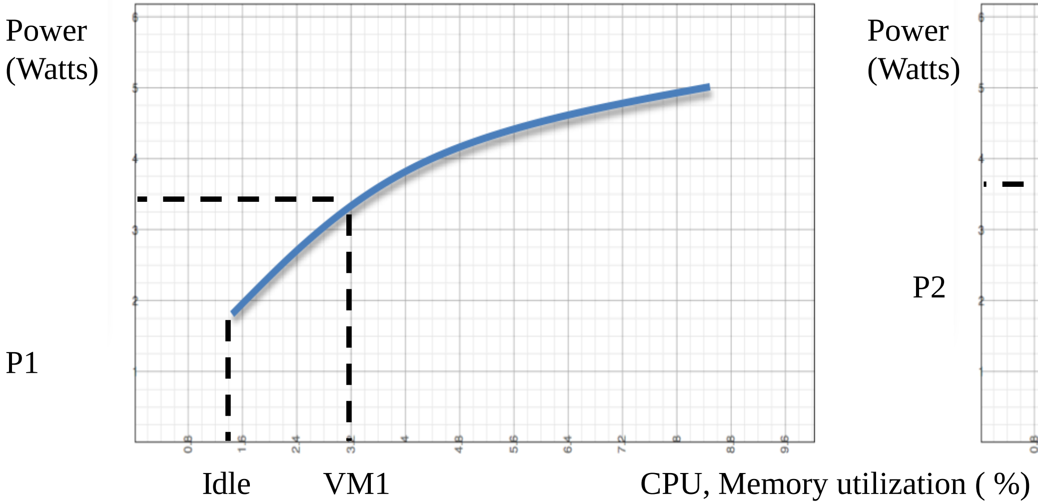


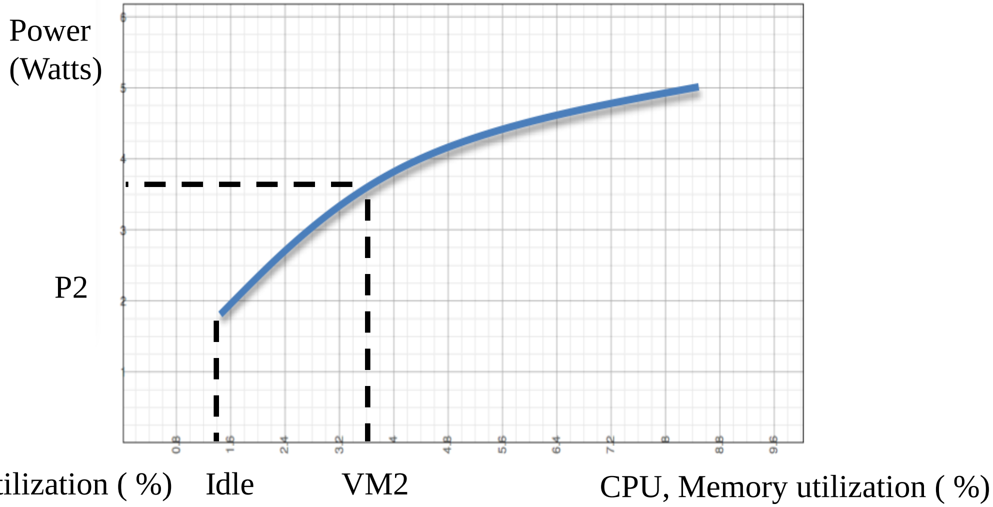
Power=f (CPU, Memory utilization)

- Can be modeled by experiment
- f (CPU, Memory utilization) can be different for different physical machine

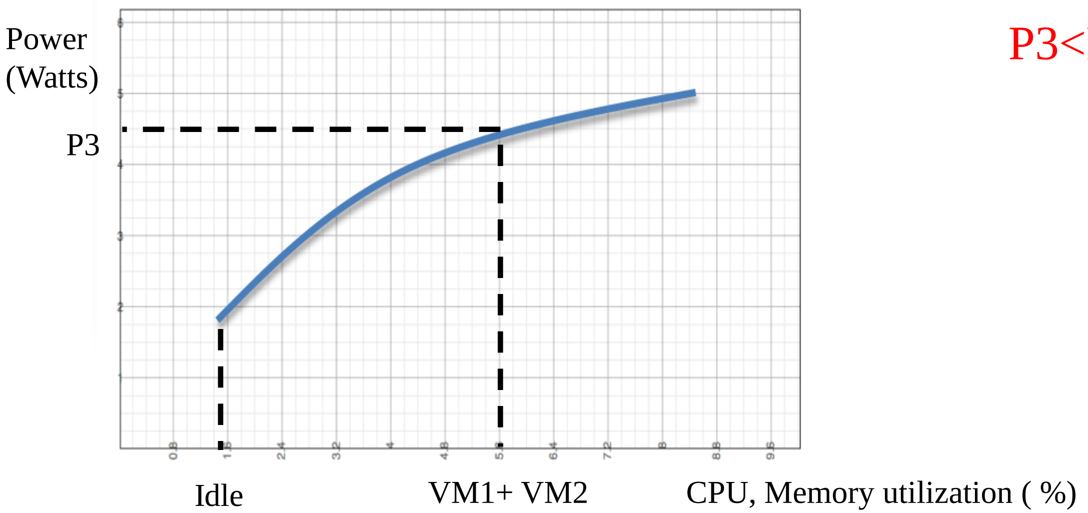
Why need migration?

Before





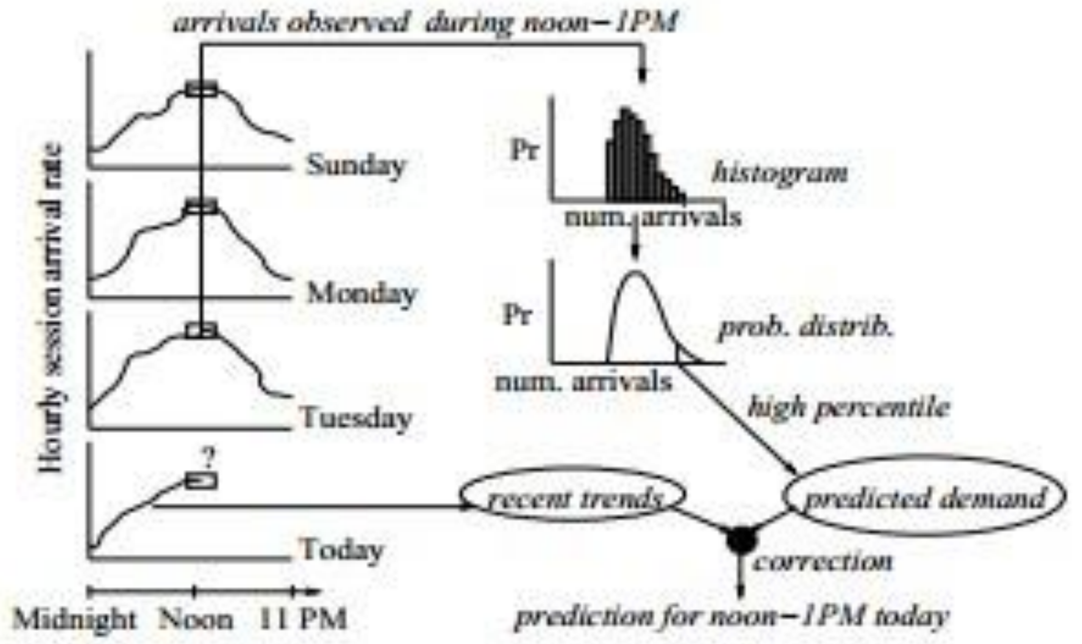
After



P3<P1+P2

Methodology - Predictive

Predict the usage based on historical data



Methodology - Reactive

```
main（）
while（(changeflag,i) = Monitor（））
    if(changeflag == Reduce)
      VM-Migration（i）
    if(changeflag == Increase)
       Join-Multicastdomain（i）
    MormalRun()
```

```
Monitor（）
in every period,monitor all PM
        if(there is a PMi that U(i)<ε)
            return Reduce&i
        else
            if(there is a PMi that U(i)>η)
               return Increase&i
        return Normal&0
```

```
Join-Multicastdomain（PMnumber i）
use the existing algorithms
```

```
VM-Migration（PMnumber i）
for k=1,2,3......n
   if(U(PMk + PMi) < η)
     compute Sk=f(U(PMi + PMk)) - f(U(PMk))
   else
       Sk= -MAX
find k that Sk is the smallest in {S1,S2,......Sn}
migrate all the VM on PMi to PMk
```

we set function U(PM) to compute the utilization of PM given.
we assume that ε is the lowest critical limit of hardware use and η is the highest critical limit of hardware use which can be set heuristically.
The CPU utilization is changing every second, and assume it is a long time average.

Preliminary result

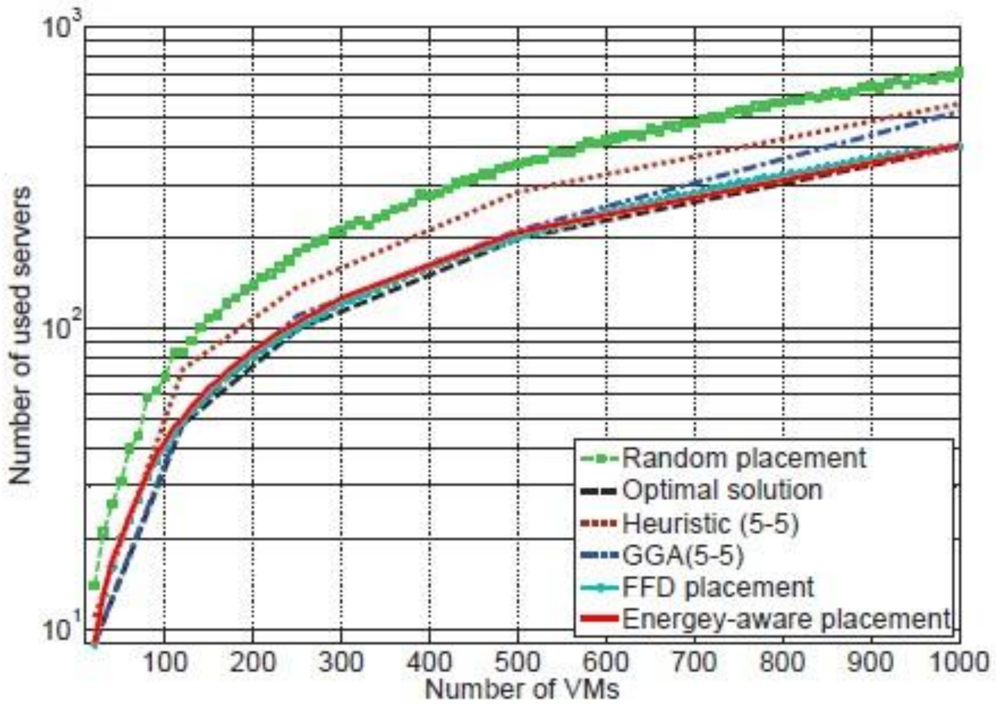
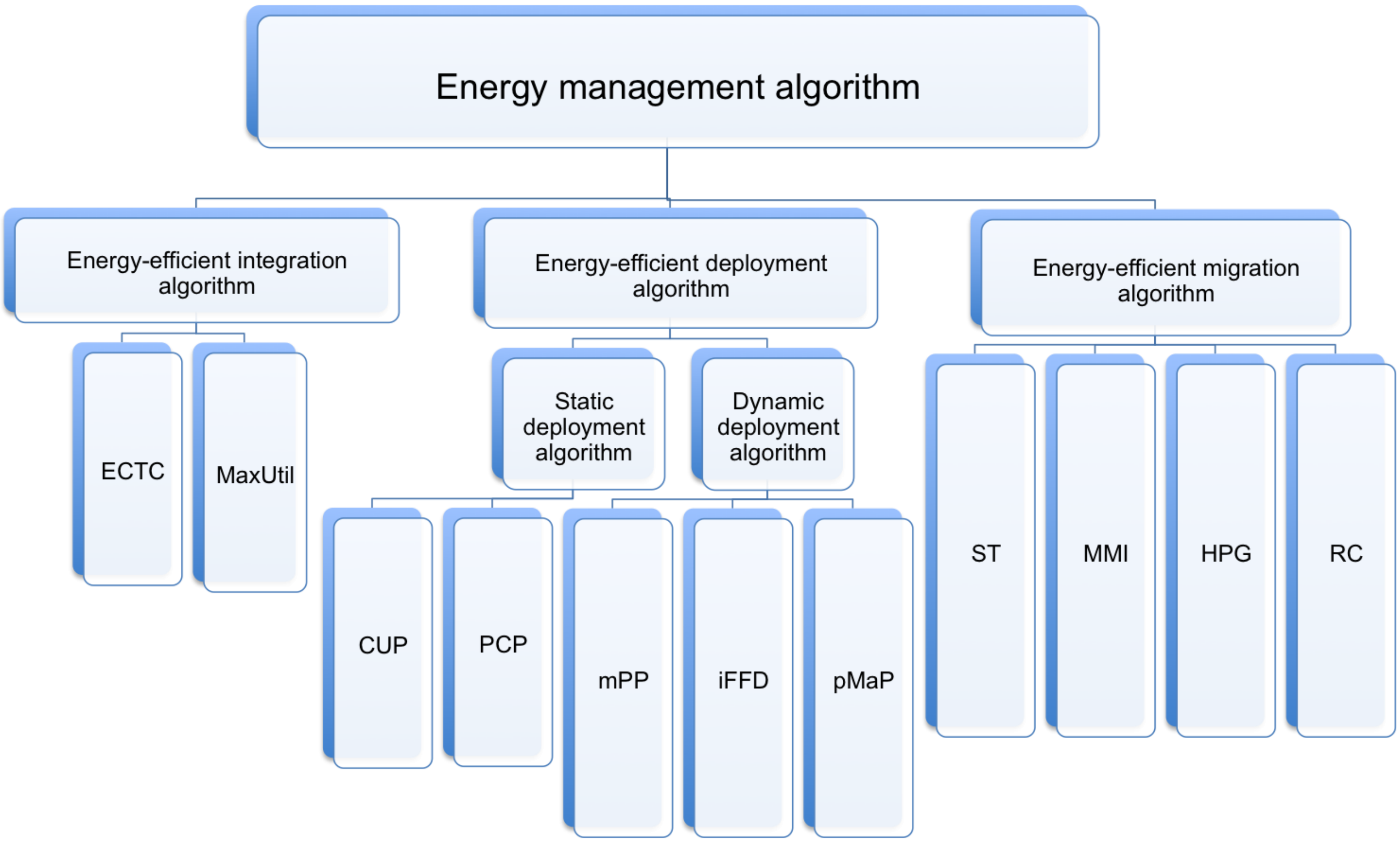


Fig. 4. Number of used physical machines for five placement algorithms

Now the best energy-aware VM placement algorithm can maximize the utilization of the physical machines while decreasing data center traffic by up to 50% ～81%, given the conditions of the server-side constraints and different data center network architectures, significantly reduce server energy consumption and network energy consumption in data centers.

Preliminary result

This graph shows useful power management of virtualized cloud computing platform algorithms we have now.



Limitations of this work

- We did not consider the migration lost. (The lost of shutting down a physical machine and re-opening a VM on another PM.)

- We assume we need to remove all the VMs from a PM and shut down a PM completely. Did not consider the case that we only migrate part of the VMs.

Q & A

References

[1] Huang D, Yang D, Zhang H, et al. Energy-aware virtual machine placement in data centers]//Global Communications Conference (GLOBECOM), 2012 IEEE. IEEE, 2012: 3243-3249.

[2]GE Jun-wei, ZHANG Bo, FANG Yi-qiu. Study on Resource Monitoring Model in Cloud Computing Environment [J]. Computer Engineering, 2011, 37(11): 31-33

[3]LIU jin-jun, ZHAO Sheng-hui. Design of multiple virtual machines management model for cloud computing [J]. Journal of Computer Applications, 2011, 31(5): 1417-1422

[4]YE Ke Jiang, WU Zhao Hui, JIANG Xiao Hong, HE Qin Ming. Power Management of Virtualized Cloud Computing Platform [J]. CHINESE JOURNAL OF COMPUTERS, 2012, 35(6): 1262-1285

[5] Urgaonkar B, Shenoy P, Chandra A, et al. Dynamic provisioning of multi-tier internet applications [C]//Autonomic Computing, 2005. ICAC 2005. Proceedings. Second International Conference on. IEEE, 2005: 217-228.