Efficient Pre-Copy Live Migration of Virtual Machines *for* High Performance Computing in Cloud Computing Environments

Kasidit Chanchio and Jumpol Yaothanee

vasabilab

Department of Computer Science,

Faculty of Science and Technology

Thammasat University, THAILAND

Introduction

- Many VMs in modern Clouds are used for running HPC applications
- HPC applications are long-running applications
- They are usually computation-intensive and memory-intensive
  - Instances with over 8 vcpus and 64 GB Ram are offered by AWS and Google compute
  - They are used for scientific computation, big data analysis, enterprise applications, etc.

VM Live Migration

- VM live migration is a mechanism to move a VM from a source to destination host
- It is highly transparent to applications because the implementation is in the hypervisor
- Advantages:
  - Provide resiliency in case of partial failures
  - Load balancing
  - Move computation to data

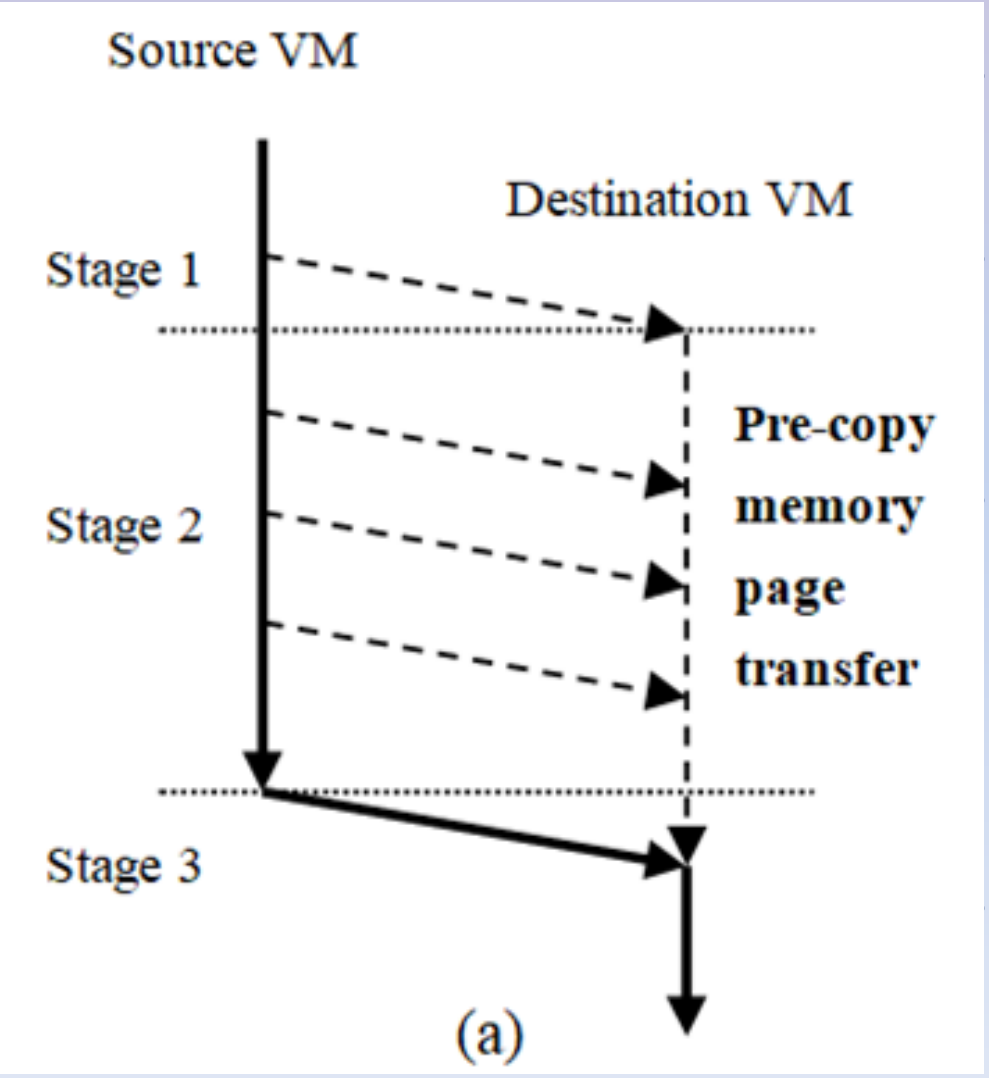
Problem Statements

- The most-popular VM live migration mechanism, namely the **pre-copy**, is NOT effective and NOT efficient
- It requires manual configurations in order to work properly
- Prior to launching a migration, users must define the maximum tolerable downtime that is suitable to VM execution
  - This parameter is hard to define

Memory-Bound Pre-copy Live Migration of VMs

- This paper presents the **Memory-bound Pre-copy Live Migration** (**MPLM**) mechanism
- It does NOT require the maximum tolerable downtime parameter
- It always complete within a Memory-Bound period of time
- It is implemented on top of the pre-copy implementation of QEMU-KVM-2.9.0

Pre-copy Mechanism



Stage 1: Setup stuffs

Stege 2: Transfer VM's memory while the VM is running

Repeat until the remaining data in memory are low enough (to send within the max downtime)

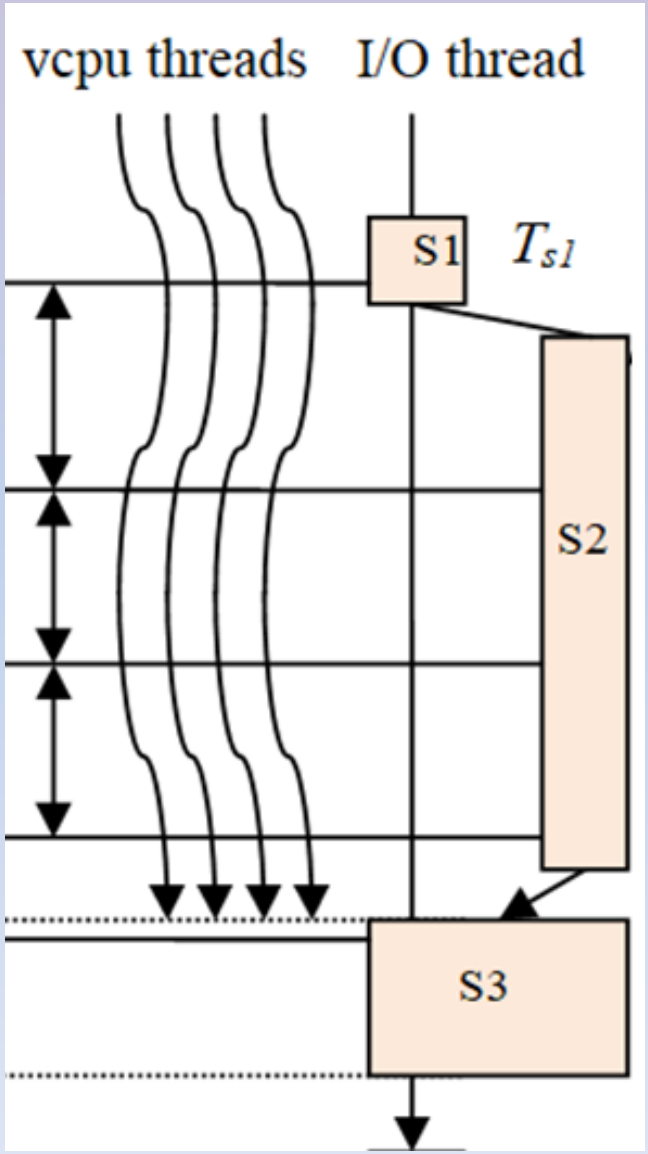
Stage 3: Stop &

transfer the rests

Problems of the Pre-copy Mechanism

- The default maximum tolerable downtime is 300 milliseconds, OK for light workloads
- The maximum tolerable downtime parameter is hard to define for HPC applications
- Set maximum tolerable downtime too Low
  - Exceedingly long live migration time
- Set maximum tolerable downtime too High
  - High downtime

MPLM

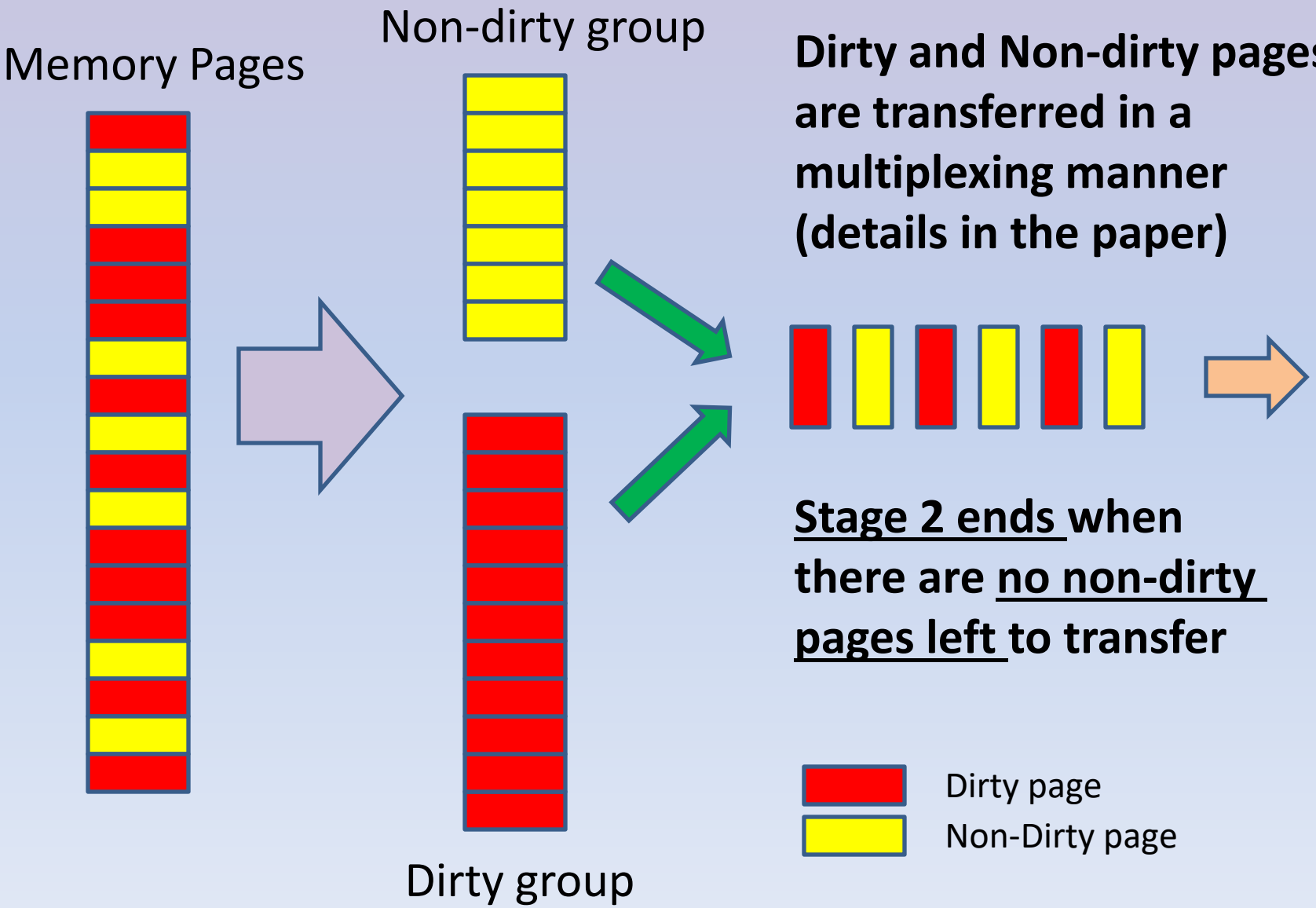


Stage 1. Setup:
Create migration thread, track memory updates

Stage 2. Transfer memory while VM is running:
2.1. Divide Memory Pages into two groups: the non-dirty-page and dirty-page groups
2.2. Transfer the non-dirty pages and dirty pages in a multiplexing manner
2.3. Stop live migration when all the non-dirty pages are transferred

Stage 3. Stop VM and Transfer remaining dirty pages

Stage 2's Live Data Transfer



MPLM Performances

- Total Migration time = Live Migration time + Migration Downtime
- Live Migration time = a period of time Stage 2 operates
- Downtime = a period of time the VM stops at Stage 3
- The higher dirty page generation, the longer the migration downtime

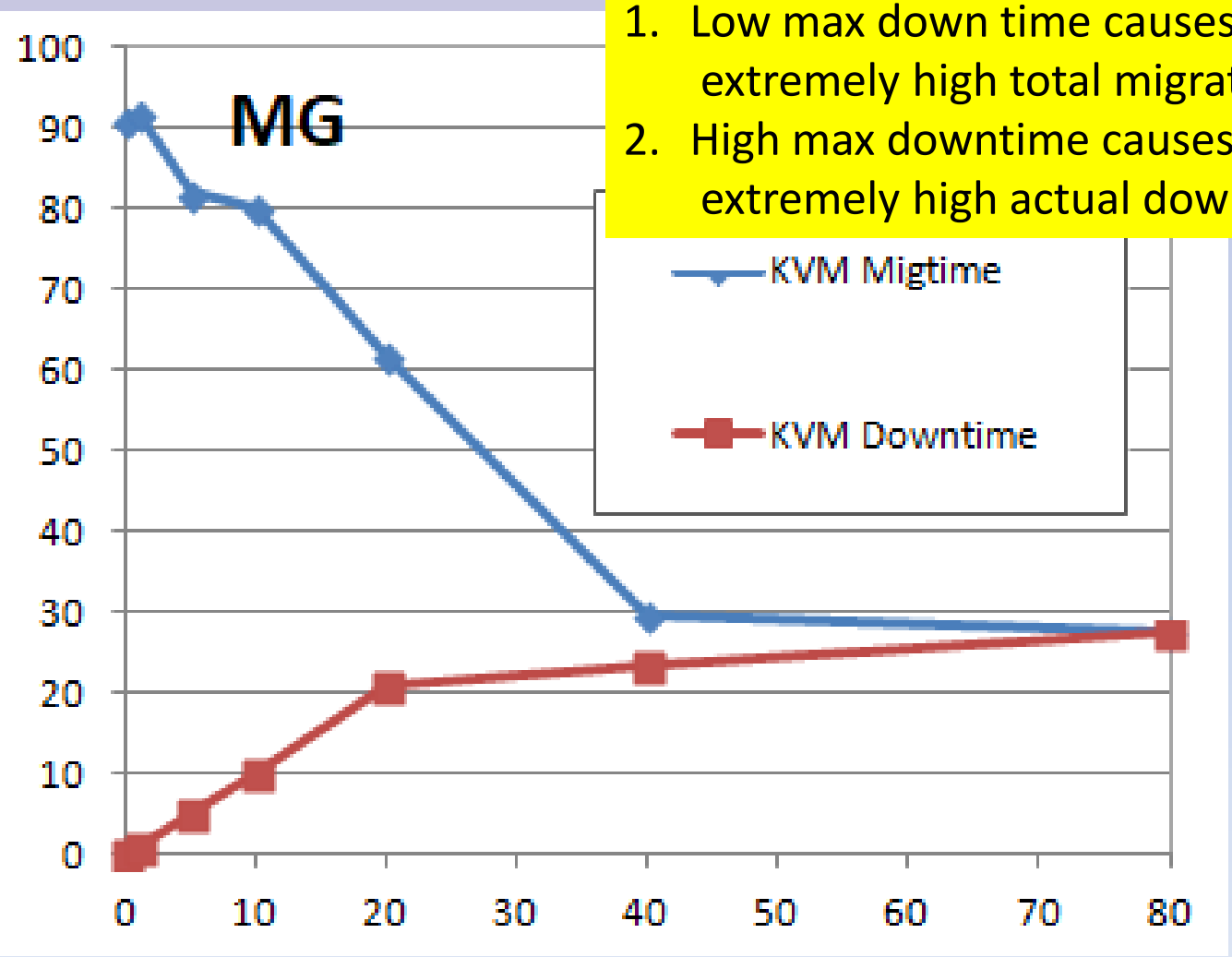
Experiments

- We use two AMD Opteron host servers as the source and destination hosts
- Create a VM with 8 vcpus and 8 GB of Ram
- Run one of the 4 OpenMP Class C NAS Parallel Benchmarks (below) on the VM
  - MG, IS, SP, BT
- Migrate the VM over a 1 Gbps network

Notations

- “KVM” represents KVM’s pre-copy mechanism
- “MPLM” represents MPLM mechanism
- “Migtime” = Total migration time
  - (including downtime)
- “Downtime” = Migration downtime
- For KVM, X-axis represents the maximum tolerable downtime
- Y-axis represents actual time in seconds

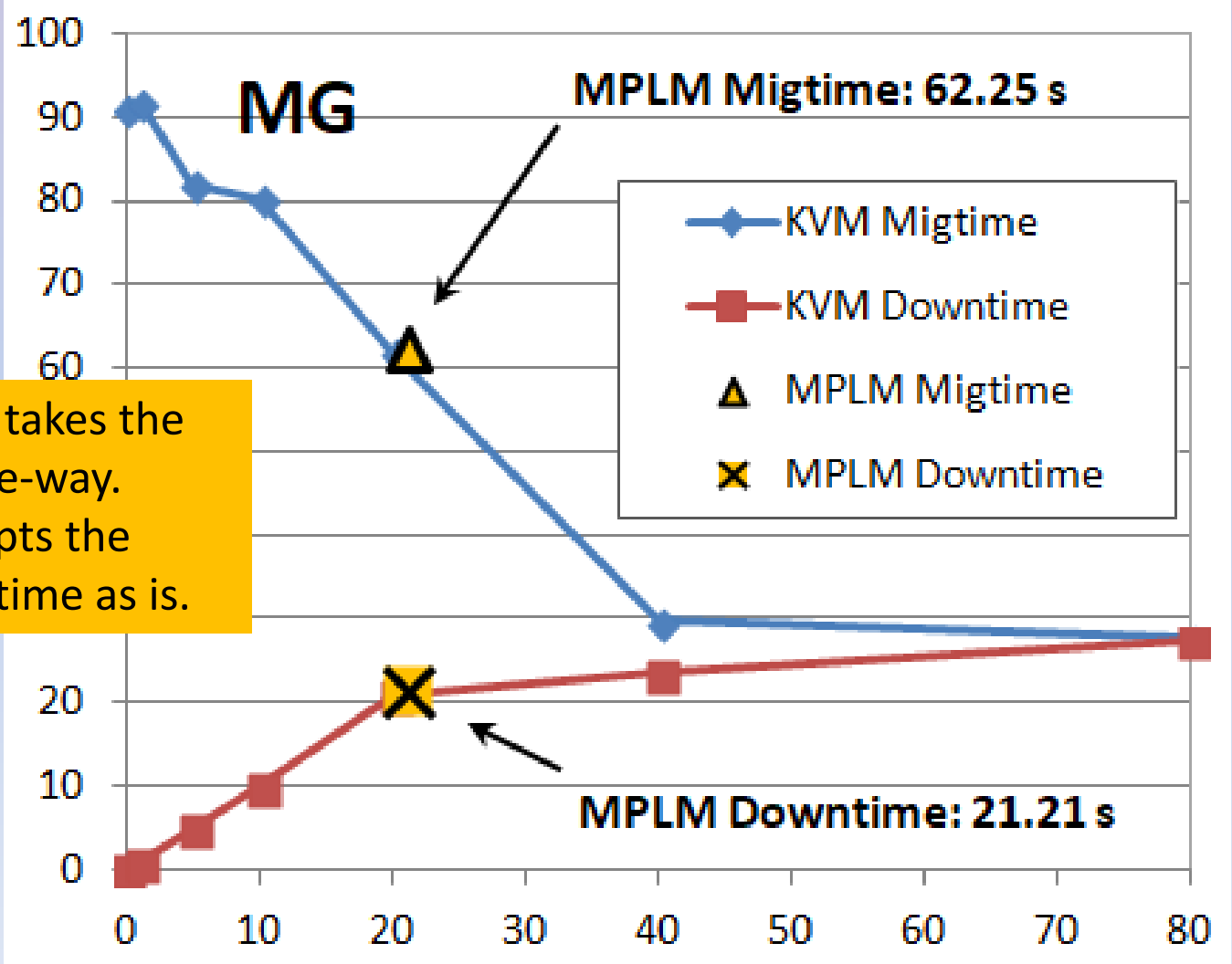
Multi-Grid Solver (MG)



1. Low max down time causes extremely high total migration time
2. High max downtime causes extremely high actual downtime

Maximum Tolerable Downtime

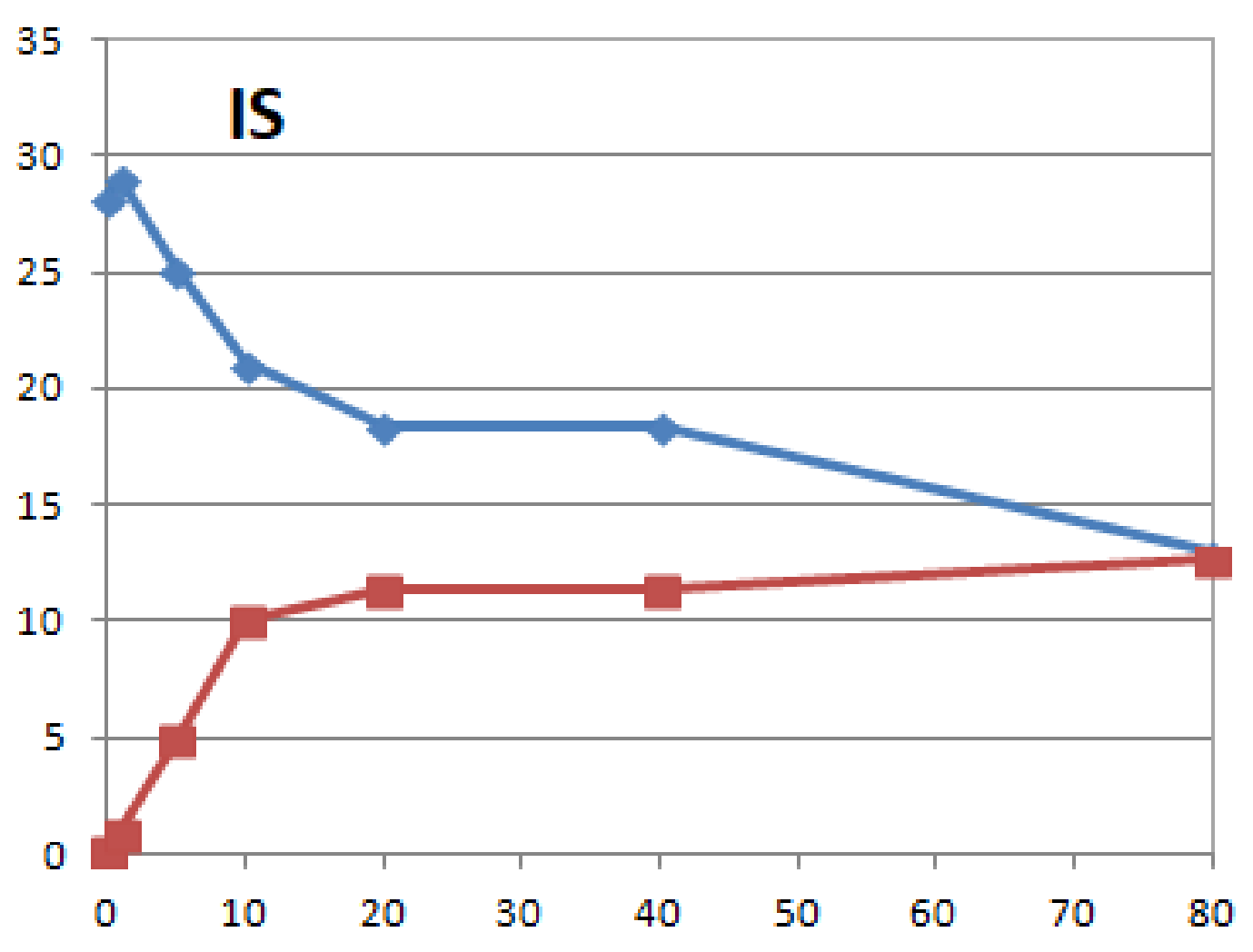
Multi-Grid Solver (MG)



1, MPLM takes the middle-way.
2. It accepts the downtime as is.

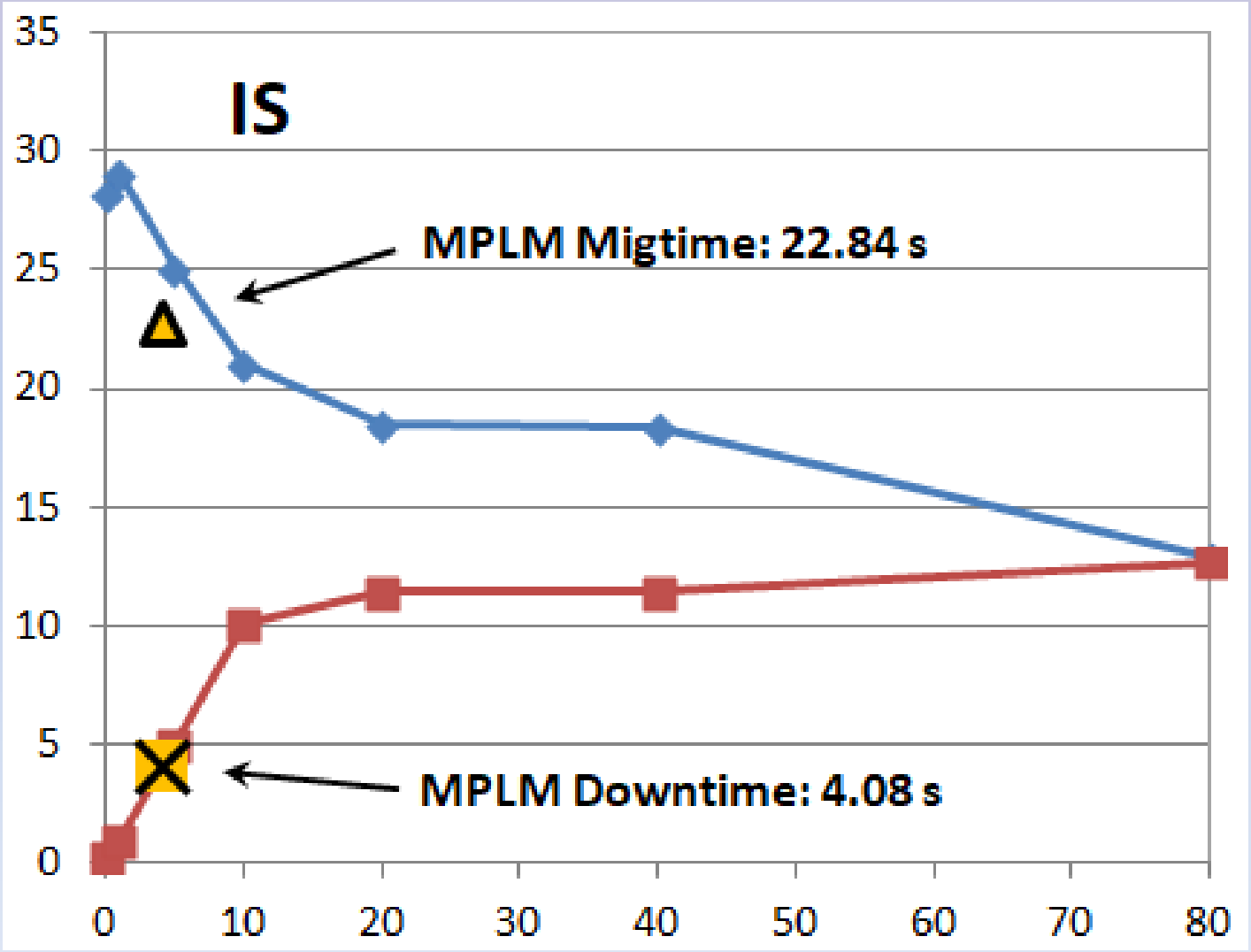
Maximum Tolerable Downtime

Integer Sort (IS)

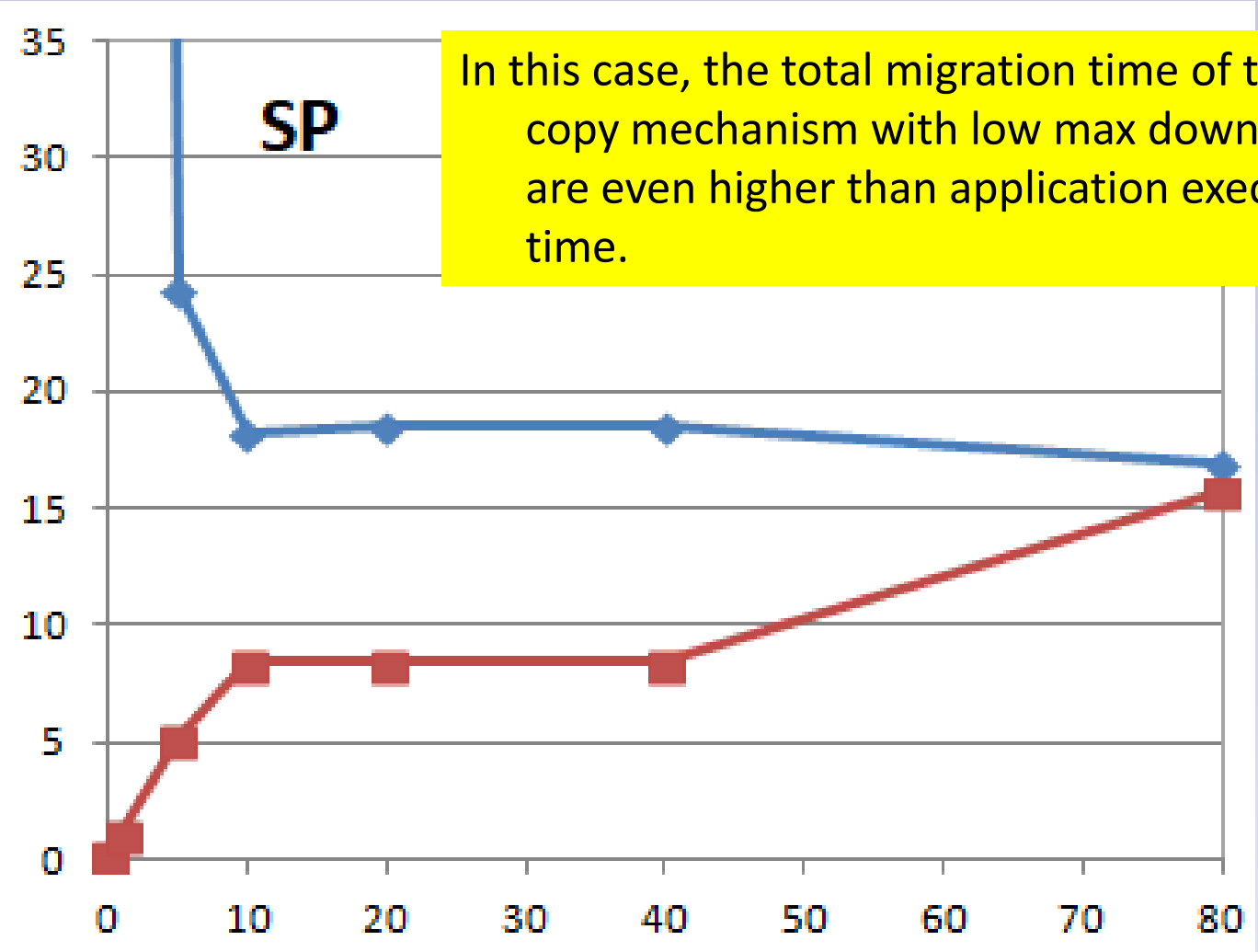


Maximum Tolerable Downtime

Integer Sort (IS)

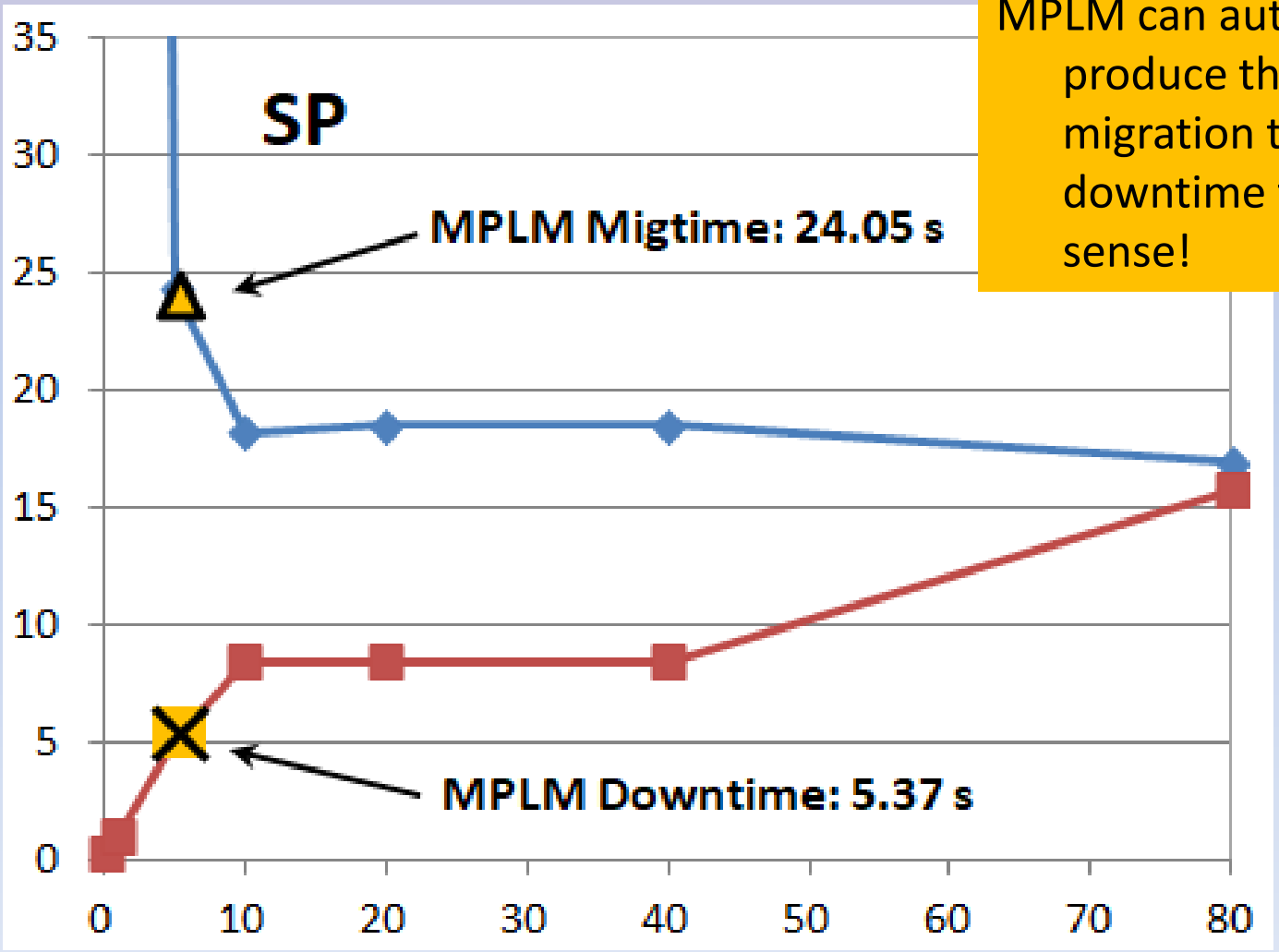


Scalar Penta-diagonal solver (SP)



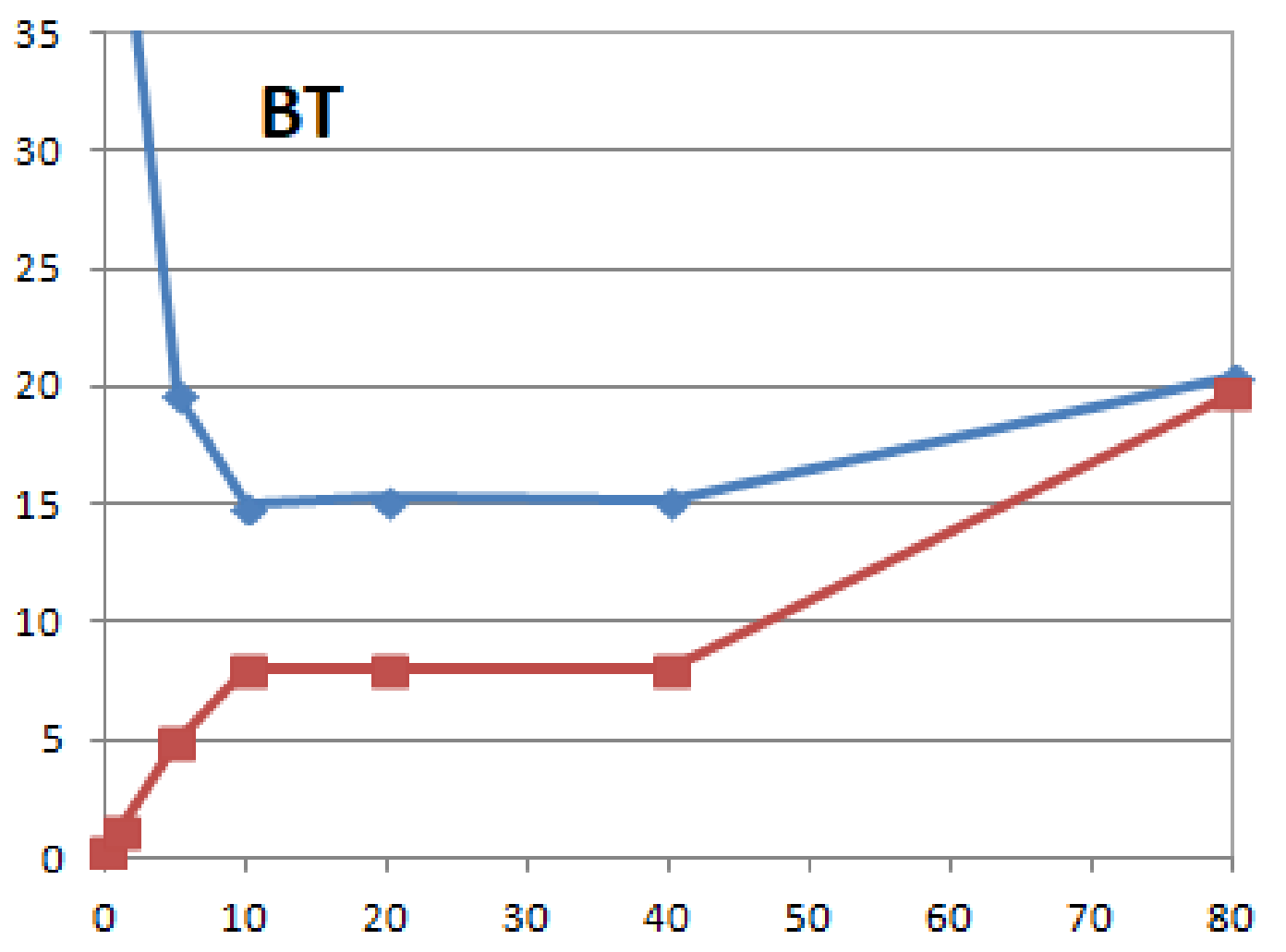
In this case, the total migration time of the pre-copy mechanism with low max downtimes are even higher than application execution time.

Scalar Penta-diagonal solver (SP)

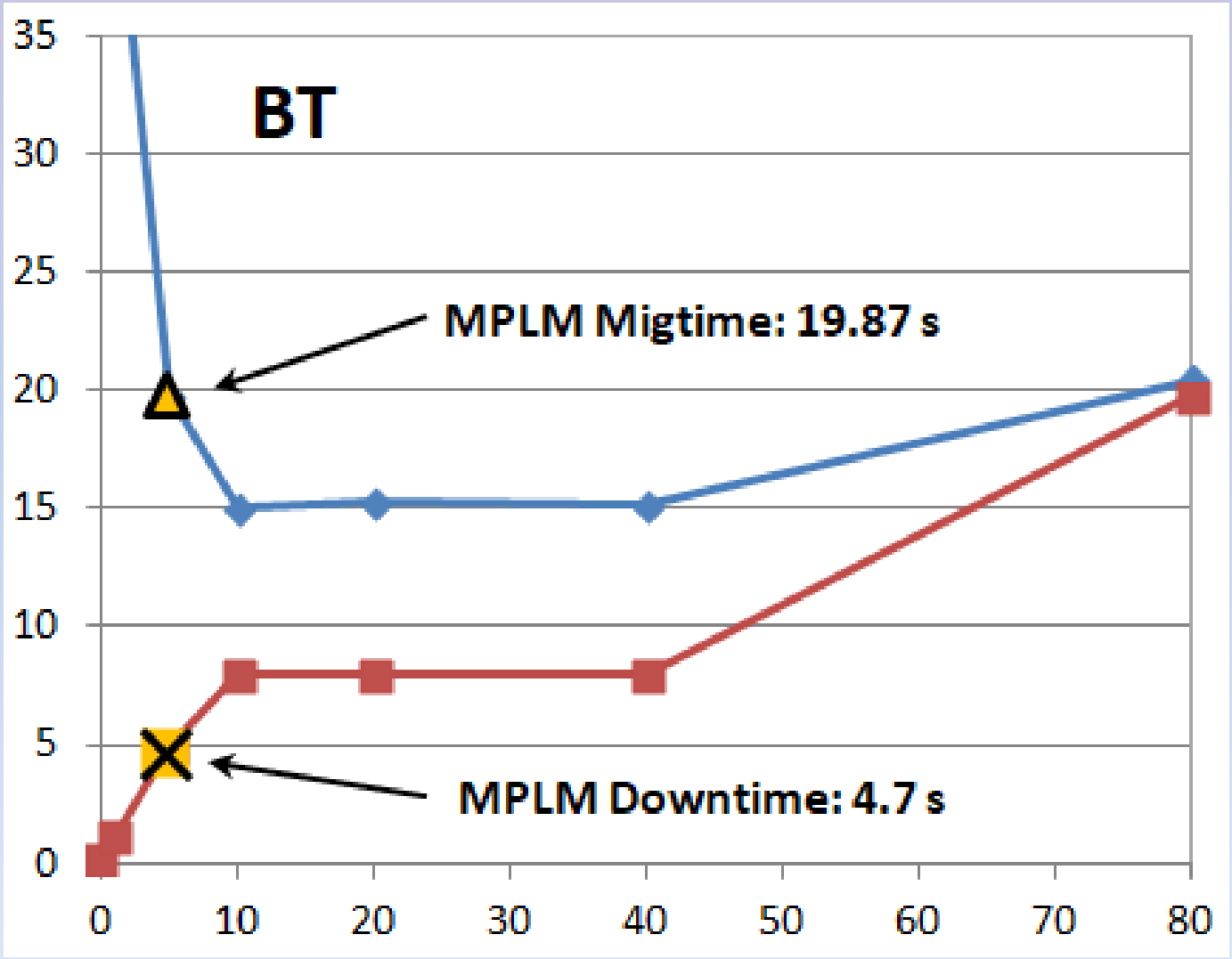


MPLM can automatically produce the total migration time and downtime that make sense!

Block Tri-diagonal solver (BT)



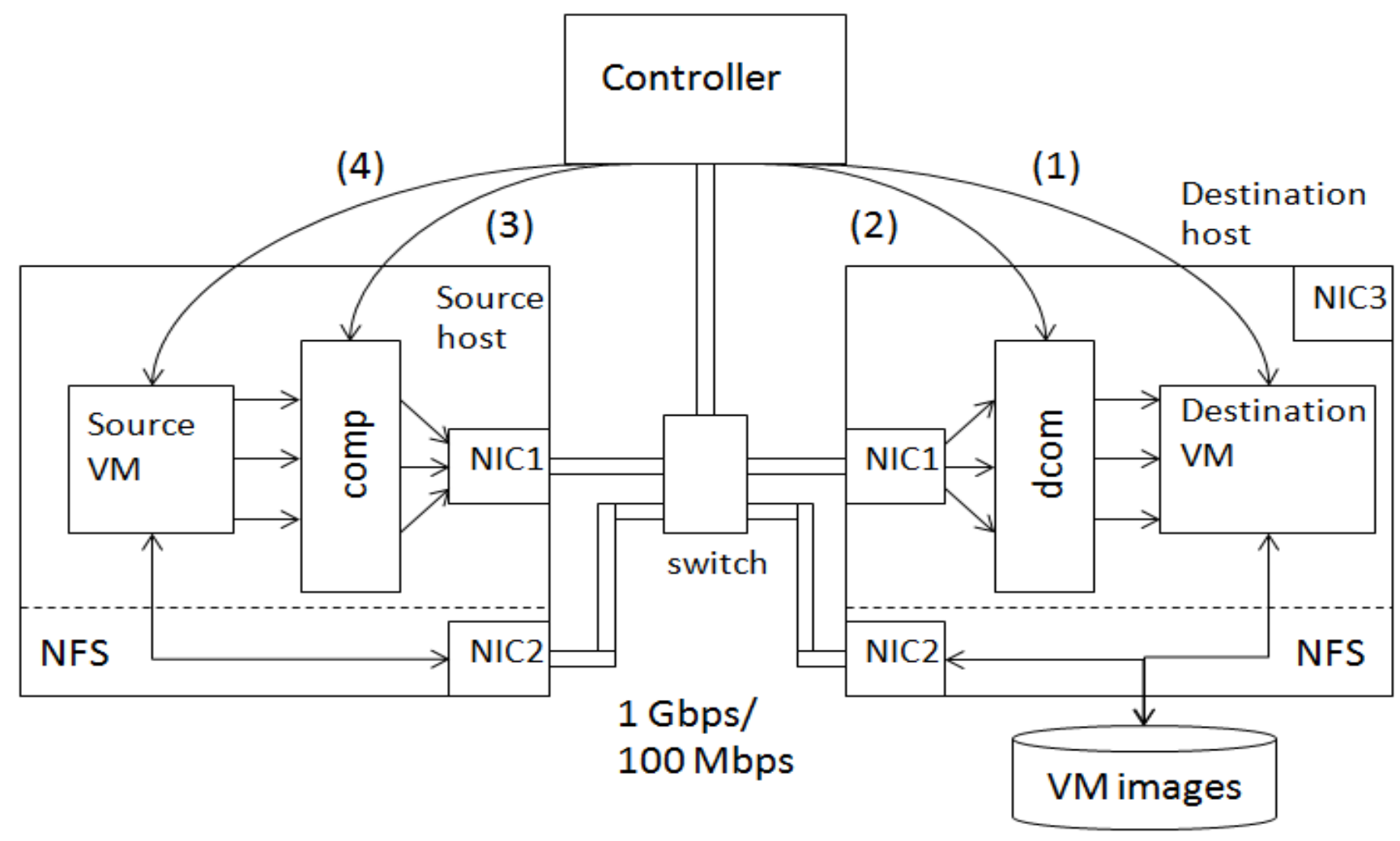
Block Tri-diagonal solver (BT)



Conclusion

- MPLM provides a middle way
- MPLM does not require Manual Configuration
- MPLM is Effective: take a predictable time to complete
- MPLM is Efficient: low migration time& low downtime
- MPLM is Good for large data center where automatic resource management is favorable

Backup


Controller
(4)
(3)
(2)
(1)
Destination host
Source host
NIC3
Source VM
comp
NIC1
switch
NIC1
dcom
Destination VM
NFS
NIC2
NIC2
NFS
1 Gbps/
100 Mbps
VM images