Virtualization Demonstrator

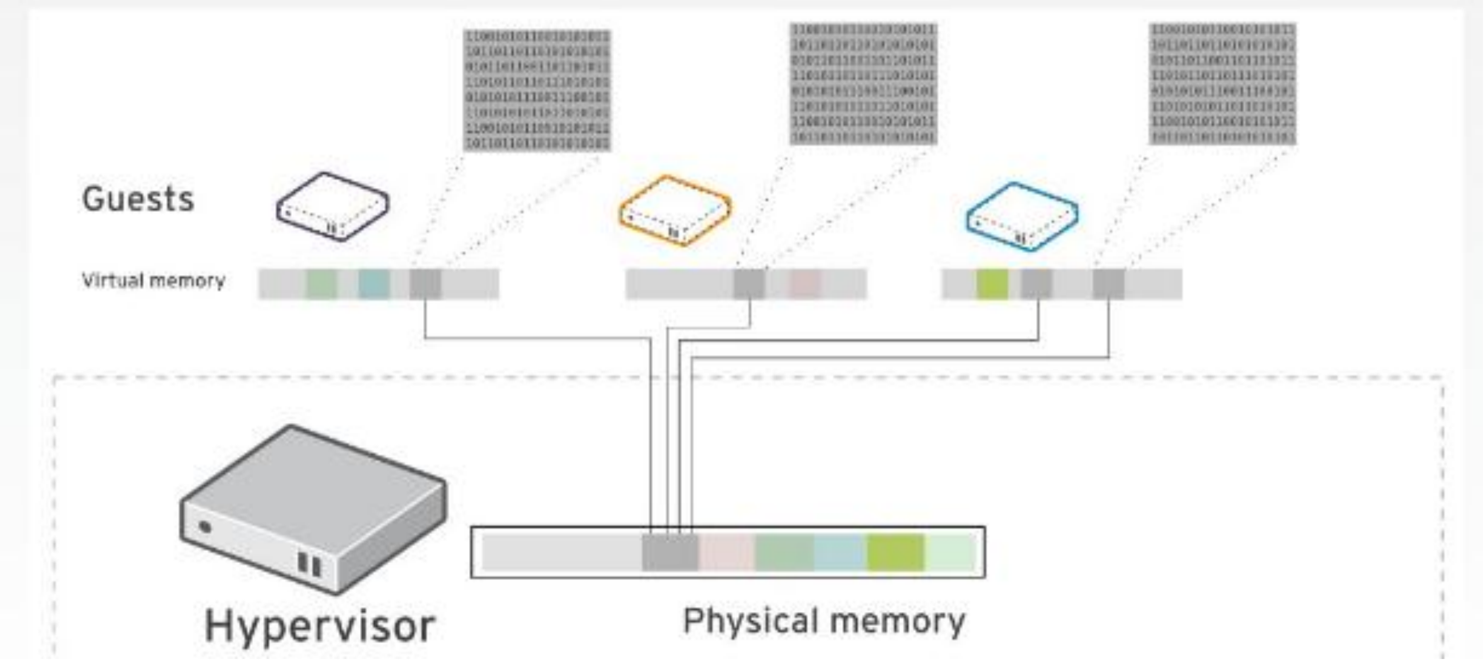
Predrag Buncic/
CERN/PH/SFT

The checklist for improving performance in many core environment

- Work on better usage of single core capabilities
  - Exploit instruction and vector level parallelism
  - Identify and recode the critical parts in the code
  - Use a low level optimized library to vector operations
- Improve I/O and data handling
  - With so many cores, bringing enough data to CPU is going to be the biggest challenge
  - Multi level caching and buffering may be required
- Reduce memory utilization per core
  - Identify the pieces of information that can be shared between processes
  - Rework data structures so that sharing is possible
  - ✓ Share what can be shared (shared memory, COW, KSM)
- Parallelize execution of certain workflows
  - Ideally those that do not require or generate lots of I/O such as simulation

If virtualization cannot help, how can we be sure that it won’t hurt?

Sharing Memory with KVM/KSM



- This is where Virtualization can really help
- KVM/KSM (Kernel Samepage Merging) provides a mechanism to share the same memory pages between guests
  - If there is something to be shared, KSM it will be shared
- Equivalent mechanisms for memory sharing exists for other VMMs (VMware…)

Our objectives

- Near term
  - Develop tools to benchmark application performance and resource consumption while running in virtualized environment
    - Basic tools/scripts exist, need to be properly packaged (2 weeks)
  - Integrate LHC application(s) benchmarks into standard CernVM test suite for regression testing
    - Testing framework developed last year as GSOC project
    - Here we need some input and guidance form the experiments
    - Investigate potential for memory sharing across virtual machines
    - Possible task for a summer student (6 months)
- Mid to long term
  - Depending on technology evolution and availability of the hardware
  - Investigate possible use of GPGPU/MIC from VM environment
  - End of the year