Performance

Cpu caches:

1) L1: 32kb size/1ns speed
2) L2: 256kb-2MB size/4ns speed
3) L3: 3-8 MB size/20ns speed

Metrics for CPU cycle, RAM capacity, Disk capacity, Disk I/O and Network I/O:

1) Utilization: The proportion of a resource that is used or the average time that the resource was busy servicing work.
2) Saturation: The degree to which the resource has extra work that it can't service.
3) Errors: The count of error events for a resource.

More metrics:

1) Latency: the amount of time required to complete a unit of work.
2) Elapsed time: time taken for a batch of operations to complete (ex: commit 1000 records to a database.)
3) Throughput: the amount of work that an application can accomplish per unit of a time.

Note: for microbenchmarking, use the Java Microbenchmarking Harness library

Performance testing tools:

1) JMC (Java Mission Control)
2) JCMD is the command line equivalent of JMC. Cannot connect to remote process.
3) JMeter and JMeter Distributed Testing

Profiler tools:

1) Java Flight Recorder

JIT Tuning:

1) Choose proper compilation mode (tiered, etc…)

GC Tuning:

1) Choose garbage collector (serial, parallel, CMS (concurrent mark sweep), G1GC (garbage first)). Note that Shenandoah collector is experimental at this time, but might be best choice.
2) GC analysis tools are GCEasy.io and GCViewer.
3) Pick initial and max size for heap with xms and xmx flags. Max heap size (sum of all if multiple jvms) should not exceed physical memory of the machine. Heap should be about 30 percent occupied after a full GC.

Algorithm Tuning:

1) Set initial capacity on ArrayLists if you can.
2) Implement Comparable and good hashcodes for your key in a hash map.