RedLine Performance Methodology (RPM)

Getting Maximum Performance Out of HPC Systems

RedLine
Performance Solutions

RedLine Performance Solutions is a world-class provider of high-performance computing (HPC) solutions. Our promise: to ensure objectively engineered top quality solutions at every phase of the HPC life cycle, minimizing labor, time, and costs. Our proprietary RedLine Performance Methodology (RPM) – developed over two decades of working with HPC systems and applications and updated regularly with lessons from each new engagement – delivers unique benefits that consistently maximize customer success.

Written By:

Chris Young

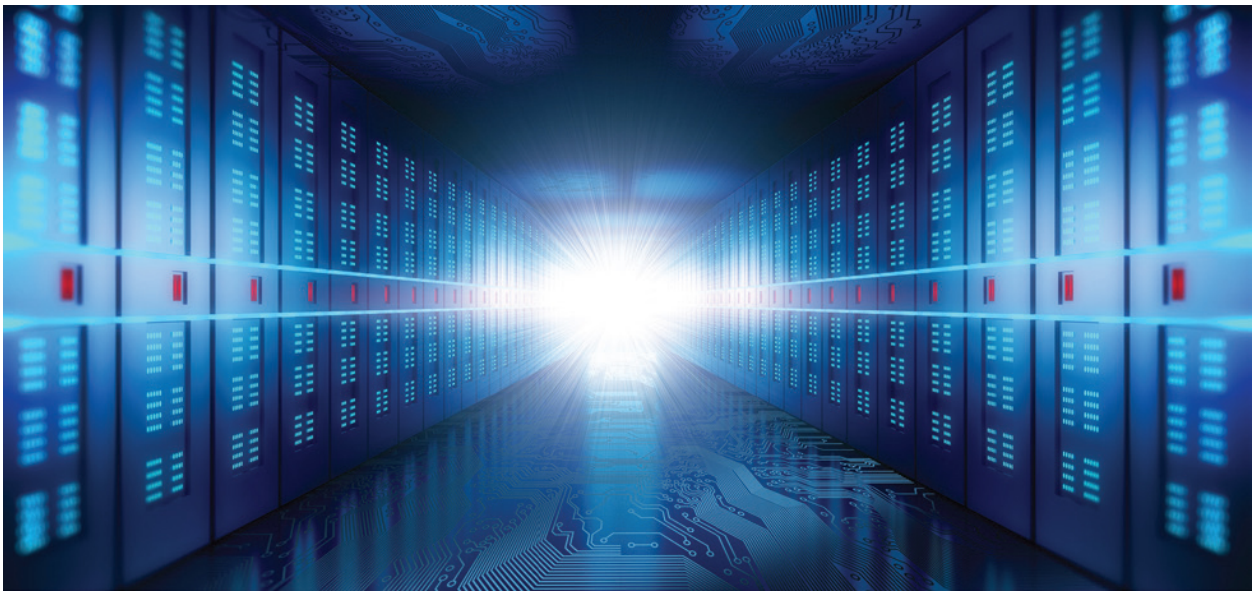
Keith Ball

Andrew Qualkenbush

Mark Potts

Carolyn Pasti

Don Avart



HPC system performance matters – a lot.

That's why we at RedLine Performance Solutions (RedLine) have dedicated our business to ensuring our customers' HPC systems perform at their very highest level.

HPC workloads are vital to citizens, governments, and organizations throughout the world. Many, if not most, of the items we take for granted today were discovered, designed, or in some way made possible by HPC systems. It's not a stretch to say that lives depend on the performance of HPC systems.

One example of this is biological research to improve medicines, cure diseases, and grow better crops. Research into new materials and new energy sources is also highly important to the world as a whole.

At RedLine, HPC performance is our only business. We have developed our RedLine Performance Methodology (RPM) to ensure consistently optimal HPC performance for our customers. Perhaps the best way to explain the RPM and how we apply it is to discuss how we work with the US National Weather Service.

Delivering RedLine Performance

For the past 19 years, RedLine has supported the National Weather Service at the National Centers for Environmental Prediction (NCEP). Our team manages the operational supercomputers that provide forecast guidance products used daily by meteorologists to predict the weather.

NCEP also issues severe weather warnings, along with flight guidance to the FAA and the military. The models used by the National Hurricane Center are also run on NCEP's supercomputers.

The performance of NCEP's HPC systems is absolutely critical to their mission and to the country as well for three primary reasons:

- **Criticality:** Lives literally depend on accurate and timely weather forecasts. Giving people advanced notice of impending weather emergencies is only possible if the systems have enough data and compute power to run the models as quickly and accurately as possible.
- **Cost:** the US government spends large amounts of money to procure and run their supercomputers. It's important they get the maximum value for their purchase. The same principle applies to non-government users. For example, if a network promises 100Gb/sec performance, then it should provide that performance in practice – and it's our mission to

help our customers get the advertised performance out of their HPC systems.

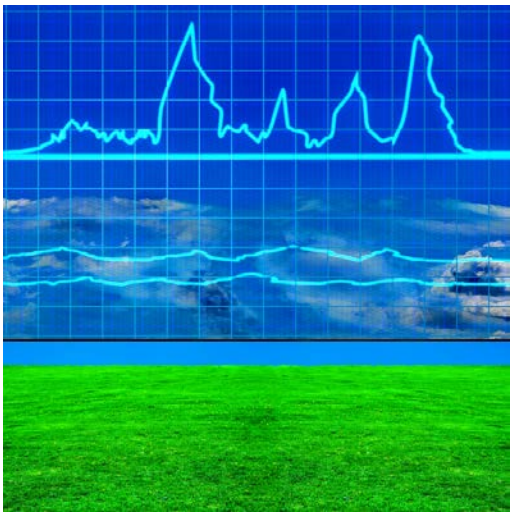
- **Consistency:** NCEP runs multiple weather models 24 hours a day, seven days a week. Production is essentially one long workflow, with data ingested, large models run, and forecast products delivered. The output of one model becomes the input of another in order to arrive at short-term, mid-term, and long-range forecasts. Each of these models uses a massive amount of data ingested from the network, which is staged on the storage network, and funneled through the servers for processing. Even a slight performance problem can have a significant ripple effect causing forecasts to be delayed.

Because our customers' workloads are of utmost importance, RedLine closely monitors system and application performance every step of the way. Potential issues are detected early and corrective action is taken before the issues become problems. This is the only way to ensure mission-critical systems consistently deliver the required performance.

The keys to sustaining consistent high performance are to have extensive, detailed knowledge of system performance, knowing how to monitor performance, and having business rules in place to deal with performance variations.

Not every organization has the same stringent performance requirements as NCEP, however all organizations need to have some level of understanding of their performance requirements and characteristics. Understanding performance enables IT managers to more accurately predict workload completion, recognize the need for more resources, and respond to growth in demand. This also allows them to eliminate guesswork when slowdowns occur or when they are perceived by end users.

Performance, The RedLine Way



We find that organizations often have performance expectations that are not quantifiable. Note that here we state 'expectations' rather than 'requirements', because requirements are quantifiable.

Expectations are usually based on end user experience. When end users complain an application is slow, IT staff are expected to react and solve the perceived problem. However, in a well-tuned and instrumented environment, IT professionals often already know when a problem exists and are working to isolate it.

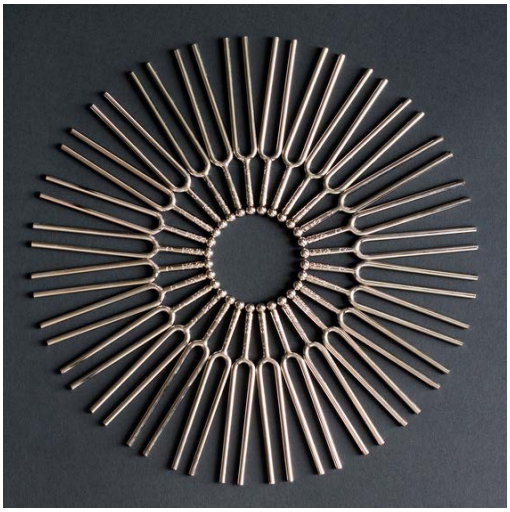
If the environment is not well tuned and instrumented, service professionals have a more difficult time determining whether the performance issue is truly valid, and if it is, determining the root cause of the problem.

Our approach to systems management is based on the premise that organizations need well-defined performance metrics. Understanding system performance requires an understanding of the system from beginning to end.

In general terms, data comes into a system, is stored, manipulated, and output is produced. The system components (network, CPU, storage) work together and have numerous interdependencies. We believe that having a thorough understanding of how each system component performs individually, as well as how they perform together, is fundamental to proactive systems management.

Keys to Systems Management & Maximizing Performance

Whether managing a small cluster of eight systems or a mega-cluster with thousands of nodes, systems management is critical to ensuring maximum performance, predictability, and rapid, effective troubleshooting. Our systems management approach heavily relies on the following five methods:



System Performance Tuning: This is an incredibly broad topic that borders on an art form. It is an iterative process that builds on prior knowledge and constantly evolves.

Great care must be exercised when tuning, as changes in individual systems almost always have downstream and upstream ramifications.

A system's current stage in its lifecycle strongly influences the best approach to adopt for performance tuning. Ideally, performance tuning is initiated in the pre-production phase, but performance tuning can and should be considered throughout all phases of the systems lifecycle. It is imperative, however, that solid systems management practices are in place if a system is farther along in the lifecycle than the pre-production phase.

Baselining: The importance of baseline performance metrics cannot be overstated. Without a measured baseline, performance expectations are based on speculation. Baselining a system includes running simple benchmarks such as network throughput tests, disk IOPs tests, and measuring end-to-end application performance.

After the baseline has been established, baseline tests are re-executed prior to deploying new hardware or software, as well as before and after an upgrade or patch.

Execute the baseline tests to ensure the system is in a healthy state prior to applying the upgrade or patch and again afterward to ensure the change had no negative or unexpected effects.

Baseline tests should be executed whenever there is an opportunity to validate performance in a measured fashion. Baseline tests are also run when performing problem determination. If a bottleneck in the I/O subsystem is suspected, execute the I/O baseline tests to validate the I/O subsystem.

A good baseline test has well-defined performance requirements and a repeatable method to test for the defined performance metric or metrics. This could be an individual component test such as network performance of a WAN link, or a IOPs test to an individual disk drive, RAID array, or filesystem. A good baseline test could also be an end-to-end system test that simulates specific functionality that, in the best case, simulates normal operations. It's vital that detailed performance metrics are collected during baseline testing. It is also important that these performance metrics are captured and stored for future reference.

Systems Monitoring: Effective systems monitoring provides service professionals insight into problems before they are reported by users. It's the difference between relying on an over-temperature light in your car vs. having a temperature gauge showing the temperature as it increases plus the alarm light when the temperature exceeds the healthy threshold.

Many systems monitoring tools are designed to alert system administrators when systems have failed. A dashboard light goes red to inform you a server has crashed. That's definitely valuable information, but having a warning that the server was having a problem before it crashes is much more valuable.

We help our customers find the right tools that will alert them to potential problems before they result in system or application crashes.

Sometimes systems fail suddenly, making it difficult (when there is no solid systems monitoring) to identify the root cause and thus render a quick fix. That's one of the reasons we help customers install extensive instrumentation that helps them quickly and efficiently identify and isolate problems.

Performance Monitoring: Performance monitoring provides insights into system health that basic system monitoring just can't accomplish.

Strangely enough, for an industry dealing with the most complex computing systems in the world, there's very little published and/or discussed around HPC system performance monitoring and performance analysis. HPC performance analysis requires understanding at a deep level where and how systems components are interacting and, most importantly, where bottlenecks exist.

Collecting both high-level and low-level performance data in a time series format is the first step. As stated above, there is no shortage of available tools. Since HPC systems are typically very overhead-conscious, use lightweight low-impact tools for system and application performance data collection.

There is significant value in being able to graphically represent time series performance data. Tools like Grafana enable administrators to rapidly plot time series data collected by various metrics frameworks (e.g., StatsD, collectd, collectl, tcollector) through a web interface. A lesser-known tool, Performance CoPilot, has excellent capabilities in performance data collection, as well as the capability to intuitively display time series performance data. These tools allow for the visual comparison of known good baseline performance data against real-time or near real-time performance data.

The key to performance monitoring is capturing the differences between your known baseline performance and today's performance. If today's performance is subpar, it could be a slow CPU, failing memory, or a drive that wasn't correctly installed. With solid historical performance data, effective data visualization capabilities, and baselines and benchmarks, administrators are able to identify (and fix) problems much more quickly. Making the investment in performance

monitoring will have long lasting benefits that far surpass the cost of implementation.

Change Management: As certain as death and taxes, change is inevitable and it will break your system. Change is an essential part of IT operations and HPC is no exception. Although there is no magic bullet that makes change risk free, good change management will reduce the mean time to repair (MTTR) when change introduces problems into your HPC system.

So what makes a change management process work? The most important aspect is the people. The best change management policies, procedures, and tools are useless unless people adhere to them. System administrators, and particularly HPC system administrators, are typically very smart folks. Unfortunately, some view adherence to a change management process as an unnecessary burden, particularly when it comes to making small and seemingly "obvious" changes to fix a problem.

What is often not fully considered are the upstream or downstream ramifications of change. Even the most experienced system administrator will have a hard time knowing all of the effects changes can have on a complex HPC system.

When failures start, it's usually not the person who made the change who gets that 3:00 AM phone call. If you're that person, the first thing to ask is "What changed?" Knowing what changed, when it changed, why it changed, and how to back out of the change are some of the most valuable benefits system administrators derive from change management.

The value of running baselines and benchmarks before and after a change is critical. Changes will often have an impact that is not obvious. How many times have you received calls from end users saying, "the system seems slower today?"

Running your baselines and benchmarks prior to a change confirms your system is healthy. Running those same baselines and benchmarks after a change allows you to assess the impact of that change. In addition, if your pre-change baselines

and/or benchmarks fail, strongly consider cancelling your change and remediating your system before applying the change.

The bottom line when it comes to change management? There should be zero tolerance for unauthorized change.

Enforcing a policy of zero unauthorized changes is the only way to know change management is being strictly followed. Dealing with change-related problems comes with the territory for an HPC system administrator, but dealing with unauthorized changes is unacceptable. The problem determination process requires system administrators to make logical deductions based on available information. When unauthorized/undocumented changes are made, MTTR suffers.

Detecting unauthorized changes to system images is manageable. Enterprise configuration management tools like Puppet and Chef can be configured to detect and, if appropriate, overwrite unauthorized changes for both stateless and stateful nodes.

These tools have been successfully deployed and utilized in HPC environments. Tools such as Tripwire or OSSEC, known mostly for intrusion detection, provide automated detection and reporting of changes.Even with the best tools, IT staff must be trusted to adhere to change management policies and practices or find a new line of work.

Adhering to the disciplines above will pay off in extraordinary ways. Downtime will be radically decreased, performance will improve, and, perhaps more importantly, performance predictability will increase.

Achieving Maximum Application Performance

So far, we've been discussing how to optimize, monitor, and manage hardware systems. But that's only half of the performance equation. The other half is ensuring your code is optimized for maximum performance and throughput.

Code Optimization: Like system performance tuning, optimizing code is almost an art form. The goal is to achieve a balance in which the system provides the best throughput and overall performance. This means eliminating bottlenecks or mitigating them by having other functions occur simultaneously, thus reducing the impact of the bottleneck.

Having an intimate knowledge of your unique application's execution is critical. Understanding if your application is typically compute, memory, communications, or I/O bound is a good start, but you need to go deeper in order to wring the highest throughput from your system.

Profiling tools are highly useful and can help you see exactly where your functions or subroutines bog down. Profiling refers to the process of observing a program while it's executing, and collecting the time required (and overall time spent) for each function or subroutine in a program. This data may then be sorted or ranked to determine where the most computing cycles are focused. Such frequently used program elements make the best candidates for optimization, because improving their performance will provide the greatest return on the time and effort invested.

Profiling can be conducted in either serial or parallel fashion. Serial profiling produces a set of timings for each function or subroutine in a program. Amdahl's Law limits the degree of acceleration that can be achieved by optimizing any single serial function or subroutine. But Amdahl's Law also extends to parallel programs (or functions and subroutines).

For example, code that is 90 percent parallel and 10 percent serial can only be accelerated 10x through parallelization.

When parallelization is put to work, parallel profiling also results in significantly more data to be analyzed. That's because timing must be measured for each subroutine or function as it runs on each processor used (for MPI-based parallelization) or execution thread employed (for OpenMP-based parallelization).

Furthermore, when a program is parallelized, it becomes necessary to measure and rank communication latency as part of the profiling process. This introduces additional factors to consider when looking for bottlenecks, and will also require time and effort to analyze and address.

Once you have some optimization targets, there are a variety of parallelization and optimization techniques you can attempt to make the code run more efficiently. Here are a few examples:

- If your code is compute bound, you could port some of the code to a GPU or FPGA accelerator. Another option is to try using a different method to solve key parts of your algorithm, like replacing an explicit method with an implicit method, or using a fast multipole method**.**
- If your applications are communications bound, you can work to reduce all-to-all and synchronous collect communications. You can also replace synchronous communications with asynchronous communications to hide communication latency. Or you can rework your algorithm to remove any MPI_Barrier calls.
- Memory bound applications can benefit from redesigning your algorithm to break up global arrays and use MPI-based processes to retrieve data that is stored on other processors.

Optimization is always a trade-off. The more time and effort you expend in adjusting and tuning, the better your processing outcomes should be. If you provide clear guidance to developers on the best options for overcoming various types of constraints – such as memory bound, communications bound, or I/O bound – you can help them boost performance by quite a bit right out of the gate. But if you spend too much time preparing, or too little time processing, you'll start affecting productivity in other ways. Strike the right balance and you'll see productivity improve. Keep up your optimizations, and those improvements should continue.

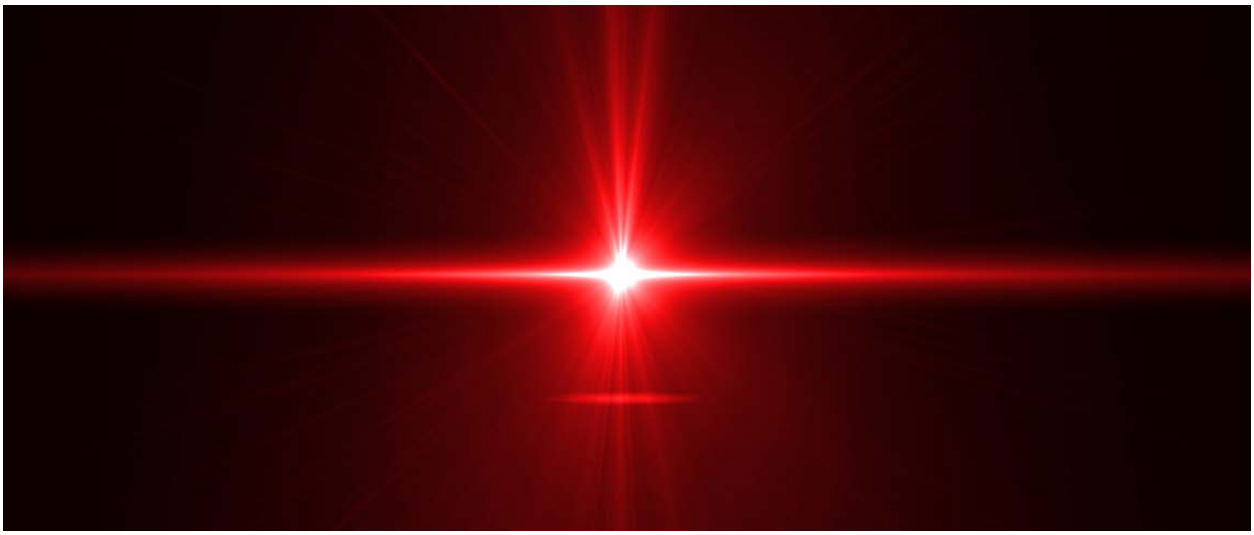
Workflow Management: An often overlooked but critical aspect of overall HPC performance and efficiency is workflow. Workflow management becomes increasingly important as your jobs become more numerous and complex. In most HPC environments, there are many dependencies that must be taken into account. We're talking about situations in which the output of one job is the input for other jobs and the entire application run needs to happen in the right order every time.

Good workflow management will track each phase of operations and ensure the processes happen in the right sequence. It takes into account all workload dependencies and enforces policies. It works with the scheduler to monitor each phase of the job and it won't allow the process to continue if the sub-processes haven't successfully completed.

Your workflow management tool should give you a wide range of options on how it should proceed if a job terminates unsuccessfully or fails to terminate at all. For instance, it could be programmed to attempt to run the preceding job two more times before giving up on it and alerting operators to the failure.

At RedLine, we've worked with a large number of workflow managers, both commercial and open source, and can work with you to find the right tool for your specific situation.

Demand More From Your HPC Systems



Stated plainly, we believe customers should demand more from their HPC systems.

Supercomputers are a scarce and precious resource; moreover, they're very expensive to acquire and operate. This also applies to clusters used in business and manufacturing organizations.

Customers want to get maximum performance, stability, and reliability out of their systems, but they often don't know where to turn for the help they need. Today's HPC systems are typically built using hardware and software from multiple hardware manufacturers utilizing both open-source and commercially licensed software. RedLine's view of systems and system performance can often shortcut the troubleshooting process when dealing with disparate hardware vendors, ISVs and software companies. RedLine's vendor agnostic approach to systems integration coupled with RPM mitigates support concerns and allows customers to select "Best of Breed" solutions.

At RedLine, we take a holistic view of your environment as a tightly coupled architecture. This includes the hardware and software that comprise the cluster, as well as the applications that run on the cluster. We don't specialize in just tuning codes, or network operations, or CPU/memory throughput – we do it all.

As we've discussed, we have developed a wide variety of operationally proven methods, practices, and tools to monitor, manage, and maximize performance for

IT infrastructures ranging from massive supercomputers to a small cluster. We incorporate all of these into our RedLine Performance Methodology (RPM).

Our RPM ensures the customer's environment is fully instrumented and managed, so administrators can identify performance aberrations before they cause problems for end users. We developed and refined this repeatable methodology over the course of more than 19 years of successfully supporting customer HPC systems and applications. RedLine constantly evolves the RPM with the lessons learned from each engagement. The benefits of leveraging our RPM include consistent top performance and minimizing labor, risk, time, and cost for our customers.

Every day we work with our customers, along with their hardware and software suppliers, to tweak and tune every imaginable aspect of the system in order to achieve peak end-to-end performance. It's not called high performance computing for nothing, right?

If you'd like to hear more about what we do and how we might be able to help your organization, just reach out to us for more information.

2275 Research Blvd., Suite 500
Rockville, MD 20850
301-685-5949
info@redlineperf.com
www.redlineperf.com