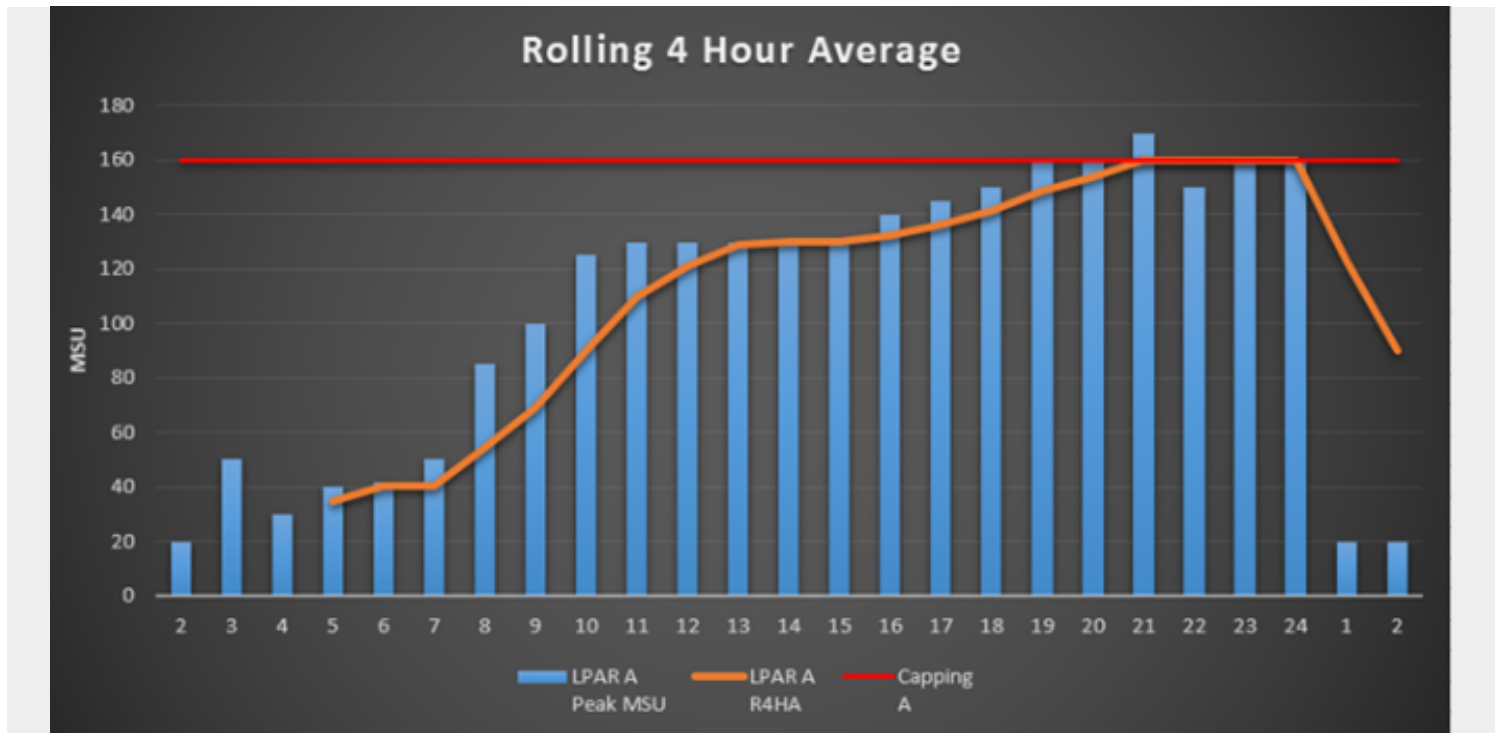


# WHAT IS THE ROLLING 4-HOUR AVERAGE?



Why is the Rolling 4-Hour Average (R4HA) important? Why should you and I care? Developers, testers, system administrators, and other technical professionals are actually very affected by this important metric. For those who are not familiar with the mainframe rolling 4-hour average, this blog helps describe its importance.



Figure 1. Hourly Peak MSU utilization, R4HA and Soft Capping on LPAR A

Take a look at Figure 1. The blue columns are the peak utilization per hour, the orange line is the rolling 4-hour average for the last 4 hours, and the red line is the **defined capacity** or **soft capping** limit.

Whether you are a DBA, system programmer, capacity manager or a developer, it is important to understand that regardless of the type of workload running, the type of code that you are writing, the tables that you support, and the system parameters that you set, all of these factors contribute to the 4-hour rolling average.

## How R4HA works

Here's the critical part: If you are able to lower this metric for your customer, you are directly helping that customer save money—perhaps significant sums of money.

IBM uses its Sub-Capacity pricing/licensing model to charge its customers on the basis of the peak rolling 4-hour average that a customer runs each month. This is also referred to as the Monthly License Charge, or MLC. You can read more about it here - [Seven High-Potential ROI Strategies for](#)

## Reducing Mainframe Costs

Every Sub-capacity product is charged based on the peak 4-hour rolling average on the logical partition (LPAR) that it runs on.



Figure 2. LPARs Running DB2 with their Hourly MSU peaks, R4HA and combined R4HA of both LPAR A and B



Figure 3. LPARs Running IMS with their Hourly MSU peaks, R4HA and combined R4HA of both LPAR B and C



Figure 4. LPARs Running CICS and z/OS with their Hourly MSU peaks, R4HA and combined R4HA of both LPAR A, B & C

## Understanding the math

In the first graph, the peak R4HA for LPARs A and B (which run DB2) is not  $140 + 160$ . It is 293.75.

Similarly, for the LPARs running IMS, the peak R4HA is not 170, but instead 166.25 and for all 3 LPARs together, it is 322.5 instead of just the sum of the individual peaks on the 3 LPARs. So, DB2 will be priced based on 293.75 MSUs, CICS will be charged at 322.5 MSUs, IMS will be charged at 166.25 MSUs, and z/OS will be charged at 322.5 MSUs. This is much more expensive than most think.

All of the LPARs seem quiet between 1 AM and 4 AM, so from a developer's perspective, if you run a poorly designed SQL application during this time, is not much of a concern, compared to running a bad SQL application when all 3 LPARs are peaking. In the above charts, this starts at around 6 PM till midnight.

Hence, it is important to understand whether the code that you are writing will run during the peak processing period or the lean period. This will not only push you to write better code, but it will also help the Mainframe box consume only those MSUs that are appropriate to process the workload and not one bit more.

Similarly, if you have a set of jobs that are currently scheduled to run during the peak processing and consume lot of MSUs themselves, you might consider moving these jobs to a different time. If such a set of jobs is moved by about 30 minutes or so, it can greatly reduce the peak R4HA. If you cannot move the jobs at all, then the current peak R4HA is justified.

The takeaway is this: Being R4HA sensitive will always help you. Every MSU saved during the peak period can significantly reduce the cost of running the same workload, bringing down the overall cost of the mainframe.