# FIVE MYTHS OF MAINFRAME CAPPING





The mainframe operating system provides a mechanism that IT can use to limit the amount of resources that workloads use, and limit exposure to excessive IBM Monthly License Charge (MLC) costs.

Defined Capacity, or capping, can be set at specific levels by IT, but the consequence is that when work reaches the cap, the operating system will not allow any additional MSUs to be used, so workloads will be delayed. The potential for creating service level impacts to critical business work has made many mainframe users reluctant to use capping. As a result, there are some misconceptions about the risks and rewards of using capping. This blog explores five of them.

1. **If I cap, I put business services at risk.**

   An effective capping strategy does not have to put business services at risk. Caps that are setup correctly account for the different importance levels of workloads and ensure that high importance work gets the resources it needs, and may restrict low importance work where delays are acceptable. Workload volumes vary widely over time, so getting "the one right cap" that is safe for all time periods is just not attainable using native tools.

   With the digitally-driven volatile and variable work impacting mainframe systems, an effective capping solution needs to dynamically and automatically adjust capacity limits across LPARs to ensure there is no risk to business services. The solution needs to evaluate workload priority, available capacity and the relative cost of MLC products running on the various LPARs. This should be accomplished under control of a policy you set. One that defines workload priorities, target MSU consumption and the cost of MLC products on LPARs. With this approach, you can mitigate risk to critical work, and achieve lower MSU consumption and lower MLC costs.

   For a more detailed discussion of how to cap without risk, view this webinar .

2. **Capping may avoid excess charges, but it cannot reduce my ongoing MLC costs.**

   This is true of standard manual capping approaches. The volatility and variability of workloads on the system dictate that caps be set to avoid excess charges, but do not constrain any priority work running on the systems. As a result, caps are usually set high to prevent excessive charges and avoid workload impacts. However this eliminates opportunities to reduce costs. Standard capping mechanisms cannot address variability/volatility, differentiate workload priorities, or recognize excess capacity that may exist across the environment.

   However, an automatic, dynamic capping adjustment approach protects against excessive usage and ensures that priority work is not resource constrained, while lowering total MSU consumption, and MLC costs. This approach dynamically adjusts caps to align priority workload requirements and available unused capacity on other LPARs. It moves cap space between them as the variability of the workloads dictates. In shifting excess capacity across LPARs, this capping approach also has to be aware of the relative cost of the MLC products which are running on each LPAR in order to avoid inadvertently increasing overall MLC costs.

3. **Capping doesn't work with variable and volatile workloads.**

   As discussed in Myth #2, a dynamic capping mechanism is the key to actually reducing MLC costs. The automatic, dynamic, workload-aware and MLC cost-aware approach is ideal for handling variable and volatile workloads, which are being driven by digital engagement. A capping approach that examines workload activity and priorities in real time, can make cap adjustments that accommodate workload changes, and balances required service levels, capacity and costs. This approach ensures service quality and cost optimization, even when

workloads are highly variable.

4. **Effective capping takes a lot of knowledge, time and continuous effort.**

   Capping is a complex activity involving the interaction of workload activity, workload priorities, available capacity, LPAR configurations, MLC licenses and MLC costs, to name a few factors. It can be daunting to try to develop a manual capping strategy and keep it up to date with constantly changing workloads. Using an appropriate automated capping solution will alleviate most of the knowledge, time and continuous effort requirements.

   An automated, dynamic capping engine can adapt to changes in workloads and capacity for you. It continuously makes adjustments to ensure service levels are met, and can reduce total MSU consumption, thus reducing MLC costs. It can also provide observational information you can use to make decisions about prioritization and MSU target levels. In addition, if it simulates the capping actions (without actually taking them), you have a combined view of the various complex factors and the capping actions the solutions may take for you. This further reduces the time and effort required for implementing a capping strategy for risk mitigation and cost reduction.

5. **Automation is scary and I can't trust it to manage my critical workloads**

   Delivering availability and performance for critical work continues to be a high priority for mainframe shops, as was reported in the recently released 2016 Annual Mainframe Research from BMC, which can he viewed [here](#) . On the other hand, mainframe IT has been at the forefront of using automation to make mainframes more available, higher performing and more cost effective than other platforms. Mainframe IT uses automation to manage responses to problems, to manage critical data bases and to manage the execution of thousands of jobs. So using automation to control caps should not be any different.

   The concern for putting critical business services at risk is valid, and there have been instances where setting an incorrect cap, or not adjusting cap setting as the workload changed, have created service level failures. So caution may be warranted.

The keys to being comfortable with automated capping are:

1. Make sure the capping approach recognizes workload importance in its capping decisions.
2. Operate under a user-defined policy that aligns automated capping decisions with IT priorities and costs concerns.
3. Require a solution that has a simulation mode that displays capping actions it would have taken, (but did not), so you can become comfortable with how it will manage your workloads and environment.

By using a deliberate approach to implementing an automated capping capability, you can verify the actions will align with your goals and be assured that capping will benefit your workloads and your costs.  More information on an approach to this can be found [here.](#)