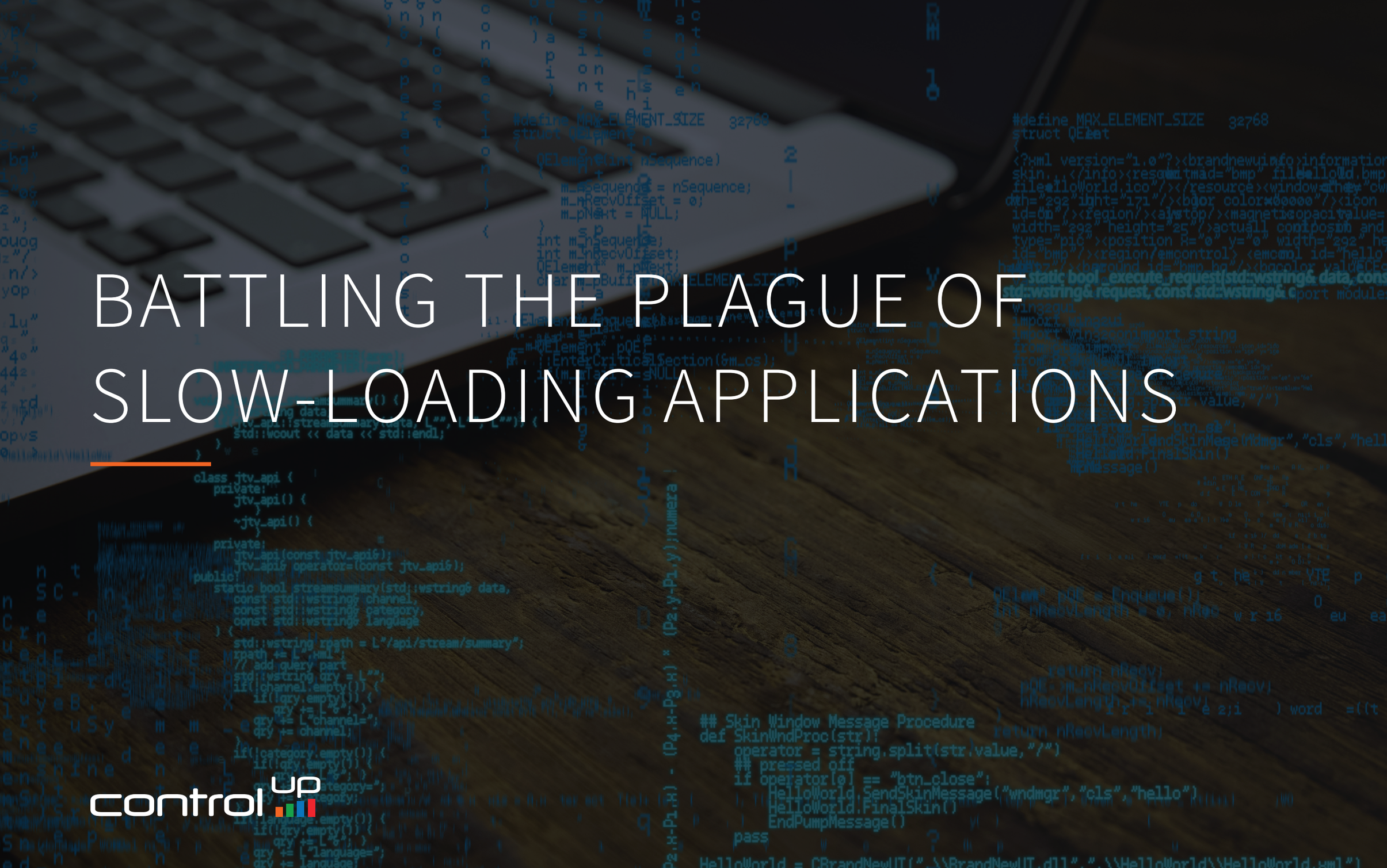


BATTLING THE PLAGUE OF SLOW-LOADING APPLICATIONS

control up

EXECUTIVE SUMMARY

In opening an application, the employee causes a chain of events which almost inevitably result in a delay of the task at hand. Slow-loading applications waste organizational resources, impair the efficiency of customer service, and create frustration among users. In this paper we seek to take a methodical approach to measuring real-world performance and determine causes of delays in app load times.

Our research is based on 116 million individual application launches at 280 organizations over a period of six months. We focused on five most common Microsoft end user applications, and Google Chrome. We studies the top three metrics (average application load time, variations of those times, and how often those applications were launched, within an organization and across the industry.

Lastly, we propose a practical framework IT can use for assessing their users' application load times and taking steps to improve the each application's load time, which will have a positive and direct impact on user experience.

To the best of our knowledge, this is the first-ever publicly available big data research on application load time, and we hope it will aid you in improving your users' IT experience.

Eugene Kalayev

Eugene has been with ControlUp for over 7 years, planning the product's technological vision and translating it into concrete R&D activities. His areas of expertise include virtualization, cloud computing, Windows Server, PowerShell, VDI.

THE IMPORTANCE OF APPLICATION LOAD TIME

An application is the tool a user needs to access and update information, and she typically relies on several applications to perform her day-to-day tasks.

For example, a doctor examining a patient's medical record may click on an icon that represents the results of an imaging procedure. Once the icon is clicked, the operating system invokes the application that is configured to open an image viewer, a word processor, or a video player. Waiting for available processing resources impact each step of application load time: Initializing the application, locating the document, and loading it from disk or network share into memory, and displaying it on screen.

Once an application starts loading, the user has no choice but to to wait for the information to be displayed. In the doctor-patient scenario, this will probably result in a few moments of silence during which both stare at the screen with anticipation, hoping that the delay will not be long. Sometimes the delay may be so short that it goes totally unnoticed and the quick flow of information is taken for granted. However, the unfortunate reality is that applications sometimes take their time to load.

How long a delay is required for the user to start worrying or become impatient? That depends on various factors, but notably on the expectation that is developed from past experience. If the doctor is accustomed to the patient records opening instantaneously, then even two or three seconds of delay may be noticeable and sufficient to cause some concern.

Now let's imagine that more than a minute has passed since the doctor clicked the icon and the patient records have not yet appeared on the screen.

KEY TAKEAWAY:
Slow-loading applications waste organizational resources, impair the efficiency of customer service and create frustration among customers and service providers.

ONE OF THE FOLLOWING IS LIKELY TO HAPPEN

1. Doctor decides to keep waiting, which results in further time waste and may or may not eventually lead to the desired result

2. Doctor tries to click the icon again, potentially restarting the process or spawning another application instance, and eventually causing additional system stress and delay

3. Doctor attempts to rectify the situation by restarting the computer, adjusting the network cable, calling tech support or switching to a different workstation - all of the above may or may not help fix the issue but will definitely waste some more time

4. Doctor gives up and decides to proceed without the test results or opts to reschedule the patient's appointment

THE IMPORTANCE OF APPLICATION LOAD TIME (CONT.)

All of the above scenarios exemplify potential outcomes of excessive application load time as experienced by computer users, and in the doctor-patient scenario - also by others who require their service. Among the possible adverse impacts of those scenarios are:

- Waste of time, which is often the most expensive resource, is easy to see in the doctor-patient case
- Waste of computing resources due to restarted operations or repetitive clicks by impatient users
- Waste of other organizational resources, such as overhead associated with contacting tech support or rescheduling appointments
- Employee frustration, which may cause stress and lead to degraded level of service or poor productivity. Specifically, slow IT systems may contribute to the perception that one is equipped with tools which are inadequate for him to perform his job functions, which may lead to a decrease in job satisfaction
- Customer frustration, which may lead the patient to perceive the level of medical care as poor because of unreliable computer systems

In any organization that relies on computer systems for its day-to-day operations, those risks should motivate decision makers to direct their attention to application load time.

The doctor-patient example is a useful illustration, but it does not imply that application load time is not important in other industries. Given the fact that work-related tasks commonly involve opening an application of some sort, the effects and consequences listed above can be generalized to many industries.

Another noteworthy aspect of slow applications that should be of interest for IT executives is related to the technical issues causing the slowness. Especially when analyzing extreme delays, systems administrators may discover gross misconfigurations or system design flaws which would otherwise go unnoticed. An example of such misconfiguration is improper network placement of the file server hosting the documents, which cause excessive network traffic over inefficient links, thus causing unneeded delays.

As this paper will demonstrate, organizations worldwide are becoming increasingly aware of application load time as one of the Key Performance Indicators of their IT systems. We will present an unprecedented global dataset, which demonstrates load time statistics for common business applications, thus establishing a global benchmark. Moreover, we will survey several action plans which can help improve application load time, while decreasing business risks and improving productivity.

But first, let's explain how application load time is monitored.

KEY TAKEAWAY:
Slowness during application launches may be cause by misconfigurations and system design flaws which often can be fixed, provided that the causes for slowness are attentively investigated.

CONSIDER THE FOLLOWING REAL-LIFE SCENARIOS

During a telephone call from a customer, a help-desk representative for a retail company opens a PDF document in order to access order information, and waits for it to load.

While reviewing a claim, an insurance agent opens a legal document attached to the claim. The document takes minutes to load due to its size and the fact that it is located on a remote file server.

A car rental service representative opens a web application

A court official opens a scheduling application in order to determine the next available date for the next court hearing. Everyone present in the courtroom at that time are forced to wait for the application to finish loading.

MEASURING APPLICATION LOAD TIME

In order to design a maximally precise method of monitoring application load time, we would strive to measure the delay as experienced by the user. That is, an accurate load time measurement tool will need to identify when the requested information is ready for viewing or editing and the user does not need to wait any longer. In the doctor-patient scenario above, this will correspond to the moment when the patient records are displayed on screen and the physician begins examining them. If a monitoring system can identify the precise moment when the waiting is over, then the recorded delay time can be considered a faithful representation of the user's experience.

Alas, the objective of identifying this moment with precision poses several significant challenges for computerized monitoring solutions. In order to be able to closely follow the readiness of the user interface, monitoring software typically requires a level of intervention that affects the monitored activity, sometimes to an extent that the process is delayed unnecessarily or encounters compatibility issues. This constitutes a fundamental dilemma for monitoring in general - how to obtain accurate and detailed information without breaking or slowing down the actual monitored activity.

Why is it so hard to identify the critical moment when the application has finished loading? While the complete technical details for this issue are beyond the scope of this article, the fundamental reasoning is as follows: in order to determine that an application is ready, there's a need for a standardized event reported to the operating system, saying something along the lines of "user interface is ready".

The reality is that common business applications are built using a variety of programming languages, are based on various development platforms, and interact with the UI in different ways. One unfortunate consequence of this diversity is that there's no single and standard "ready" event recorded by the application or the operating system. Although there are several systems events that can be leveraged to deduce that the UI is ready, utilizing them as application load time markers is problematic due to the following reasons:

- Sometimes the user perceives the application as still loading even though the "UI ready" event is reported by the application
- In other cases, from the user's perspective the application is in fact ready before the "UI ready" event is reported

KEY TAKEAWAY:

Application Load Time as measured by ControlUp captures the time interval between opening an application and being able to interact with it.

MEASURING APPLICATION LOAD TIME (CONT.)

- Implementation of “UI ready” events by different software packages is not standardized and therefore inapplicable for comparing application load time across different software suites, platforms and operating systems

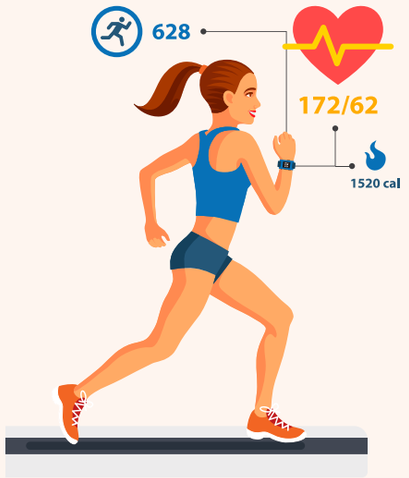
In this section, ControlUp’s methodology for monitoring application load time is outlined for the sake of clarity. This article does not intend to claim that this methodology is the only one, or the best one, for monitoring delays experienced by users when launching applications.

ControlUp monitors application load time by using a heuristic model based on process performance metrics. This model assumes that application launch is associated with a bout of activity (CPU, network or disk I/O) that is expected to quiet down once the application is fully initialized and awaiting user instructions. For example, when opening an Excel spreadsheet, your computer is expected to consume some CPU cycles to load the application binaries, and then some more CPU cycles on loading the spreadsheet file. When the Excel window with the spreadsheet is fully loaded and usable, the application process typically becomes idle and proceeds to consume little or no CPU cycles. In summary, ControlUp considers an application fully loaded when its initial bout of resource consumption has decreased.

This heuristic approach enables ControlUp to passively obtain load time measurements which are close to the delays measured by human observation, without endangering the stability of monitored applications or slowing them down unnecessarily.

For the applications in this sample, this assumption was supported by thorough testing. In the simplest terms, the best way to validate ControlUp’s measurement is to take a stopwatch, start it when the application icon is clicked and stop it when the user interface is readily available (e.g. when you are able to edit the spreadsheet by typing on your keyboard).

It’s worth noting that some applications (such as a browser streaming a video) may continue consuming significant system resources long after they are fully loaded from the user’s perspective. Such applications should be examined on a case-by-case basis to determine the appropriate load time monitoring alternatives. For those situations, ControlUp relies on a proprietary algorithm which distinguishes the characteristic pattern of activity that occurs when the application is loading, from the resource consumption pattern which continues afterwards.



KEY TAKEAWAY:

Reliably measuring application load time for a wide range of applications has proven a challenging and complex task. ControlUp’s heuristic algorithm produces results that are consistent with actual delays experienced by users, for a variety of applications.

MEASURING APPLICATION LOAD TIME (CONT.)

Let's describe the basic phases of application load and explain how ControlUp determines the boundaries of the application loading process:

Phase I

User clicks an icon representing an application or a document associated with a known application. The operating system determines which application is required and what is needed to launch it

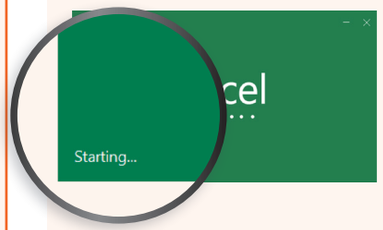
Phase II

Operating system locates the application process by following the executable path, and invokes the application process

[ControlUp begins the app load time "stopwatch" at this time, AKA process start time]

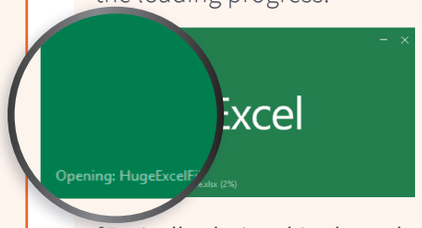
Phase III

The applications starts to load, typically displaying a "splash screen", for example in the case of Excel.



Phase IV

After the application itself has initialized, it starts reading the document from disk or a network location into memory. Some applications will display an indicator which notifies the user on the loading progress.



[Typically, during this phase the application is not yet accessible to the user. Some applications may show the user interface but not allow the user to click it. Commonly, the mouse cursor appears as an hourglass throughout phases III and IV]

Phase V

The application allows the user to interact with the UI. At this point, we are expecting all operations associated with the application's initialization and document loading to be complete.

[ControlUp considers the application fully loaded at this time, calculates the load time in seconds and displays it in the Processes view]

ControlUp's monitoring approach was designed to enable IT professionals to identify major delays that are likely to be noticeable by the end user. For that reason and to make the findings easier to interpret, the results will be reported on a granularity level of seconds. For some purposes, such as when assessing responsivity of high end real-time applications, a millisecond level may be desired. Those advanced cases will remain out of the scope of this paper.

DESCRIBING THE DATASET

ControlUp is an IT monitoring troubleshooting and analytics solution deployed in hundreds of organizations worldwide. For the purpose of historical analysis, benchmarking and troubleshooting, application load time data is recorded along with other performance metrics in a global big data warehouse. The accumulated data permits us, with the consent of our customers, to publish anonymized statistics and research findings based on large representative samples.

The dataset reported in this article includes data gathered by ControlUp from 280 organizations worldwide, including companies from various industries, sizes, and geographical regions.

The statistics reported below are based on more than 138 million application instances. As a rule, each application instance corresponds to a single launch of an application, typically triggered by the user's request to open a document (e.g. web page, spreadsheet, Word document, etc.). In technical terms, each application launch corresponds to a Windows process that was started.

All the applications included in the dataset are based on Microsoft Windows. This is due to the fact that as of 2018, the Application Load Time monitoring feature in ControlUp is only available for the Windows operating system. ControlUp enables systems administrators to enable Application Load Time monitoring for specific applications, while the following applications are monitored by default:

- **Internet Explorer**
- **Excel**
- **Word**
- **Outlook**
- **PowerPoint**
- **Google Chrome**

The above applications comprise the lion's share of data (more than 116 million launches), and the analysis below will focus predominantly on those applications.

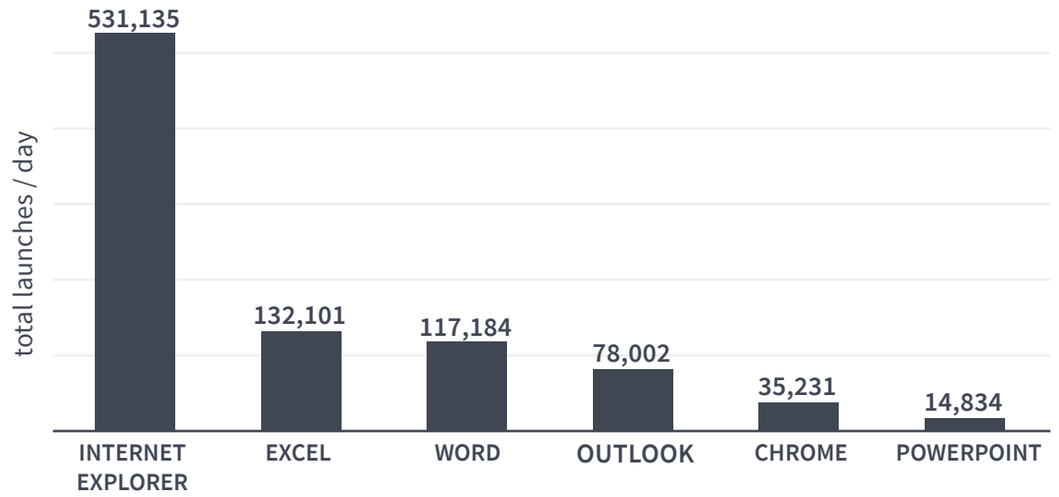
DESCRIBING THE DATASET (CONT.)

The chart below demonstrates the average frequency with which those applications were launched per day in all the entire sample:

In addition, systems administrators from 19 organizations have enabled load time monitoring for a total of 169 other applications, which contributed over 21 million additional results to the dataset. Due to the fact that by and large those applications were monitored by a small number of companies, we will defer our discussion of those results in order to prevent disclosing information that can be used to identify specific customers.

The dataset was gathered between June and December of 2017.

POPULARITY BY APPLICATION



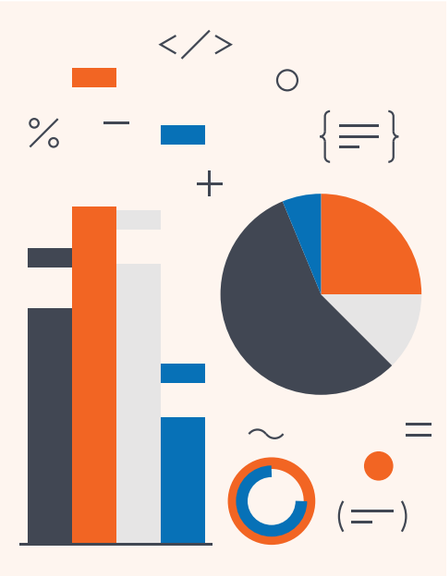
KEY TAKEAWAY:

This article reports on Application Load Time statistics gathered from 280 organizations worldwide over the course of six months. This dataset contains over 138 million launches of common workplace applications, such as browsers and Microsoft Office.

METRICS DEFINED

In order to summarize the load time statistics obtained for each application in every organization in the sample, let's define several key metrics:

- **Average Load Time** - the average delay, in seconds, for all launches of a particular application in a given organization. This metric is represented by the Y axis on the charts below (note that the axis values are reversed so that apps that are faster to load appear in the top area).
  - For example, in the first chart below Microsoft Word in Organization A had an average load time of about 2 seconds.
- **Load Time Standard Deviation** - the degree of variance between load times of a specific application in a given organization. When monitoring user experience, this metric indicates the degree of consistency - the lower the standard deviation, the more consistent are the load times. This metric is represented by the X-axis on the charts below (note that the axis values are reversed so that apps with lower standard deviation / higher consistency appear to the right).
  - For example, in the first chart below Microsoft Excel in Organization B had a load time standard deviation of almost 5 seconds.
- **Average number of application launches per day** - the average frequency of use for each application in every organization. This metric is represented by the size of bubbles in the charts below - the large the bubble, the more instances of the application were observed for the given organization, on average for all days for which data was gathered
  - For example, in the first chart below Microsoft Outlook in Organization C was launched on average about a thousand times a day.
  - Note: the average number of application launches per day is reported in order to provide a straightforward measure of size that is indicative of the monitored user population in each organization. It is not to be used to deduce the number of employees, computers or branch offices in each organization. For example, for the purposes of describing this dataset, an organization with 10,000 launches of Internet Explorer per day is considered larger than a company with 2,000 launches per day, regardless of the actual size of the respective companies in which those results were obtained.



ALL APPS

For the purpose of demonstrating the complex results, consider the visualization on the next page.

As explained above, each bubble represents all load time results for a single application from a single organization in the sample. The chart background was colorized in order to map the obtained results to a user experience scale, so that results in the green area represent optimal user experience (low load time, low standard deviation), the red area corresponds to poor user experience (high load time, high standard deviation), and the yellow area roughly represents average results.

The bubbles in the chart vary in color according to the application they represent, e.g. green bubbles represent Google Chrome. Before drilling down into each application, it is worth noticing the general distribution of averages in the chart above. Consider the following general notes in order to calibrate your perception of this chart:

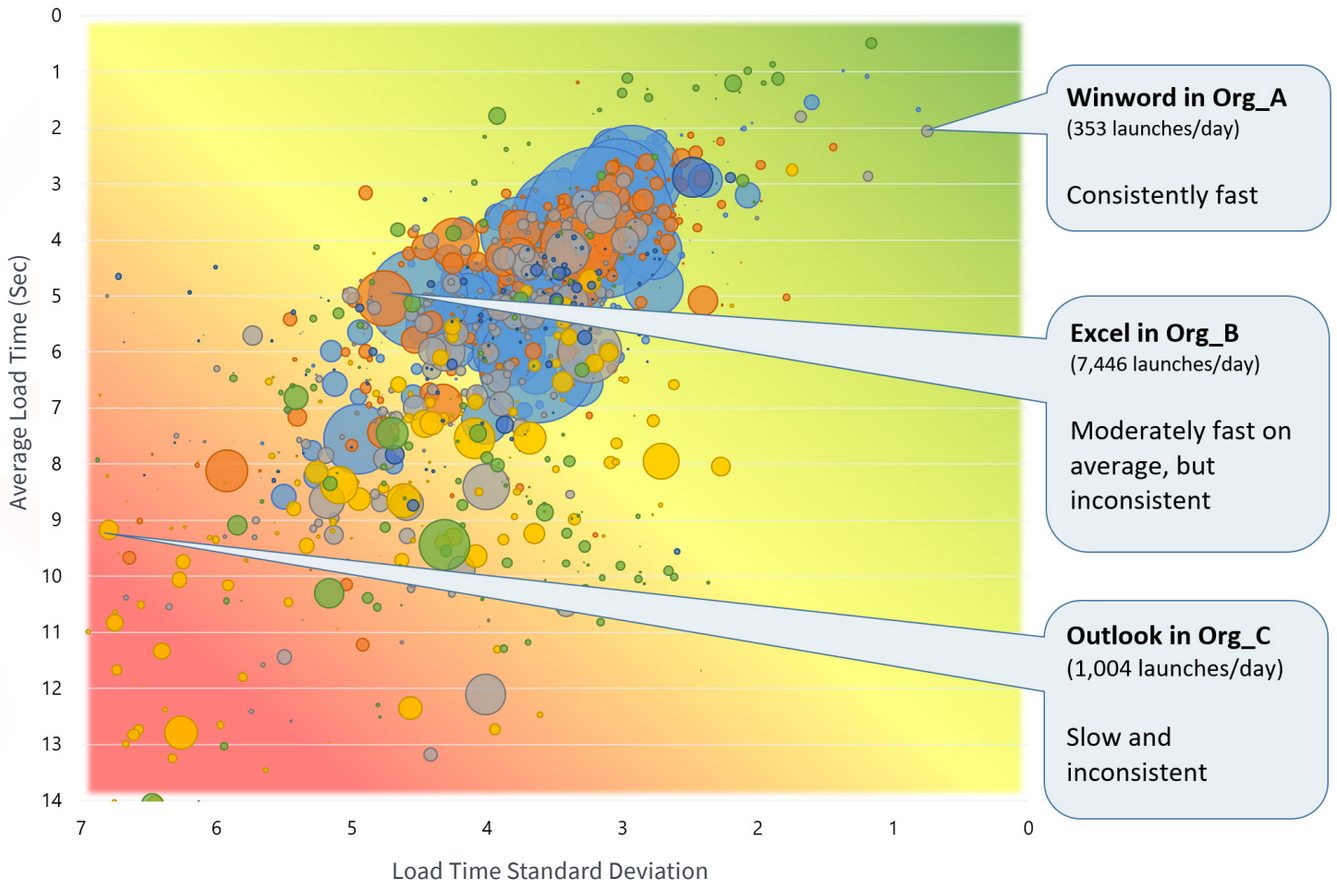
- The fastest-loading applications have an average load time that is faster than 4 seconds (their bubbles appear above the 4 second line).
- The fastest averages were obtained from smaller organizations (recall the caveat explained above regarding the organizational size as defined here). This is hardly surprising, because the more samples are gathered in a given organization, the more likely this sample is to include high results. In other words, when measuring UX statistics the IT department should expect results that deteriorate gradually as the number of results increases. This also means that the larger the organization, the harder is the task to maintain an IT environment in which the average and standard deviation stay low.
- Internet Explorer (represented by blue bubbles) was a relatively popular application in this sample. In almost a third of the organizations in the sample, IE was launched over 1000 times a day on average, which explains the relatively large size of the blue bubbles.
- Outlook appears to exhibit high average load time and deviation (shown by yellow bubbles).

Hopefully, the notes above have clarified somewhat the visualization logic of the chart. In the next sections, let us drill down into the results obtained for each application and examine them when viewed separately on the same chart.

ALL APPS (CONT.)

DURATION AND CONSISTENCY OF APPLICATION LOAD TIME

(Org averages, bubble size = avg daily launch count)



INTERNET EXPLORER

Average	Standard deviation	252 organizations
5.04 seconds	3.96	81M launches

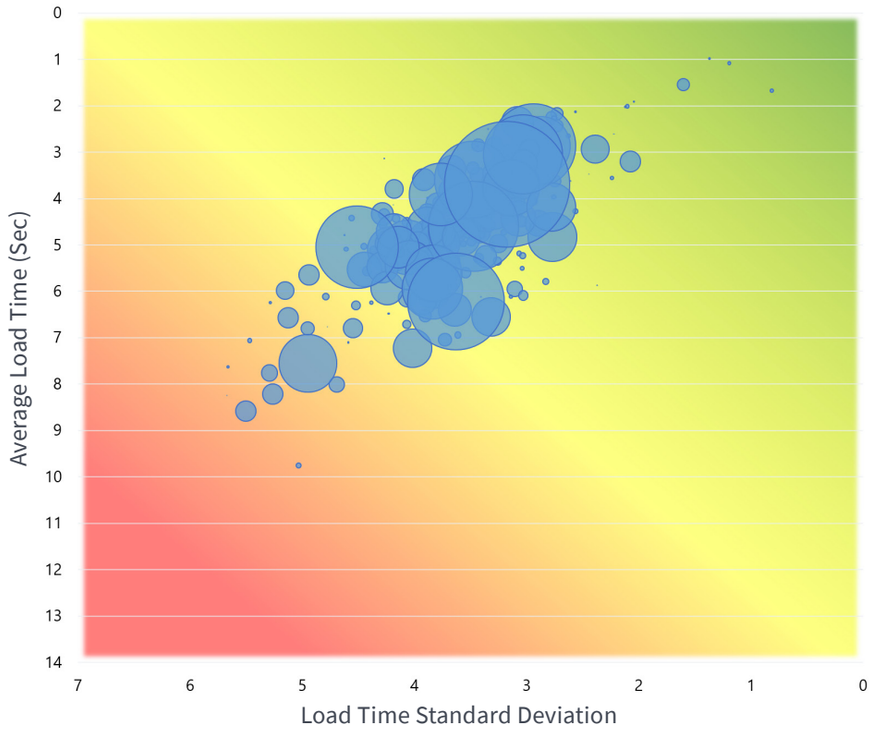
The popularity of Internet Explorer launches in the sample is in itself a point that deserves elaboration. The results suggest that although its share in the consumer market is decreasing, Internet Explorer is still the leading browser in use in virtual desktops and as a published application. It is important to consider that in those scenarios, the browser is commonly used in order to provide access to web-based applications hosted on the corporate Intranet.

The current research does not partition the results by type and location of the web-based resource launched in the browser. However, the typical VDI use case may be different from the set of online activities performed by users at home or on their own machines. For example, employees may be less likely to perform activities such as playing online games or using social networks on their virtual desktops.

Note that organizational load time averages for Internet Explorer range between 1-2 seconds in the fastest cases, while the slowest averages are above 8 seconds.

DURATION AND CONSISTENCY OF INTERNET EXPLORER LOAD TIME

(Org averages, bubble size = avg daily launch count)



KEY TAKEAWAY:

In virtualized desktop environments, Internet Explorer is widely used to access web applications on the corporate Intranet, as well as other web-based resources.

INTERNET EXPLORER (CONT.)

Note that organizational load time averages for Internet Explorer range between 1-2 seconds in the fastest cases, while the slowest averages are above 8 seconds.

As with any software used at the workplace, the IT department typically controls different Internet Explorer settings by means of Group Policy, scripts or third-party user environment management solutions. Here are just a few examples of controls that have the potential of speeding up browsers, for example:

- Unneeded browser add-ons and extensions can be disabled centrally, and the list of allowed extensions can be enforced.
- Browser cookies tend to accumulate and slow down I/O operations due to the fact the OS needs to read a large amount of small files from disk in order to initialize. Profile management software and other user environment management solutions can help reduce the amount of cookies to the minimum required in order to retain essential functionality of web-based business applications.
- Similarly, a bloated browser cache can affect performance. Implementing a centralized solution for keeping the amount of files in the users' browser cache folders under control is a good idea for most IT environments.
- The browser may be configured with a home page (or a number of home pages), which can include non-essential content. If time is of essence when the browser is opened, consider enforcing a lightweight home page or a blank one, when applicable.

KEY TAKEAWAY:

Organizations in which the average load time for Internet Explorer is above 5 seconds should consider taking action in order to improve user experience.

MICROSOFT EXCEL

Average	Standard deviation	245 organizations
4.76 seconds	4.15	22.9M launches

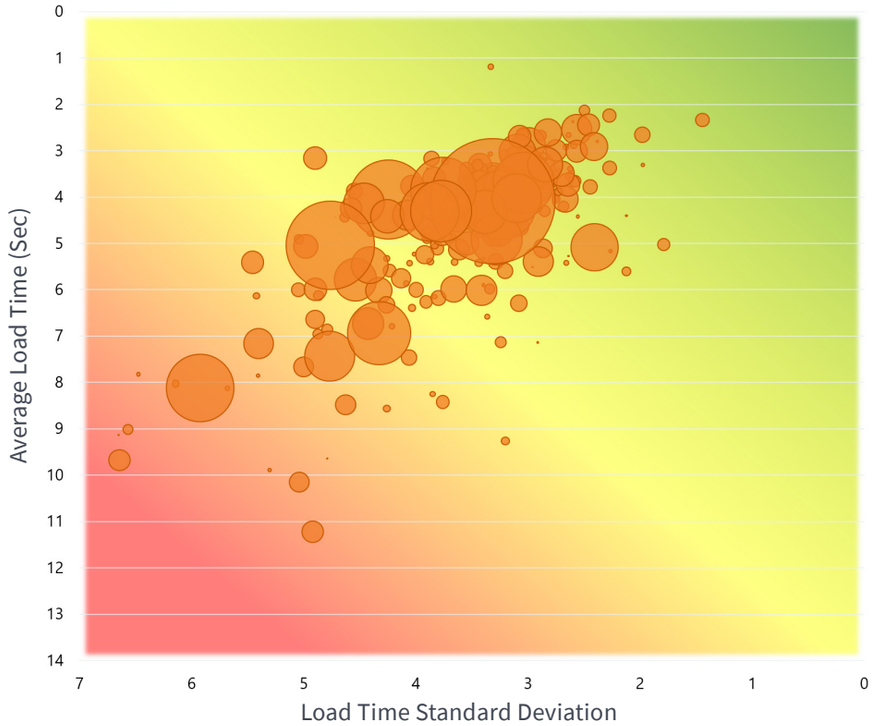
The load time findings for Microsoft Excel are somewhat different from the results obtained for Internet Explorer. Notice how the bubbles representing the different organizations in the sample are more spread out on the x-axis of the chart. This indicates that there are noticeable differences in the standard deviation of load times between different customers, or in other words - that Excel launches are likely to vary in their duration.

Excel spreadsheets may include large amounts of data or a wealth of complex formulas that need to be recalculated when the sheet is opened. Both those factors are likely to delay load time for Excel documents, in addition to the general factors to be discussed later in this document. There are a number of Excel-specific techniques that enable for some savings in resource consumption, notably the following:

- For one-time calculations that do not need to be refreshed every time, copy the formula results and paste them in place as values.
- Disable automatic calculation for your spreadsheet.
- For analytic spreadsheets that include statistics calculated from large amounts of raw data, consider dropping the raw data and retaining only the calculated stats, where appropriate.

DURATION AND CONSISTENCY OF MICROSOFT EXCEL LOAD TIME

(Org averages, bubble size = avg daily launch count)



KEY TAKEAWAY:

To shorten load time for complex Excel spreadsheets, reduce the amount of calculated cells and / or disable automatic calculation.

MICROSOFT WORD

Average	Standard deviation	233 organizations
6.69 seconds	5.48	22.4M launches

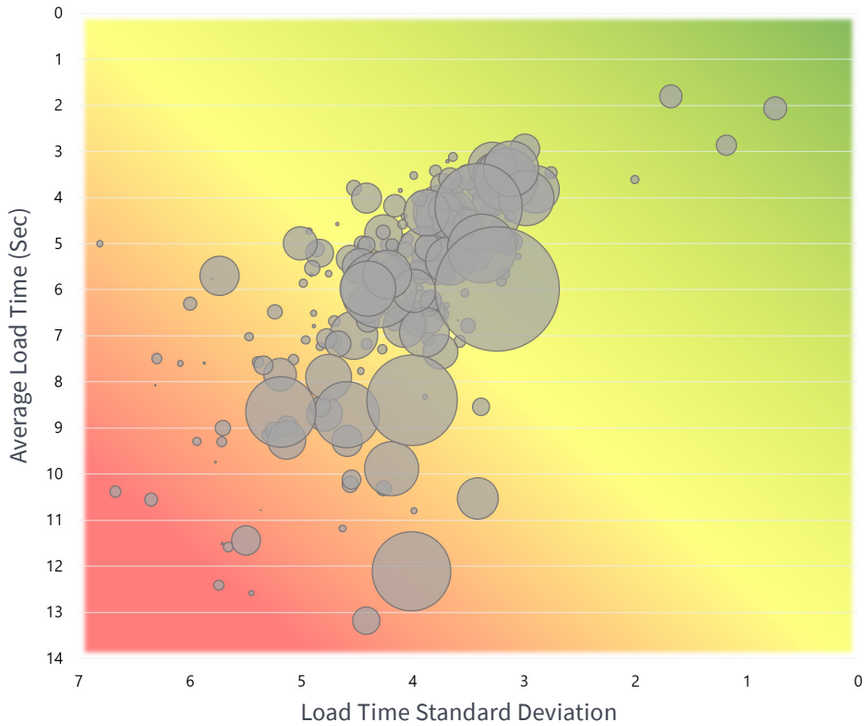
When compared to Internet Explorer and Excel, the load time statistics for Microsoft Word in this sample suggest an even larger variance. This chart demonstrates that it is not uncommon to encounter average load times of over 8 seconds, and in some large organizations the average load time can be as high as 12 or 13 seconds.

Like Excel spreadsheets, Word documents can sometimes contain large amounts of data, which prolongs their load time. There are multiple settings specific for Microsoft Word that are likely to improve its performance, for example:

- Some documents can be optimized by reducing the number of fonts and graphics, or by reducing the resolution of embedded images. As a somewhat extreme measure, the “Show Picture Placeholders” option can be used to disable automatic loading of pictures.
- Automatic grammar and spelling checks can be disabled as an optimization technique.
- The “Recently Used File List” is known to delay document load time, and can be disabled.
- Using Draft mode, disabling background repagination and disabling change tracking (especially for formatting changes) have been reported to improve Word performance.
- Word offers a variety of printing options that are available for tweaking, some of which can improve document load time.

DURATION AND CONSISTENCY OF MICROSOFT WORD LOAD TIME

(Org averages, bubble size = avg daily launch count)



KEY TAKEAWAY:

Word settings include a variety of behaviors that can be configured in order to speed up load time for documents. Most of these settings can be orchestrated centrally using Group Policy.

MICROSOFT OUTLOOK

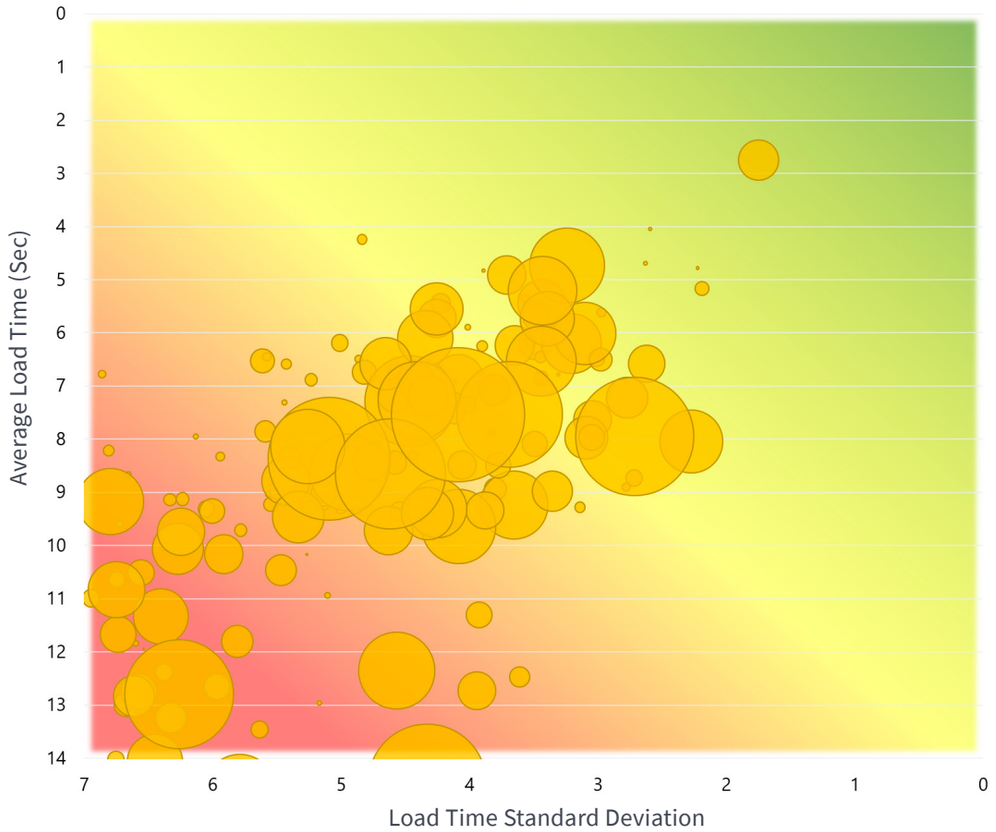
Average	Standard deviation	219 organizations
10.91 seconds	8.3	15M launches

The overall load time for Outlook was the highest among the applications measured in this sample. In only a handful of organizations Outlook exhibited an average load time of less than 5 seconds, and an average delay longer than 10 seconds was not uncommon.

Given the role of email as an essential tool at the workplace, those results may have worrying consequences for the productivity of computer users. From a technical perspective, the prolonged load time can be explained by the fact that a typical Outlook launch involves loading the outlook data file (also known as PST), which serves as a database for Outlook. This file tends to grow in volume as emails accumulate, and a large data file may extend the time Outlook takes to load. Various optimization techniques for Outlook involve manipulations of the data file, such as cleaning it with AutoArchive or compacting it. Other optimization techniques specific to Outlook include disabling non-essential RSS feeds and disabling unused email accounts.

DURATION AND CONSISTENCY OF MICROSOFT OUTLOOK LOAD TIME

(Org averages, bubble size = avg daily launch count)



MICROSOFT POWERPOINT

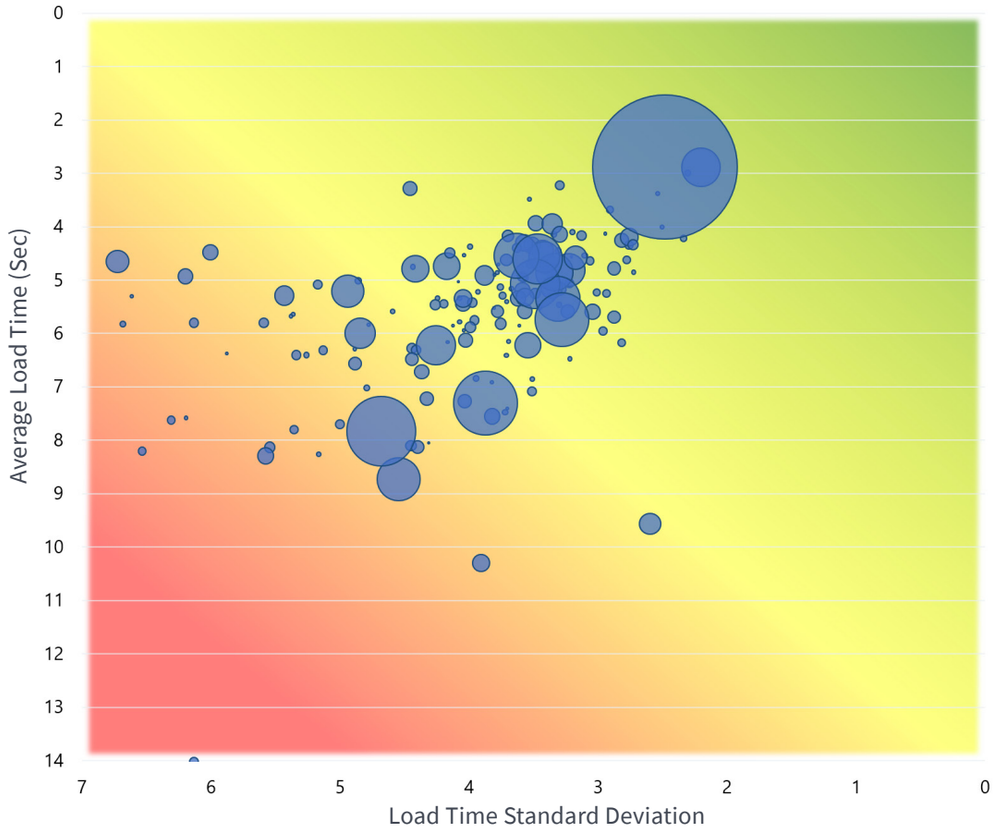
Average	Standard deviation	178 organizations
5.56 seconds	4.77	2.3M launches

The results obtained for PowerPoint suggest that this application is relatively quick to load in most organizations. High load time averages are rare, and for the bulk of the sample, organizational averages range between 4 and 8 seconds, and averages below 4 seconds are very rare. This may mean that the 4 second mark is a good estimate for a realistic improvement goal for PowerPoint load time, and obtaining a better organizational average will probably be difficult.

The wide horizontal spread of bubbles in the chart above suggests that there is considerable variance between load time of particular PowerPoint instances, or in other words - that even in organizations with low average load times there are measurements that are much higher than the average. So even though the organizational average may be relatively low, some users may experience slow load times and therefore may benefit from systems tuning that will improve their experience.

DURATION AND CONSISTENCY OF MICROSOFT POWERPOINT LOAD TIME

(Org averages, bubble size = avg daily launch count)



GOOGLE CHROME

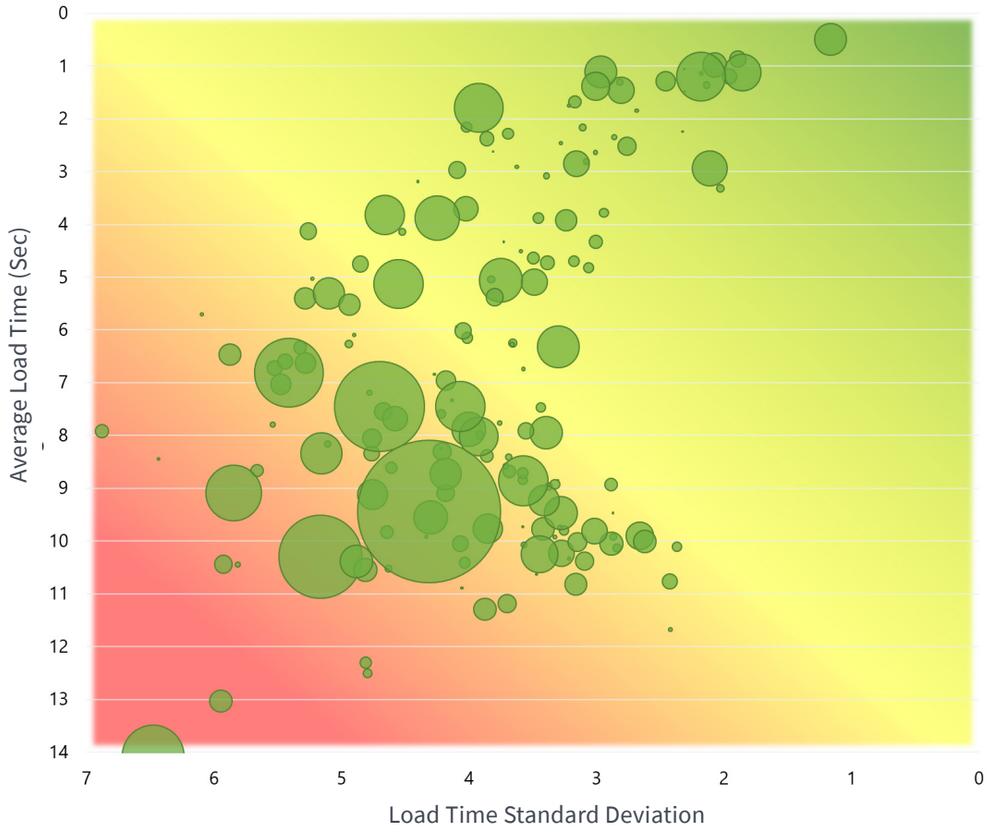
Average	Standard deviation	167 organizations
8.31 seconds	5.21	7.3M launches

The popularity of Google Chrome is an established reality in the browser market. The current sample is not an exception, even though it focuses on virtual desktops and published applications. In such environments, users typically do not have the privileges to select their browser of choice and are limited to the browser selected by their IT department. Therefore, the adoption of new and non-default applications is somewhat more conservative and slower when compared to the home market or to personal workstations, on which users are more free to use their applications of choice. This may explain the finding that Internet Explorer is still fairly common in virtualized workspaces.

As for load time, the organizational averages for Google Chrome exhibit a considerable variance. Organizational averages as low as 1-2 seconds and as high as more than 8 seconds were common in this sample. For an IT administrators those are good news - even if Chrome is slow to load in a given environment, it is probably realistic to reduce the average load time considerably.

DURATION AND CONSISTENCY OF GOOGLE CHROME LOAD TIME

(Org averages, bubble size = avg daily launch count)



GOOGLE CHROME (CONT.)

The following techniques are examples of optimizations that may assist with this task:

- As mentioned above for Internet Explorer, disabling unneeded plugins, extensions and web apps, as well as reviewing the home pages the browser opens automatically may remove some logic that may be slowing Chrome down
- As with other browsers, managing the cache and cookies storage may affect browser performance, especially if those locations have accumulated a large amount of files
- Some browser settings that may affect load time include “Continue running background apps…”, “Use hardware acceleration”, “Use a prediction service…” and others. The effect of those settings is largely dependent on environmental factors, and is therefore to be considered on a case-by-case basis.
- The chrome://flags page includes a wealth of advanced tweaks which may also affect browser load time. The effects of changing those settings may also vary, and should be considered carefully.

DISCUSSION

This sample is based predominantly on results from virtual desktop deployments, in which applications are delivered by the IT department from a centralized datacenter or a set of datacenters. Therefore, most of the variance in load times is likely due to factors which are under the immediate control of the IT department, or at least should be. If the sample was to deal with decentralized computing environments, such as applications launched locally on physical desktops or laptops, then identifying any reasons for slowness and acting upon the findings would have been more complex. The administrators of VDI deployments have a relatively greater degree of control over the technical aspects of application usage, such as network topology, storage architecture, user environment management and allocation of computing resources.

So what are the common factors likely to make a difference for application load times? In the following sections, let's outline some potential aspects of user activity, as well as some technical factors that are likely to contribute to variance in application load time. Those factors will then be linked to some general directions which systems administrators can explore in order to improve user experience in their domains.

App or Document?

For applications that involve documents or web pages (all but Outlook in this sample), one basic decision that a user makes is whether to open an application and then decide which document to open (click on the Word shortcut and then use the file menu to browse for the needed file), or to locate the requested document, and double-click it, thus invoking the application implicitly (open My Documents, double-click a Word document). A similar decision applies to browsers - whether a specific web page is opened automatically or the browser is opened with a blank page, and then a URL is typed or a bookmark is clicked.

Obviously, opening an application without a document will be faster. This difference definitely accounts for some variance in our results, although the results were not partitioned by type of launch.

This factor is an artifact of user preference, and is not likely to be controlled by the IT department. However, it is advisable to consider the typical work pattern when interpreting app load time results. One constructive way to utilize this distinction is to perform isolated tests on the application alone, without involving any documents.

DISCUSSION (CONT.)

Local or Remote?

In a typical enterprise IT system, the users access a variety of documents. These documents may be stored in one of the following locations:

- Locally on the workstation or server on which the application is hosted
- On a file share or web server on a network segment accessible with low latency and high throughput
- On a remote file share or web server, accessing which requires traversing WAN or VPN links, characterized by high latency, low throughput, and/or high error rate

The closer the document is located with regard to the application, the faster it can be expected to be read and loaded. Therefore, the location of documents opened by the users is an important factor to be examined when investigating slowness during application launches.

Multimedia Content

The variety of media types included in documents and web pages is growing constantly to include content such as high-quality graphics, animations, video, and more. The richer the content of the accessed document or web page, the more processing resources are required in order to load it. The IT administrators may not have total control over the content of all documents accessed by users, but some policies may be implemented in order to monitor and enforce the usage of specific content types, especially if those are affecting application performance. One such example is the setting that determines whether to load images automatically in Word documents.

An additional point to consider in this regard is the increasingly popular usage of GPUs in work environments. There are many cases in which graphical processors may offload CPU tasks and enhance user experience, thus also speeding up application load time, especially when the content is rich with multimedia.

Hardware Resources

It is perhaps trivial to note that the amount of available processing resources is linked to the performance of a computer system. Specifically in the case of application load time, the following resources should be considered:

- **CPU** - due to the bursting nature of computation bouts associated with application launch, it is important that the workstation or server have is able to allocate sufficient

DISCUSSION (CONT.)

processor cycles in order for the loading process to complete in a timely manner. Even if the average CPU consumption for the given system is low, spikes due to multiple concurrent requests to launch applications can cause CPU scheduling issues. In this context, performance counters like “CPU Queue Length” and “CPU Ready” for virtual machines are important ones to monitor in order to detect inappropriate resource allocation.

- **RAM** - this resource may affect application load time, especially when there’s a shortage of it, which is forcing memory management mechanisms such as paging to kick in. It is recommended to monitor RAM usage patterns for common applications and plan resource allocation so that sufficient RAM is provisioned for all application instances which are expected to load concurrently.
- **IOPS** - another valuable resource, which in cases of scarcity can cause slow load times. The operating system needs to access the disk to read the application executable bits, then the program’s settings and user customizations, and finally the document itself. As in the case of CPU, I/O tends to be bursting so it is important to closely monitor the I/O activity associated with application launches and use the findings to provision appropriate resources.
- **Network** - the computer on which the application is launched often needs to access the network, whether in order to retrieve data to load the application itself, or to read the document or web page requested by the user. The system design should provide sufficient network bandwidth, low network latency, and minimal error rate, in order to minimize network-associated delays.
- **GPU** - more and more applications and OS components are benefitting from graphic acceleration. Therefore, provisioning GPU resources to virtual workloads may hold the promise of speeding up load time for some applications.

It might be the case that the resources listed above are sufficiently provisioned to the computer on which the application is running. However, there may be a resource bottleneck on another (remote) system which is involved in the application initialization process, for instance a file server hosting documents. If this server is experiencing I/O congestion, the document is likely to take its time loading, even though the application itself is hosted on a properly sized workstation or server. Therefore, as with any performance issue, the systems administrator should pay attention to the weakest link in the entire delivery chain.

DISCUSSION (CONT.)

Peak Times

The level of activity in a typical organization's IT systems varies with time of day, so that bursts of application launches may occur in the mornings as employees arrive at their desks, while the number of application instances open concurrently is likely to peak in the middle of the work day. The difference in work patterns during those times may mean that even though application load times look acceptable on average, they may be suboptimal during peak times. Note that some applications are only launched once in the beginning of the work day or shift, so the analysis of their load time should concentrate on those peak hours.

It is therefore important to examine how application load time varies over the course of the work week. If slowness tends to appear during specific hours, measures should be considered in order to provision extra resources to allow fast load time even during peak hours.

NEXT STEPS

The objective of the current research was to shed light on real-world statistics of load times for common workplace applications, to analyze the main factors affecting load time, and to suggest general directions that may assist IT departments in optimizing their environments for the performance the users expect.

Hopefully, the findings have inspired you to take an interest in the application load time state in your organization. Here is a basic framework for assessing this aspect of user experience in your organization's using a step-by-step approach.

Step 1 - Ask Around

To make a preliminary assessment, you can perform some easy tests, such as asking your users whether they have experienced slowness while opening applications. The answers may vary and are subject to bias due to differences in subjective definition of slowness. However, if even a small fraction of your surveyed users answers affirmatively, you should consider deepening your research.

Another important question is what is the impact of application load delays for your users, or in other words - how much does the delay damage the day-to-day productivity. The answer to this question may depend on the organizational role of the respondent. As an IT decision maker, you may want to utilize the different answers in order to focus on a subgroup of employees for which the impact of delays is the highest.

Step 2 - Define Slow

At this step, we suggest using the most basic measurement equipment for assessing user experience - the stop watch. By measuring manually how long do your applications take to load and comparing different measurements obtained at different times and in different environments you can define what "slow" and "fast" means in your environment.

By establishing your own subjective boundary between acceptable delays and unacceptable ones, you make an important step towards improvement. Now it's time to expand your dataset.

Step 3 - Go Pro

ControlUp's capabilities for monitoring application load time have a proven track record in the industry. While various measurement tools exist that can provide a logon delay measurement, ControlUp provides an easy and accurate way to obtain user experience metrics which have a high correlation with delays measured from the user's perspective.

After some exploratory testing, it's time to start measuring load times continuously, while aiming to provide a full coverage for the target user audience. If your organization has typical seasonal work patterns, it may make sense to monitor load times continuously for at least a month in order to obtain a sample that represents variations over time.

Step 4 - Start from the Extremes

Once you have obtained some measurements and analyzed them by means of a historical reporting solution such as ControlUp Insights, you can start investigating the findings. If you focus on the most extreme results such as applications that take longer than a minute to load (and you are bound to have some, given a large enough sample), you are likely to discover gross misconfigurations or resource shortages which you can address easily.

CONTACT US

We can help you measure, monitor and maximize logon performance, and optimize resource utilization.

It literally takes 15-minutes to install ControlUp in your environment.

or