RESEARCH IN PROGRESS

RAPID SCALING OF A DANISH PUBLIC HEALTH SYSTEM UNDER COVID-19

NICKLAS FREDERIKSEN, ERIK L. MØLLER, JARL TUXEN,
SARAH E. O'NEILL, MORTEN BOESEN

Copenhagen School of Design and Technology, Copenhagen, Denmark
nifr@kea.dk, erlm@kea.dk, jart@kea.dk, saro@kea.dk, mobo@kea.dk

In recent years cloud infrastructure services have acted as engines for scaling applications when user demand spikes. A discipline typically recognized as complex, expensive, error-prone, and time-consuming. In the field of healthcare services, data is considered sensitive under the European Union's data protection law and are therefore under strict jurisdiction disallowing the Danish public services to utilize cloud scalability.

During the COVID-19 lockdown a small group of expert practitioners was tasked with scaling public health services to accommodate an exponential number of excess users who needed to access test results and immunity passports. An effort further restrained by a severely limited timeframe of two weeks.

By utilizing the critical incident technique this paper is an effort empirically to capture the most significant decisions in the scaling process including organizational aspects, virtualization, content delivery network, lazy-loading, and firewall interface configuration.

Keywords:
scalability, health information systems, rapid scaling, sensitive data, Bled eConference

University of Maribor Press

DOI https://doi.org/10.18690/um.fov.6.2023.46
ISBN 978-961-286-804-8

1 Introduction

Scaling information systems is an important discipline for both academics and practitioners from various industries. The discipline can be defined as the process of expanding in scope or size (for example increasing the number of features or the number of end-users) (Sahay & Walsham, 2006). Scaling has been studied from various theoretical perspectives and with different aspects in focus, including the technical, organizational, and institutional aspects. It has been noted that the inability to scale is not caused by technological problems alone. Often, the difficulties are caused by organizational or managerial issues such as confusion of roles, people making bad decisions, and lack of attention to organizational and technical implementation (Abbott & Fisher, 2015).

With the COVID-19 pandemic, the rapid scaling of a public health information system suddenly became critical for the Danish public. Citizens were vaccinated and tested regularly since it became required to document one's status when traveling and physically attending restaurants, shops, and other public areas. Consequently, a digital corona passport was introduced and the national website for health information sundhed.dk was updated with additional features to provide digital vaccination or test certification. Within two weeks, a time constraint was imposed by the government and sundhed.dk had to scale a digital infrastructure from accommodating a rough estimate of a few hundred daily user logins to 5500 concurrent users with a recorded peak load of 1.4 million daily logins. Modern online cloud services such as Amazon Web Services usually accommodate unanticipated and immediate requirements for rapid scaling infrastructure. Still, due to regulations concerning sensitive health data – this was not an option.

This leads to a research question: What critical technical and organizational decisions enabled the rapid scaling of sundhed.dk within two critical weeks during the COVID-19 pandemic?

2 Related literature

In the literature, scalability covers a wide variety of subtopics. Within the organizational part of the information system literature, it is argued that scaling is not only a technical matter but also involves huge managerial efforts if a company wants to scale successfully. Abbott & Fisher (2015) argue that setting the right team with the right roles and clarified responsibilities is a critical prerequisite when scaling. Sahay & Walsham (2006) focus more on the socio-human aspects of scaling and allude to a variety of dilemmas associated with scaling. Some of them include standardized versus customized solutions, top-down versus bottom-up approaches, and appropriate versus complex technological solutions. Furthermore, they point our attention to considering scaling both concerning the increase in users of the systems and the number of members participating in the implementation team at different stages of the scaling and implementation process. In this regard, they distinguish between two approaches for scaling information systems, the cultivation approach, and the construction approach. The cultivation approach is a more incremental way of seeing scaling which favors a smooth, situated, and improvisational strategy changing smaller parts at a time while aligning those changes with the rest of the system. The cultivation approach stands against the construction approach which offers a more rational and planned approach to scaling emphasizes. It is argued that neither of the two should be prioritized over the other (Sahay & Walsham, 2006).

In classical computer science, scalability has mainly been concerned with algorithm optimization, multithreading, processor optimization, or other techniques that allow scalable performance and execution in a given context, such as a piece of hardware (Ahn et al., 2015; Rajan, 2010; Thierens, 1999; Vachharajani et al., 2005; Yeung, 1999). In other branches, scalability has been studied through the lens of software engineering and architecture, highlighting models, patterns, and processes to make a system scalable and evaluating the potential for scalability (Brataas & Hughes, 2004; Isoyama et al., 2012; Leesatapornwongsa et al., 2017; Mirakhorli et al., 2008; Pahl & Jamshidi, 2016; Rajan, 2010; Srinivas & Janakiram, 2005; Vaquero et al., 2011). The scale cube was introduced to structure at least some of the architectural scale options (Abbott & Fisher, 2015). The scale cube breaks down horizontal scaling into three dimensions. The three dimensions are defined as the x-axis, replication, y-axis, functional splitting, and z-axis, request splitting.

In recent years, a main subtopic of academic interest within scalability has revolved around component-based systems or microservices (D'Antonio et al., 2004; Hasselbring, 2016; Kächele & Hauck, 2013; Lehrig et al., 2015; Márquez et al., 2018) to leverage the elastic horizontal/vertical scaling potential of online modern PAAS or IAAS solutions in the cloud. However, in the field of healthcare services and other critical systems (Knight, 2002), data is considered sensitive and therefore under strict jurisdiction making it difficult to utilize cloud scalability (Heitmeyer, 2005; Walling, 2020).

3 Methodology

The purpose of this study is to examine the successful rapid scaling of sundhed.dk that took place over two weeks during the COVID-19 pandemic as a juxtaposition to other public IS projects (Lauesen, 2020). By examining success stories from the novel context of the COVID-19 pandemic we hope to uncover and retain unheard experiences (Boéri & Giustini, 2023) and tacit knowledge (Schluter et al., 2008) of event-based IS development and scaling processes that evolved throughout the events.

We focus on the period between the day when the Danish prime minister announces the partial re-opening of the Danish society until 14 days, later when the scaled health platform, sundhed.dk should be fully functional and ready to handle the heavily increased user load. This two-week period constitutes the context of our interviews.

Our selection of participants and interview design was inspired by the critical incident technique (CIT) (Flanagan, 1954), to gain entry to, and collect data about the events of the 14 days of rapid scaling (Cenfetelli & Schwarz, 2011) (Gogan et al., 2014). By framing our research interest in a CIT-inspired perspective we aim at identifying the important and relevant events of the rapid scaling process in the unexpected and unusual context of the COVID-19 pandemic.

Our study draws on the technique from a phenomenological perspective and interpretative paradigm (Chell, 1998) rather than the positivist perspective from which it was originally developed. We are interested in subjective nuances that can bolster our understanding of what happened during the scaling of sundhed.dk, hence.

The critical incident interviews were planned and conducted according to Chell's (1998) eight distinguishable aspects of the method (preliminary design work, gaining access, introduction CIT, focusing the theme, controlling the interview, concluding the interview, ethical issues, and analyzing the data. Access was obtained from one of the IT architects involved in the scaling process who pointed out additional respondents inspired by the snowball sampling method. The criteria that we gave to him was that he should choose those who were most intensively involved in the scaling process. We are aware of a potential selection bias regarding the number and criteria for selecting the respondents and we consider also including respondents with a more peripheral role in the scaling process such as managers, developers, and testers in the sample. After interviewing the four respondents, they were pointing out the same critical incidents and thereby we reached the level of saturation (Glaser & Strauss, 1967). The length of each interview was between forty-five minutes and two hours. All the interviews were conducted at the company which supported a natural and relaxed atmosphere for the respondents. A semi-structured interview guide was designed consisting of the following steps. First, the motivation, focus, and aim of the research were presented. The respondent was then asked to present himself including tasks and responsibilities related to the scaling process. To reconstruct the critical incidents in the scaling process a timeline was drawn to let the respondent point out the critical incidents graphically. The interview continued with a detailed focus on each of the identified incidents starting from the launch of the system and going backward from that. During the interview, the respondent was encouraged to describe both the organizational and technical aspects related to the specific incidents. During the interview, we gave attention to potential discrepancies in the respondents' descriptions of the incidents and asked follow-up questions to clarify misunderstandings, thereby increasing the validity of the data. At the end of the interview, the respondent was asked whether we could contact them in case we needed to get some of the discussed topics and incidents clarified. For analyzing the data, we used analytical triangulation where each interview was processed by all four authors. When analyzing the data, we started describing and analyzing each of the identified critical situations inspired by the typical application of the CIT. However, this process revealed several themes across the incidents which were much more interesting to analyze than the individual incidents. Hence, we moved away from an event-based analysis to a more thematic analysis.

4 Preliminary results

The following analysis will examine the prerequisites, diagnostics, and critical decisions made by the task force during the two weeks of intense work that transformed the critical infrastructure of sundhed.dk and enabled the platform to scale accordingly to user demand and regulatory requirements. The analysis will be divided into groups of scalability perspectives such as organizational aspects, virtualization, content delivery network, lazy-loading, and firewall interface configuration.

The initial challenge that was overcome was the formation of a task force itself. Before the rapid scaling requirements, sundhed.dk was limited to a quarterly deployment schedule and processes involving several confirmation steps, stakeholders and budgetary approval hereby presenting overhead for decision-making. Their solution was an appointment of the task force that conducted daily meetings with top-level management to present findings and approve solutions. The task force furthermore was organizationally relocated from development to operations to pre-emptively remedy challenges caused by the principles of separation of duties.

Several challenges were identified by the task force from a technical standpoint. To horizontally scale a platform, the platform should be able to perform and accommodate concurrent users by load-balancing requests between a dynamic set of virtual machines containing all required services. By fulfilling such requirements new hardware was introduced to scale horizontally – and proved to reduce overhead due to service communication located locally on virtual machines as opposed to network communication. Communication overhead between services interfacing was then identified as the result of an infrastructure relying on synchronous network calls and timeouts. Timeout limits were reduced which in turn reduced response time for non-responding services and a large-scale design pattern change in the shape of circuit-breaking was proposed but was not implemented due to the time limitations presented. Consequently, their incoming and outgoing bandwidth was expanded, hardware was added to their prior setup, firewall-interface policy was configured to accommodate more users and an asymmetrical volume was identified between low-volume ingoing requests and high-volume outgoing responses. When identified, the task force implemented a content delivery network that provided additional static insensitive content to clients such that bandwidth from within secured services provided the sensitive data.

5 Conclusion

This research-in-progress paper explored a single unique case. In this case, a select group of Health IT expert practitioners was faced with a scaling problem. The problem progressed under a set of unprecedented circumstances caused by sudden and extensive user request increase during COVID-19. Due to the sensitive nature of health data, it was not an option to scale horizontally and elastically by utilizing cloud services meaning that infrastructure, application, platform, and network had to be configured to allow the service to scale. It was found that the practitioners used a multiplicity of techniques to accommodate the increased number of requests such as reconfiguration of services to allow virtualization and horizontal scaling, lazy-loading, content delivery network of static content, and firewall interface configuration to name a few of which they successfully implemented within a short frame of time.

References

Ahn, J., Hong, S., Yoo, S., Mutlu, O., & Choi, K. (2015). A scalable processing-in-memory accelerator for parallel graph processing. Proceedings of the 42nd Annual International Symposium on Computer Architecture, 105–117. https://doi.org/10.1145/2749469.2750386

Boéri, J., & Giustini, D. (2023). Qualitative research in crisis: A narrative-practice methodology to delve into the discourse and action of the unheard in the COVID-19 pandemic. Qualitative Research.

Brataas, G., & Hughes, P. (2004). Exploring architectural scalability. Proceedings of the 4th International Workshop on Software and Performance, 125–129. https://doi.org/10.1145/974044.974064

Chell, E. (1998). Critical Incident Technique. In G. Symon & C. Cassell (Eds.), Qualitative Methods and Analysis in Organizational Research (pp. 51–72). SAGE Publications.

D'Antonio, S., Esposito, M., Romano, S. P., & Ventre, G. (2004). Assessing the scalability of component-based frameworks. ACM SIGMETRICS Performance Evaluation Review, 32(3), 34–43. https://doi.org/10.1145/1052305.1052311

Hasselbring, W. (2016). Microservices for Scalability. Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering, 133–134. https://doi.org/10.1145/2851553.2858659

Heitmeyer, C. (2005). Developing Safety-Critical Systems: The Role of Formal Methods and Tools. 10th Australian Workshop on Safety Related Programmable Systems.

Isoyama, K., Kobayashi, Y., Sato, T., Kida, K., Yoshida, M., & Tagato, H. (2012). A scalable complex event processing system and evaluations of its performance. Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems, 123–126. https://doi.org/10.1145/2335484.2335498

Kächele, S., & Hauck, F. J. (2013). Component-based scalability for cloud applications. Proceedings of the 3rd International Workshop on Cloud Data and Platforms, 19–24. https://doi.org/10.1145/2460756.2460760

Knight, J. C. (2002). Safety critical systems. Proceedings of the 24th International Conference on Software Engineering - ICSE '02, 547. https://doi.org/10.1145/581339.581406

Leesatapornwongsa, T., Stuardo, C. A., Suminto, R. O., Ke, H., Lukman, J. F., & Gunawi, H. S. (2017). Scalability Bugs. Proceedings of the 16th Workshop on Hot Topics in Operating Systems, 24–29. https://doi.org/10.1145/3102980.3102985

Lehrig, S., Eikerling, H., & Becker, S. (2015). Scalability, Elasticity, and Efficiency in Cloud Computing. Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures, 83–92. https://doi.org/10.1145/2737182.2737185

Márquez, G., Villegas, M. M., & Astudillo, H. (2018). A pattern language for scalable microservices-based systems. Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings, 1–7. https://doi.org/10.1145/3241403.3241429

Mirakhorli, M., Azim Sharifloo, A., & Shams, F. (2008). Architectural challenges of ultra large scale systems. Proceedings of the 2nd International Workshop on Ultra-Large-Scale Software-Intensive Systems, 45–48. https://doi.org/10.1145/1370700.1370713

Pahl, C., & Jamshidi, P. (2016). Microservices: A Systematic Mapping Study. Proceedings of the 6th International Conference on Cloud Computing and Services Science, 137–146. https://doi.org/10.5220/0005785501370146

Rajan, H. (2010). Building scalable software systems in the multicore era. Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, 293–298. https://doi.org/10.1145/1882362.1882423

Schluter, J., Seaton, P., & Chaboyer, W. (2008). Critical incident technique: a user's guide for nurse researchers. Journal of Advanced Nursing, 61(1), 107–114.

Srinivas, A. V., & Janakiram, D. (2005). A model for characterizing the scalability of distributed systems. ACM SIGOPS Operating Systems Review, 39(3), 64–71. https://doi.org/10.1145/1075395.1075401

Thierens, D. (1999). Scalability Problems of Simple Genetic Algorithms. Evolutionary Computation, 7(4), 331–352. https://doi.org/10.1162/evco.1999.7.4.331

Vachharajani, N., Iyer, M., Ashok, C., Vachharajani, M., August, D. I., & Connors, D. (2005). Chip multi-processor scalability for single-threaded applications. ACM SIGARCH Computer Architecture News, 33(4), 44–53. https://doi.org/10.1145/1105734.1105741

Vaquero, L. M., Rodero-Merino, L., & Buyya, R. (2011). Dynamically scaling applications in the cloud. ACM SIGCOMM Computer Communication Review, 41(1), 45–52. https://doi.org/10.1145/1925861.1925869

Walling, S. (2020). A Comprehensive Review on Cloud Computing and Cloud Security Issues. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 483–490. https://doi.org/10.32628/CSEIT206489

Yeung, D. (1999). The scalability of multigrain systems. Proceedings of the 13th International Conference on Supercomputing, 268–277. https://doi.org/10.1145/305138.305203