Izbrana poglavja iz računalništva in informatike

6KT: 45 ur predavanj + 30 ur laboratorijskih vaj
Program: MAG, 2. letnik, spomladanski semester
MAG, 1. letnik, spomladanski semester
UN, 3. letnik, spomladanski semester
Omejitev: 36 študentov

Course title

Fog Computing for Smart Services

(slo. Računalništvo v megli za pametne storitve)

Lectures: Assoc. Prof. Vlado Stankovski
Laboratory work (tentative): Sandi Gec, M.Sc., Uroš Paščinski, Petar Kochovski, M.Sc.
Semester: Spring Semester
Enrollment limit: 36

Objectives and competencies

The course goal is to teach students the overall software engineering process for the development of smart Artificial Intelligence-based fog computing applications. Fog applications usually employ Internet of Things (IoT), Artificial Intelligence (AI), cloud computing and blockchain technologies. This course will focus on the theory and practice when developing complex, component-based, compute, memory and network-intensive multi-tier cloud applications.

The student will gain software engineering competencies at the cross-section of the above mentioned four technology types. Passing this course will result in an improved understanding of the use cases, the functional and non-functional requirements, the current approaches to addressing various dependability, Quality of Service, and trust-related concerns, the major technologies that are commonly used in order to address the whole spectrum of requirements, including orchestration, and the overall integration process.

Prerequisites

General knowledge, experience or interest in the areas of software engineering, DevOps, Internet of Things, Artificial Intelligence, requirements engineering, distributed computing, software integration, ambient intelligence, mobile sensing and cyber-physical systems is necessary for this course. In depth knowledge in any of these areas is very welcome.
This course builds on top of the knowledge gained under several existing courses including:

- (required for all UN or MAG students) Software Development Process (63254) (slo. Postopki razvoja programske opreme):
  - https://fri.uni-lj.si/sl/predmet/63254
  - This course will provide fundamental knowledge about the software engineering process. The present course builds on the concepts, approaches and technologies learnt under the Software Development Process course.
- (asset for UN students) Computer Science Skills (63767) (slo. Tehnične veščine) (DevOps, Kubernetes):
  - https://fri.uni-lj.si/sl/predmet/63767
  - This course helps the students achieve technical versatility in the cloud computing technologies landscape with particular focus on orchestration. It provides necessary basis for understanding the requirements engineering approaches introduced by the present course.
- (asset for MAG students) Cloud Computing (63541) (slo. Računalniške storitve v oblaku):
  - https://fri.uni-lj.si/sl/predmet/63541
  - This course will help the student understand the process of cloud-enabling services and applications. The present course complements this by focusing on

    - the Quality of Service and data management aspects along the software engineering process.
- (complementary to) Deep Learning (63561) (slo. Globoko učenje):
    - https://fri.uni-lj.si/sl/predmet/63561
    - This course will help students understand the relationships between Deep Learning methods and type of compute, memory and network-related Quality of Service requirements in fog computing. The present course builds on this and outlines various optimisation approaches for Deep Learning methods.
- (complementary to) Ambient Intelligence (64m26) (slo. Ambientna inteligenca):
    - https://fri.uni-lj.si/sl/predmet/64m26
    - Knowledge from this subject will complement the understanding of the Digital Twin concept.
- (complementary to) Mobile Sensing (63545c) (slo. Mobilno zaznavanje):
    - https://fri.uni-lj.si/sl/predmet/63545c
    - Knowledge from this course will complement the understanding of the requirements for dynamic information fusion.

Topic justification

The quadrumvirate technologies – Internet of Things (IoT), Artificial Intelligence (AI), cloud and blockchain are currently converging with the aim to facilitate numerous new smart services and applications. They are the substrate of fog computing – an architecture that uses edge devices to carry out a substantial amount of computation, storage, communication locally and thus addresses the requirements of Big Data pipelines, which start at the edge of the computing network and may run across multiple computing tiers up to various public or private Cloud providers.

With Big Data pipelines implemented by means of fog computing it is increasingly possible to realise dependable smart environments, notification systems, Ambient Intelligence solutions and even Cyber-Physical Systems. Fog computing applies to both static (e.g. sensors, cameras) or moving (e.g. sensors, cars) Things. As its application potential is vast, the industrial interest is immense. Application domains include smart cities and communities, smart buildings and homes, smart trading chains, the circular economy, sustainable food production, sustainable tourism, factories of the future, e-health, smart mobility and beyond, and are well exemplified by projects of the Slovenia's Smart Specialisation programme of research and innovation.

This course covers not only theory, but also practice of realising compute, memory and network-intensive smart services and applications through the use of fog computing technologies.

Content

In short terms, fog computing is an umbrella concept that encompasses four different types of technologies: the Internet of Things, Artificial Intelligence, cloud computing and blockchain. It represents the area where these four technologies meet and are made to interact in order to produce many new powerful smart services and applications. In order to deliver dependable and trusted fog applications it is important to focus on improvements of the overall software engineering process in terms of agility, application designs, quality of implementation, operational costs and similar. Most of the involved problems are complex, so it is necessary to reduce the complexity in every step of the way.

Luckily, with the emergence of the DevOps culture and practice, the containerisation of everything, and thus, the ability to use, reuse and repurpose a whole spectrum of AI methods in fog computing applications is already at a touching distance. During this course, at practical level, each student will attempt at understanding the complexities and intricacies involved when developing an elementary smart application.

In the beginning of the course, we shall introduce the main concepts, vision and objectives of fog computing. Particular attention will be paid to the potential use cases, for example, those of the Slovenia’s smart specialisation programme that pose significant requirements for the integration of IoT, AI, cloud and blockchain based services and applications. The existing designs of various smart services and applications will be discussed.

While the students contemplate on these initial set of lectures, their initial laboratory work will focus on trying out a set of technologies required to build a smart application. The work will begin by trying out Machine Learning (ML) models. The students will be reminded about the requirement to submit a project proposal by the end of the third week.

The necessity for dynamic information fusion and some advanced approaches towards the realisation of Digital Twins will be discussed in the third week. The laboratory work will extend the initial ML application with a notification functionality. The lecturer will accept and/or agree the initial project proposals.

The fourth week will focus on the DevOps life-cycle with focus on fog computing applications. During their laboratory work the students will set-up and try out the Fabric8 software engineering tool. We will discuss various approaches to orchestration. In the fifth week, UML-based requirements engineering for concrete compute, memory or network-intensive software utilities will be discussed during the lectures. Edge computing requirements analysis for moving Things (e.g. robots, cars, mobile phones) as opposed to static Things (e.g. homes). The practical work will focus on all the requirements for implementing the ML application and essential UML diagrams will be produced. Further to this, approaches for monitoring applications that may spread across multiple computing tiers will be discussed in the sixth week. Monitoring tools such as Prometheus with Banana visualisation will be used to try out monitoring of the ML application.

When preparing service components for virtualisation, the software engineer must take into account the possibility to optimise the resulting container or Virtual Machine (VM) image for performance, storage and delivery. Various aspects of this process will be discussed in theory and tried out in the laboratory (e.g. ENTICE image optimisation tools). The students will containerise the ML application and the monitoring system. At the end of the seventh week, the students will present the current status of their project, and will pass a mid-semester milestone.

The eighth week will be focused on multi-tier (edge-cloud) application designs with the purpose of addressing the non-functional requirements of the application. During their laboratory work, the students will have the ability to compare their existing designs and consider various improvements. They will proceed with their project work.

The following three weeks (9-11) are devoted to developing and understanding the plethora of AI methods that may be quickly containerised and delivered at the network Edge, the taxonomy of Big Data methods with focus on QoS aspects, and benchmarking methodologies. The laboratory work will focus on achieving QoS monitoring of variants of the ML application.

The final part of the course (12-14) is devoted to blockchain-related concepts, approaches, methodologies and technologies (tokens, Smart Contracts, Smart Oracles) that form part of the fog computing vision. At practical level, the students will attempt at monetising the use of the cloudified ML application by developing and deploying a Solidity based Smart Contract on an Ethereum testnet node.

Finally, we shall conclude by analysing the potential of upcoming new technologies, such as those of the Next Generation Internet initiative of the European Commission. In the laboratory, the students will present the results of their projects to their colleagues and the lecturer.

Learning and teaching methods

Lectures; laboratory work, where the students follow the process of developing a multi-tier smart application; seminar work (project), where the students develop an integrated smart application by following the DevOps practice; consultations; study of literature.

Detailed syllabus

Week 1: Introduction to fog computing. Fog computing and related concepts. Virtualisation and multi-cloud environments. The computing continuum: the Internet of Things, edge, fog and cloud computing. Motivation for fog computing: Big Data pipelines – the four Vs of the Big Data problem. Organisation and control styles. Prospects of decentralisation. Related layered architectures and examples. Variability in fog computing. Energy efficiency and other high-level concerns. Overall DevOps process for delivering fog computing applications. The role of semantics in the integration process. *Laboratory work:* The work will begin by trying out ML models (e.g. with TensorFlow). The students will be reminded about the requirement to submit a project proposal by the end of the third week.

Week 2: Motivation: smart applications and environments. The strategy of smart specialisation in Slovenia – application areas and application examples. Requirements analyses. Multi-tier application designs. System and application adaptation approaches. Implementation and deployment styles. *Laboratory work:* Continued from the first week.

Week 3: Information fusion approaches. Digital twin, augmented virtuality, notification and cyber-physical systems examples.The laboratory work will extend the initial ML application with a notification functionality. The lecturer will accept and/or agree the initial project proposals. *Laboratory work:* Extend the application from week 2 with notification functionality (e.g. email, Slack).

Week 4: The DevOps culture and practice for fog computing. Step-by-step through all DevOps phases. Advanced workbenches for software engineering such as SWITCH for time-critical cloud applications. Approaches to orchestration. *Laboratory work:* the students will set-up and try out the Fabric8 workbench.

Week 5: Requirements engineering for smart services and applications. Functional and Non-Functional Requirements. Quality of Service. High-level requirements. Energy efficiency. Operational costs. Using UML to capture requirements. Trade-offs. *Laboratory work:* UML based requirements engineering for concrete problems at hand including the ML application under development.

Week 6: Software and computing resources and monitoring. Software and computing resources across the edge-to-cloud spectrum. Resource models. Multi-level monitoring approaches. Infrastructure, container, Virtual Machine (VM), Application level metrics. Monitoring systems comparison. QoS modelling approaches. Assurances, ranking and verification of cloud deployment options. *Laboratory work:* Monitoring tools such as Prometheus with Banana visualisation will be deployed to monitoring the chosen ML application.

Week 7: Container and VM images management. Anatomy of a container and VM images. Theory and practice of preparation, fragmentation, optimisation, delivery and images repositories operation. *Laboratory work:* Introduction to Docker compose. Containerise the ML application and the monitoring system. The ENTICE repository of VM images. Perform different operations with containers. At the end of this week, the students will present the current status of their project, and will pass a mid-semester milestone.

Week 8: Multi-tier application designs. Approaches for addressing various non-functional requirements with the design of the smart application. Computing at the Edge. Implementation of Big Data pipelines from the edge to the cloud. Using hybrid cloud providers (IaaS, PaaS, SaaS etc.). *Laboratory work:* the students will have the ability to compare their existing designs and consider various improvements to address the application requirements. They will proceed with their project work.

Week 9: Implementation of AI methods for fog computing. Taxonomy and containerisation of AI methods. Splitting AI methods to be deployed between the edge and the cloud. Optimisation of AI methods for different purposes (e.g. saving energy, speed-up etc.) *Laboratory work:* QoS monitoring of different-purpose variants of the ML application.

Week 10: Big Data pipelines and data management principles in fog computing. IoT platforms and integration. Multi-tier Big Data pipelines. Elasticity and scalability requirements. Data types. Data streams. Moving data to processes versus moving processes to data. Cross-border data management. *Laboratory work:* Trying out an advanced IoT platform (e.g. SensiNact). Integration requirements with the containerised ML application.

Week 11: Big Data systems benchmarking. An overview of Big Data benchmarking methodologies and tools. Qualitative and quantitative evaluation. Benchmark designs in the context of smart applications. *Laboratory work:* Practical work with the Big Data Bench 4.0 benchmark. Designing benchmark for the ML application.

Week 12: Introduction to blockchain. History, design, technology and philosophical aspects of blockchain. Blockchain alignment with fog computing. IoT, AI, cloud and blockchain based business models. Ledger designs. Public and private ledgers. Examples of blockchain-based services and applications. *Laboratory work:* design a monetisation approach for the ML application.

Week 13: Smart Contracts and Smart Oracles. Programming languages, methods, triggers, contracts, decentralised oracles. Service-Level Agreements styles and examples. Using Smart Contracts with multi-tier applications. *Laboratory work:* The students will write a Smart Contracts (e.g. in Solidity).

Week 14: High-level concerns. Achieving dependability, trust, security and privacy in fog computing. Study of various approaches. Using semantics in blockchain (e.g. ont.io). *Laboratory work:* The students will finalise a Smart Contract implementing monetisation for the ML application.

Week 15: Next Generation Internet initiatives. A state-of-the-art analyses of upcoming new technologies and trends. The vision of human-centred Internet. *Laboratory work:* The students will present the outcome of their projects.

Intended learning outcomes

Knowledge and understanding:

- The student understands the expected impact of new smart services and applications.
- The student understands the software engineering process which is essential in order to be able to design, develop and deliver compute, memory and network-intensive smart AI-based applications.
- The student understands the key technology requirements when enabling an ML application for execution in a decentralised multi-tier cloud environment.
- The student understands information fusion approaches and high-level concerns for fog computing applications.
- The student understands monitoring and Quality of Service modelling approaches for smart applications.
- The student will experience the integration of IoT, AI, cloud and blockchain technologies.

Acquired competencies:

- Skills to use and integrate advanced IoT, AI, cloud and blockchain technologies.
- Skills to design and develop a smart application on their own.
- Skills to monitor the applications.
- Skills in Quality of Service modelling for compute, memory and network-intensive applications.
- Skills to monetise an application.

Assessment

Coursework (50 pts total)

- Project proposal (10 pts)
- Mid-semester progress (10 pts)
- Final presentation and report (30 pts)

Final exam (50 pts total)

Biography

Dr. Vlado Stankovski is associate professor of computer science at the Faculty of Computer and Information Science at the University of Ljubljana. He has over 15 years experience in software engineering, grid, cloud, edge and fog computing, distributed systems, semantics, machine learning and data mining technologies. Dr Stankovski has experience in software integration, and has worked with various middleware technologies in the course of the past 15+ years. He has been involved in several National and International projects (see list below). He is involved in the Consortium Supercomputing Centre of Slovenia, and in Slovenia's smart specialization project IQ DOM. Vlado Stankovski currently actively contributes to the software engineering cluster of Horizon 2020 projects, as representative of the ENTICE, SWITCH and DECENTER projects.

Projects

1. 2018–2021, DECENTER: Decentralised technologies for orchestrated Cloud-to-Edge intelligence. European Commission, Horizon 2020, e2.197.700, University of Ljubljana part e305.250. https://www.decenter-project.eu.
2. 2015–2018, ENTICE: dEcentralized repositories for traNsparent and efficienT vIrtual maChine opErations. European Commission, Horizon 2020, e2.767.563, University of Ljubljana part e359.625. http://www.entice-project.eu.
3. 2015–2018, SWITCH: Software Workbench for Interactive, Time Critical and Highly self-adaptive cloud applications. European Commission, Horizon 2020, e2.922.500, University of Ljubljana part e288.000. http://www.switchproject.eu.
4. 2010–2013, mOSAIC Cloud project: Open Source API and Platform for Multiple Clouds. European Commission, Framework Programme VII, e3.705.784, University of Ljubljana part e210.560. http://www.mosaic-project.eu.
5. 2004–2006, DataMiningGrid (technical coordination): Data Mining Tools and Services for Grid Computing Environments. European Commission, Framework Programme VI, e1.883.000, University of Ljubljana part e366.000. Technical Manager.
6. 2004–2007, InteliGrid (coordination): Interoperability of Virtual Organizations on Complex Semantic Grid. European Commission, Framework Programme VI, e2.100.000, University of Ljubljana part e410.000.

References

1. Kochovski, P., Gec, S., Stankovski, V., Bajec, M., Drobintsev, P. D. Trust management in a blockchain based fog computing platform with trustless Smart Oracles. Future generation computer systems, dec. 2019, vol. 101, str. 747-759, doi: 10.1016/j.future.2019.07.030.
2. Taherizadeh, S., Stankovski, V. Dynamic multi-level auto-scaling rules for containerized applications. The Computer Journal, febr. 2019, letn. 62, št. 2, str. 174-197, https://doi.org/10.1093/comjnl/bxy043.
3. Kochovski, P., Drobintsev, P. D., Stankovski, V. Formal Quality of Service assurances, ranking and verification of cloud deployment options with a probabilistic model checking method. Information and Software Technology, maj 2019, letn. 109, str. 14-25, https://doi.org/10.1016/j.infsof.2019.01.003.
4. Kochovski, P., Stankovski, V. Supporting smart construction with dependable edge computing infrastructures and applications. Automation in Construction, 2018, letn. 85, št. jan., str. 182-192, https://doi.org/10.1016/j.autcon.2017.10.008.
5. Paščinski, U., Trnkoczy, J., Stankovski, V., Cigale, M., Gec, S. QoS-aware orchestration of network intensive software utilities within Software Defined Data Centres - An

architecture and implementation of a Global Cluster Manager. Journal of Grid Computing, 2018, letn. 16, št. 1, str. 85-112, https://doi.org/10.1007/s10723-017-9415-1.

6. Taherizadeh, S., Jones, A.C., Taylor, I., Zhao, Z., Stankovski, V. Monitoring self- adaptive applications within edge computing frameworks: A State-of-the-Art review. The Journal of Systems and Software, feb. 2018, letn. 136, 20 str, https://doi.org/10.1016/j.jss.2017.10.033.
7. Taherizadeh, S., Stankovski, V., Grobelnik, M. A capillary computing architecture for dynamic Internet of Things - Orchestration of microservices from Edge devices to Fog and Cloud providers: 2938. Sensors, 2018, letn. 18, št. 9, str. 1-23, https://doi.org/10.3390/s18092938.
8. Gec, S., Kimovski, D., Paščinski, U., Prodan, R., Stankovski, V. Semantic approach for multi-objective optimisation of the ENTICE distributed Virtual Machine and container images repository. Concurrency and Computation: Practice and Experience, nov. 2017, str. 1-19, https://doi.org/10.1002/cpe.4264.
9. Kimovski, D., Marosi, A. Gec, S., Saurabh, N., Kertesz, A., Kecskemeti, G., Stankovski, V., Prodan, R. Distributed environment for efficient Virtual Machine image management in federated Cloud architectures. Concurrency and Computation: Practice and Experience, nov. 2017, letn. 30, št. 20, str. 1-16, https://doi.org/10.1002/cpe.4220.
10. Casale, G., Chesta, C., Deussen, P. et al. Current and future challenges of software engineering for services and applications. Procedia Computer Science, 2016, letn. 97, str. 34-42, https://doi.org/10.1016/j.procs.2016.08.278.
11. Zhao, Z., Martin, P., Wang, J. et al. Developing and operating time-critical applications in clouds: The State of the Art and the SWITCH approach. Procedia Computer Science, vol. 68, 2015, str. 17-28, https://doi.org/10.1016/j.procs.2015.09.220.
12. Stankovski, V., Petcu, D. Developing a Model Driven Approach for engineering applications based on mOSAIC: Towards sharing elastic components in the Cloud. Cluster Computing, 2013, letn. 17, št. 1, str. 101-110, https://doi.org/10.1007/s10586-013-0263-x.
13. Južna, J., Češarek, P., Petcu, D., Stankovski, V. Solving solid and fluid mechanics problems in the Cloud with mOSAIC. Computing in Science & Engineering, 2014, letn. 16, št. 3, str. 68-77, DOI:10.1109/MCSE.2013.135.
14. Multiple authors (including Stankovski, V.) Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges, https://doi.org/10.1016/j.iot.2019.100118, in press.
15. Gec, S., Lavbič, D., Bajec, M., Stankovski, V. Smart contracts for container based video conferencing services: architecture and implementation. V: Coppola, M. (ur.). Economics of grids, clouds, systems, and services : proceedings, (Lecture notes in computer science (Internet), ISSN 1611-3349, Computer communication networks and telecommunications, 11113). Cham: Springer. cop. 2019, str. 219-233, ilustr. https://link.springer.com/chapter/10.1007/978-3-030-13342-9_19, doi: 10.1007/978-3-030-13342-9_19.
16. Kochovski, P., Bajec, M., Sakellariou, R., Stankovski, V. A smart and safe construction application design for fog computing. V: Chang, Carl K. (ur.). IEEE services 2019: proceedings. Los Alamitos (CA); Washington; Tokyo: IEEE. cop. 2019, str. 376-377, ilustr., doi: 10.1109/SERVICES.2019.00112.
17. Kavakli, E., Sakellariou, R., Stankovski, V. Towards a Methodology for Evaluating Big Data Platforms. V: Chang, Carl K. (ur.). IEEE services 2019: proceedings. Los Alamitos (CA); Washington; Tokyo: IEEE. cop. 2019, doi: 10.1109/SERVICES.2019.00113.
18. Kochovski, P., Sakellariou, R., Bajec, M., Drobintsev, P.D., Stankovski, V.An architecture and stochastic method for database container placement in the edge-fog-cloud continuum. V: IPDPS 2019 : proceedings, (Proceedings - IEEE International Parallel and Distributed Processing Symposium, ISSN 1530-2075). Los Alamitos (California); Washington; Tokyo: CPS: IEEE Computer Society. cop. 2019, str. 396-405, ilustr. doi: 10.1109/IPDPS.2019.00050.
19. Štefanič, P., Kimovski, D., Suciu, G., Stankovski, V. Non-functional requirements optimisation for multi-tier cloud applications: An early warning system case study. V: 2017 IEEE SmartWorld : Ubiquitous Intelligence & Computing, Advanced & Trusted Computed,

Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI): 2017 conference proceedings: San Francisco Bay Area, California, USA, August 4-8, 2017. Piscataway: IEEE, 2017, str. 1-8, ilustr., doi: 10.1109/UIC-ATC.2017.8397637.

20. Taherizadeh, S., Taylor, I.J., Jones, A.C., Zhao, Z., Stankovski, V. A network edge monitoring approach for real-time data streaming applications. V: BAÑARES, José Ángel (ur.), TSERPES, Konstantinos (ur.), ALTMANN, Jörn (ur.). Economics of grids, clouds, systems, and services: revised selected papers, (Lecture notes in computer science, 10382). Cham: Springer. 2017, str. 293-303, ilustr., doi: 10.1007/978-3-319-61920-0_21.