University of Glasgow | School of Computing Science

Assessed Coursework

Course Name	Distributed Algorithms and Systems – DAS(H)			
Coursework Number	COMPSCI 4019			
Deadline	Time:	16.30	Date:	01 December 2017
% Contribution to final course mark	20 %			
Solo or Group	Solo		Group	✓
Anticipated Hours	20 hours per person			
Submission Instructions	See 'What to submit' on page 4			
Please Note: This Coursework cannot be Re-Done				

You must complete an "Own Work" form via https://webapps.dcs.gla.ac.uk/ETHICS for all coursework UNLESS submitted via Moodle

Distributed Algorithms and Systems – DAS(H)
Group Assessed Exercise: 2017-2018

Date issued	*Friday, 20th October 2017*
Demonstration	*Thursday, 30th November 2017 (Glasgow)* *Tuesday, 28th November 2017 (Singapore)*
Final submission deadline	*16:30, Friday, 1st December 2017*

This is the main exercise for the DAS(H) module; it is worth 20% of the assessment for the course (the remaining 80% is for the May examination).

Description

You are asked to *design* and *implement* a novel distributed system. You will then write a report in the form of a research *article* documenting your design and implementation. The implemented system should demonstrate some *original contribution* that you will need to argue in your report by comparing and contrasting it to related research and/or industrial work.

You are free to choose what will be the focus of your contribution. It can be a distributed application that introduces some novel functionality such as, for example, a distributed measurement system to give a network-wide view of performance; a distributed data collection and analytics framework; a peer-to-peer network engaging in distributed computation; an airline reservation system; etc.; or it can be a novel implementation of a distributed algorithm to enable, for example, coordination and agreement in the presence of extraordinary network events or failures.

You will undertake this assignment *in groups* to accommodate for the extra load of research in deriving your own specification and design for the system you will be implementing. Groups will be formed randomly from the class roster, and group membership will be communicated to you via email and/or will be published on Moodle[1]. It is the group members' responsibility to divide the work and manage the overall project.

The following list contains a few *examples* of what can constitute an original contribution but it is by no means exhaustive:

- A distributed application that offers some novel functionality (e.g., implements a new idea, or some existing functionality in a new way)
- A smart distributed sensor-actuator application that offers intelligent automation and data processing (e.g., implement a new approach to collect data and perform distributed data processing)
- Porting an existing distributed system to a new RMI environment (Java RMI, in this case)
- Adding distribution to some centralised/monolithic system
- Adding redundancy to an existing distributed computation to ensure, e.g., correctness through voting or better performance
- A more efficient implementation of a known distributed system or algorithm
- An implementation of a novel distributed algorithm for communication or coordination (e.g., leader election, mutual exclusion, multicast, etc.)

[1] http://moodle2.gla.ac.uk/course/view.php?id=964

Continued Overleaf/

- An implementation of a distributed algorithm that tackles an unsolved problem/works under certain conditions where other (equivalent) algorithms fail (e.g., by converting certain types of failures to more acceptable types or avoiding them altogether)

You will have to argue about the originality of your contribution in your article.

Example articles have been uploaded on Moodle under the "Papers" section. Furthermore, the following events and outlets can give you an idea of research contributions in distributed systems and related areas. Some of the systems/approaches described therein may seem overly complex (your coursework can be something much simpler!) but they should still give you good inspiration for the kinds of systems you can focus on as well as an idea of how to shape your article. You should definitely do some research at these places (and others you can find online) before initiating discussions within your group about the system/functionality you will be developing.

ACM/IFIP/USENIX International Middleware Conference:

http://www.middleware-conference.org/

IEEE International Parallel & Distributed Processing Symposium:

http://www.ipdps.org/

IEEE Pervasive Computing and Communications Conference:

http://www.percom.org/

IEEE Transactions on Parallel and Distributed Systems (TPDS):

http://www.computer.org/portal/web/tpds/

Elsevier Journal of Parallel and Distributed Computing:

http://www.elsevier.com/wps/find/journaldescription.cws_home/622895/description

You should be able to access papers from these outlets via the University's electronic subscriptions (e.g., to IEEEXplore and the ACM Digital Library), and/or possibly on authors' websites.

Your code should be written in Java, using RMI. Marks will be awarded based on the elegance of your implementation, the level of challenge involved in realising it, and features included such as e.g., handling concurrency (interference, memory consistency etc.), server callbacks, concurrent tasks for e.g., housekeeping. You should not assume particular IDEs/toolsets. – You will be asked to demonstrate your code in the lab, hence you must make sure it works on the School/lab Linux machines.

Your article should be around 3,000 words and no more than 6 pages long following either the IEEE or the ACM proceedings format. MS Word and LaTeX templates can be found at the following locations (feel free to increase font to 11pt.), respectively:

http://www.ieee.org/conferences_events/conferences/publishing/templates.html

http://www.acm.org/sigs/publications/proceedings-templates

The article should contain at least the following sections (title, additional sections, sub-sections, numbering and actual heading names are left at your discretion):

Abstract: A summary of your contribution; what problem does your system solve; what new functionality it enables; what you have achieved by building it; no more than 200 words.

Introduction (and motivation): Put your work into a wider context; give a brief history of the area and the state of the art; elaborate on your choice for building the particular system; justify why this system is important and why it is needed (expand on the main points sententiously covered in the abstract and explain what the work documented in the article is all about).

Continued Overleaf/

System Design: Describe your design choices and give a high-level overview of your system and its functionality; include figures and/or diagrams to explain concepts, interactions between components etc.; justify your design choices and discuss any resulting implications

Implementation: Describe the details of your implementation; how you have realised your design; what aspects of it worked as expected and whether there has been any missing functionality; how you have dealt with important issues such as concurrency, server-to-client communication, etc.

Evaluation: Provide evidence that your system works; include quantitative indicators that verify any claims you have made; provide proof of the performance of your system; use graphs to visualise your main results.

Related Work: *Critically* describe related work in the field you chose to focus on and highlight the most important contributions made by others; compare and contrast existing work with the system you have implemented and therefore highlight your main achievements; say what existing systems do not do that your system does.

Conclusions (and future work): Provide a brief description of the overall system you have documented and highlight your main contributions. Include a main take-away message and possibly identify future dimensions for your work.

References: Provide a numbered list of related works you have identified and discussed; it should include articles and possibly on-line resources; this is not a bibliography, only include works that you have cited from within your article.

Assessment

Implementation (40%): Marks will be awarded for working systems that implement the advertised functionality. Level of marks will depend on the novelty/contribution of the system and the level of challenge involved in realising it. Inclusion of advanced features such as concurrency control, server callbacks, etc. will also be taken into consideration. You are NOT expected to provide any sophisticated (G)UI for your system. Your solution should mainly focus on aspects of the system that make it function in a distributed fashion.

Demonstration (20%): Each group will be required to give a 10-minute demo of their implementation, detailing the functionalities and the design justification of the distributed application they have developed. Demos are scheduled during normal DAS(H) contact hours on **Thursday, 30th Nov. 2017 (Glasgow), Tuesday, 28th Nov, 2017 (Singapore)**.

Article/Report (40%): Marks will be awarded for coherent articles that will satisfactorily address the main points of the relevant sections outlined above. Particular attention should be given to the construction of the argument regarding the originality of the implemented system and its comparison to related work, as well as to the quantitative evaluation of the system.

What to submit

Coursework will be submitted *electronically* via Moodle **no later than 16.30 on Friday, 1st December 2017**. Submission links will be provided on Moodle DAS(H) page for the following items:

- Softcopy version of your system – all of the *source code* for your system (Java files, launch scripts, test data, etc.) should be produced as a single .tar file, called GROUPNN.tar where NN is your group number. The archive should also contain a text file describing how your system should be tested starting from extracting the source code from the submitted .tar file, etc.
- Softcopy version of your *article* – a single .pdf file called GROUPNN.pdf where NN is your group number.

One submission for each of the aforementioned *four files* should be made per group by one of the members. I will subsequently acknowledge submissions to all relevant group members sometime after the submission deadline.