The University of Western Ontario
London, Canada

Department of Computer Science

cs 9864b

Software Engineering for Big Data Applications and Analytics

Course Outline – Winter 2018

Logistics and Instruction:

Class Venue	MC 320
Day and Hours	Fri 10.30 am – 1.30 pm
Instructor	Nazim H. Madhavji (last-name <<aatt>> geee-may-l ☺)
Office Hours:	Right after the class: 1.30 pm – 2.30 pm.

Sessional Dates

Term begin	Mon 8th January, 2018.
First class	Fri 12th January, 2018.
Last class	Fri 6th April, 2018.
Term end	Wed 11th April, 2018.
Reading week	Mon 19th – Fri 23th February, 2018.

Important Announcements

NEW/RECENT OLDER	Date	Description
☛		○ **For all the course resources, please click here (restricted access): click here.**
☛	8-Jan-2018	Please bring your laptops in the class, we will need them!

1. Introduction

The focus in this course is on the *development*, *maintenance* and *evolution* of applications dealing with large volumes of data (called "Big Data"). Data has generally been everywhere! This is therefore not new. With recent advances in technologies (e.g., ubiquitous computing, internet of things, cloud computing, etc.), however, it has become more practical to capture and process large volumes of both structured and unstructured data (e.g., patient records, traffic data, video data, images, sporting statistics, events, logistics data, on and on). Also, new kind of data, not previously existing, has become available with the advent of mass use of internet and communication technologies (e.g., online access, e-commerce, social media data, mobile data, and others). There is thus considerable and growing interest amongst organisations and institutions to analyse such data for their purposes. We are only at the beginning of this paradigm shift.

In this course, we shall focus on such topics as:

- models of lifecycle processes in the context of Big Data environments;

- technical processes for the development, maintenance, and evolution of Big Data applications;
- underlying technologies for operational support for Big Data applications;
- scalable data analytics; and
- business models centered on Big Data.

This is an emerging area in the field of software engineering.

Big Data refers to data sets on a massive scale, usually produced by, or obtained from, different sources. Initially, such data was characterised by the attributes: *volume* (amount of data), *variety* (different types of data), and *velocity* (speed with which data arrives and is processed). Subsequently, other data characteristics have emerged: *variability* (inconsistency in the data set); *veracity* (accuracy of the data upon which analysis depends greatly); *complexity* (inherent complexity involved in linking, connecting, and correlating data items from different sources, so that meaningful information can be inferred and conveyed to the stakeholders); *validity* (relevance for intended use); *volatility* (retention aspects, including change and length of life issues), and *value* (to the stakeholders). The first three characteristics are popularly referred to as "the 3 V's of Big Data" and there are debates about inclusion/exclusion of other Vs in the core set of attributes.

Traditional database management and processing systems do not have the capacity or processing power to deal with Big Data. Thus, this called for creation of novel algorithms and system architectures to store, curate, manage and process Big Data. The field of Big Data provides new opportunities for the analysis of such data and for discovering interesting trends and unknown or unforeseen relations among the data items. This technical domain is referred to as Data Analytics.

Researchers and practitioners in the software engineering and technologies community have recognised that there is a need to create novel architectures and frameworks to support the storage, curation, management and processing of big data sets. Thus, distributed reference architectures and programming paradigms have been developed recently, giving rise to concepts such as the *Map-Reduce* processing model. Examples of frameworks supporting this processing model are Hadoop and MongoDB. In addition, Big Data encompasses unstructured data, the modelling, searching and processing of which requires novel techniques such as Latent Semantic Indexing (LSI) and its variants.

In parallel, the past few years have seen a paradigm shift in enterprise computing, moving from static, in-house, systems to large clusters of private, public, or hybrid systems (referred to as *clouds*). Furthermore, resources (e.g., infrastructures, platforms, and software applications) are provisioned as *services* (over virtual platforms) that are referred to, respectively, with the acronyms: IaaS, PaaS, and SaaS. Such services are provisioned under the paradigm of "service oriented" systems embodying service oriented architectures (SOA).

Given that "data" is at the centre of systems, it is inevitable that the "Data as a service" (DaaS) would be added as a model for conducting business. Examples services include: continuous data security for clients, provision of data, facilitating data sharing among collaborating partners, etc. With SOA, data may reside on any of the platforms, and services can be provided on demand to geographically remote clients.

This type of deployment gives rise to new architectures referred to as System-of-Systems (SoS) or Ultra-Large-Scale (ULS) systems. These systems produce massive amounts of transactions and internal data for monitoring purposes. The analysis of such data is critical for maintenance (perfective, corrective, and adaptive) and verification of such systems. For instance, Big Data Solution (BDS) components talk to each other and their subcomponents distributed across a cluster of computers. In this arrangement, for example, database failure to access data might be caused not by a defect in the database but by corruption in the underlying distributed storage. Detailed logs generated by the Big Data system can reach tens of terabytes. Data gathered from multiple systems require petascale storage. Manually analysing such data is not practical, thus calling for innovative techniques for supporting system recovery and maintenance.

To date, little has been accomplished as to how to use big data analytics to support SoS and ULS maintenance and evolution. Issues related to the collection, modelling, storage and processing of massive amounts of logged data for maintenance, evolution, compliance and verification purposes is a subject of emerging research.

Big Data also needs to be represented and denoted by formalisms that facilitate efficient storage and processing. Conceptual modelling as well as knowledge management research has produced a number of frameworks that permit structural representation and semantic interpretation of large data sets. Also, the software engineering community has created novel meta-languages so that large data sets can be efficiently modelled. Example meta-languages include: the Meta-Object Facility (MOF), the Resource Description Framework (RDF), and the Web Ontology Language (OWL) that give rise to efficient data representation models such as Linked Data. Tools have emerged to support such modelling activities. Examples of tool frameworks for defining domain models and schemas include: the Eclipse Modelling Framework (EMF) and the XML MetaData Interchange (XMI).

While progress is being made in the technological areas of Big Data and Analytics, there is little movement in the area of disciplined development of "applications" for processing and generating Big Data. For example, little is known about lifecycle processes for: engineering requirements and architectures, and for testing applications with particular focus on Big Data.

2. Style of Course

The theoretical aspect of this course relies almost exclusively on published papers and third-party reports. Students will be expected to have read scheduled material prior to attending classes where sessions will be driven primarily by discussions and questions and answers (both assessed throughout the term). Guest speakers will be invited as appropriate. In groups, students will be

conducting an in-depth search of relevant literature, conducting critical analysis of this, and presenting their findings. The practical aspect in this course will involve an application development class project focused on Big Data.

3. Learning outcomes

The following learning outcomes are anticipated:

- Domain understanding of Big Data and Data Analytics
- Literature review and analysis skills
- Presentation skills and defence
- Identification of research gaps
- Reading and comprehension of literature on Big Data and Data Analytics
- Understanding of engineering, maintenance and evolution of Big Data applications software

4. Course Evaluation

- All material covered in the course (including lectures, discussions, assignments and projects, books and other cited resources) is examinable.
- The teaching staff reserve the right to adjust (lower or raise) a student's marks for the tabulated components below based on their judgment of the student's knowledge and understanding of the subject matter during the term.
- Project logistics:
    - Projects will be carried out in groups.
    - The membership of a group will be assigned based on the provided descriptions of an individual's background, skills and experience. Any adjustments in team membership to be made will be done only at the beginning of the course. Once formed, the group membership will not be changeable for the rest of the term.
    - Rules for group behaviour, responsibilities, constraints, consequences, etc., will be presented in the class by the instructor.
    - In group work, each member is expected to contribute equitably. There will be peer reviews which will be considered in moderating an individual's mark.
    - **EXTREMELY IMPORTANT**: In the event a group member is removed from his or her group for reasons of discontentment, please note that placement of that individual in another group will not be possible. In this case, the student concerned would have no choice but to withdraw from the course. PLEASE

note that there is nothing else that can be done within the parameters of operation of this course.

- The grading criteria, as applied to each evaluation component, will be described with the details of the component.
- **Attendance in class is mandatory**. See the table below for consequences of absenteeism.
- Those who miss the quiz, test, in-class question and answer session, etc., will receive zero marks for this component (exceptions only as per the university policy).
- Late submissions of assignments and projects will not be accepted, so please be forewarned to commence tasks upon assignment.
- If for any reason any evaluation component cannot be adhered to by the instructor, the rest of the marks will be prorated.
- *There will be no makeup Quiz or Test, except for students requesting a Special Quiz or Test for religious reasons. These students must have notified the course instructor, by email, at least 2 weeks prior to the Quiz or Test.*

  If you miss the Quiz or Test for any other reason, follow the procedure for Academic Accommodation for Medical Illness. If accommodation is approved by your Dean's office, the Quiz or Test component will be redistributed to the other evaluation components of the course.

- IMPORTANT: grading will begin on day one. There will be no make up mark for days missed.

Description	**% marks**	**Deadline**
• *Individual work*: Weekly readings of assigned literature. • summary of readings (weekly deliverables throughout the term)	20%	weekly
• *Individual work*: In-class discussions (Q&A)	15%	weekly

• *Group work*: Topic presentation:
  ▪ literature search on an approved topic
  • Approx. 5 (or more?) substantive papers expected
  ▪ creating an analytic spreadsheet
  ▪ creating a powerpoint presentation
  ▪ presentation
	20%	To be scheduled

• *Group work*: Software development project
  ▪ Preliminary: Core idea and core requirements
  ▪ Intermediate-1: Refined requirements;
	45%	To be scheduled

Preliminary Architecture ▪ Intermediate-2: Refined architecture; preliminary demo. ▪ Intermediate-3: Near final demo. ▪ Final: Delivery of documentation and system.		
• Attendance: mandatory	Minus 5 % per class missed	

5. Prerequisite

- Registration in a graduate program.
- Undergraduate level course on Software Engineering (instructor's discretion for equivalent background and experience).

6. Course Material

Selected weekly readings for in-class discussion.

7. Other

Assignment Backups:

- It is your responsibility to keep up-to-date backups of assignment files in case of system crashes or inadvertently erased files. Keep electronic copies of all material handed in, as well as the actual graded assignment, to guard against the possibility of lost assignments or errors in recording marks.

Email Contact

We will occasionally need to send email messages to the whole class, or to students individually. Email will be sent to the UWO email address assigned to students by Information Technology Services (ITS), i.e. your email address @uwo.ca. It is each student's responsibility to read this email on a frequent and regular basis, or to have it forwarded to an alternative email address if preferred. See the ITS website for directions on forwarding email.

However, note that email at ITS (your UWO account) and other email providers such as hotmail.com or yahoo.com establish quotas or limits on the amount of space available to you. If you let your email accumulate there, your mailbox may fill up and you may

lose important email from your instructors. Losing email is not an acceptable excuse for not knowing about the information that was sent.

Academic Accommodation for Medical Illness

- for work representing 10% or more of the overall grade in the course:

If you are unable to meet a course requirement due to illness or other serious circumstances, you must provide valid medical or other supporting documentation to your Dean's office as soon as possible and contact your instructor immediately. It is the student's responsibility to make alternative arrangements with their instructor once the accommodation has been approved and the instructor has been informed. In the event of a missed final exam, a "Recommendation of Special Examination" form must be obtained from the Dean's Office immediately. For further information please see:

http://www.uwo.ca/sci/counselling/pdf/Submission-of-Medical-Documentation-for-Course-Appeal.pdf

A student requiring academic accommodation due to illness should use the Student Medical Certificate when visiting an off-campus medical facility or request a Record's Release Form (located in the Dean's Office) for visits to Student Health Services. The form can be found here:

http://www.uwo.ca/univsec/pdf/academic_policies/appeals/medicalform.pdf

Students who are in emotional/mental distress should refer to

Mental Health@Western

for a complete list of options about how to obtain help.

- for work representing less than 10% of the overall grade in the course:

***There are no such components in this course**.*

Links to the policies on Accommodation:

Link to policy on Accommodation for Illness

www.uwo.ca/univsec/pdf/academic_policies/appeals/accommodation_illness.pdf

(which includes a link to the Student Medical Certificate)

Link to the policy on Accommodation for Students with Disabilities

www.uwo.ca/univsec/pdf/academic_policies/appeals/accommodation_disabilities.pdf

Link to the policy on Accommodation for Religious Holidays

www.uwo.ca/univsec/pdf/academic_policies/appeals/accommodation_religious.pdf

Link to the website for Registrarial Services:

- http://www.registrar.uwo.ca

Link to services provided by the University Students' Council:

- http://westernusc.ca/services/

Accessibility Statement

You may wish to contact Services for Students with Disabilities (SSD) at 661-2111 x 82147 for any specific question regarding an accommodation.

Ethical Conduct

Scholastic offences are taken seriously and students are directed to read the appropriate policy, specifically, the definition of what constitutes a Scholastic Offence, at the following Web site:

http:// www.uwo.ca/univsec/pdf/academic_policies/appeals/scholastic_discipline_undergrad.pdf

Plagiarism: *Students must write their essays and assignments in their own words. Whenever students take an idea, or a passage from another author, they must acknowledge their debt both by using quotation marks where appropriate and by proper referencing such as footnotes or citations. Plagiarism is a major academic offence.*

You may discuss approaches to problems among yourselves; however, the actual details of the work (coding, answers to concept questions, etc.) must be an individual effort.

The standard departmental penalty for assignments that are judged to be the result of academic dishonesty is, for the student's first offence, a mark of zero for the assignment, with an additional penalty equal to the weight of the assignment also being applied. You are responsible for reading and respecting the Computer Science Department's policy on Scholastic Offences and Rules of Ethical Conduct .

The University of Western Ontario uses software for plagiarism checking. Students may be required to submit their written work and programs in electronic form for plagiarism checking.