Vorlesung Big Data

Seminarthemen

Bitte nennen Sie drei Themen in absteigender Wunschreihenfolge, beispielsweise: „5, 2, 7“. Wenn Thema Nr. 5 bereits vergeben ist, wird Ihnen dann Thema Nr. 2 zugeteilt und dann, sollte Thema Nr. 2 ebenfalls vergeben sein, Thema Nr. 7.

Alle Seminarthemen sollen auf 10 Seiten (Arial o.ä., 12pt, 2cm Seitenrand, einfacher Zeilenabstand) ausgearbeitet werden. Es folgt zum Prüfungstermin ein kurzer Vortrag von 15 Minuten Dauer. Die Seminararbeiten müssen spätestens zum Vortragstermin per E-Mail an eric.veith@offis.de eingegangen sein.

1. **Wertschöpfungspotenziale von Big-Data-Anwendungen**
   Wo wird mit Big-Data-Technologien Geld verdient?
2. **Das *Google File System***
   Erläutern Sie die grundsätzlichen Entwurfsentscheidungen des GFS: Wie erreicht es seine Ausfallsicherheit? Wie funktioniert die Replikation? Wie erreicht es seine Geschwindigkeit?
3. **Das Hadoop-Ökosystem**
   Welche Softwareprojekte sind rund um Hadoop entstanden? Wo finden sie ihre Anwendung?
4. **Big Data & Deep Learning**
   Sind Big-Data-Stacks notwendig für Deep-Learning-Anwendungen? Wo liegt der Nutzen? Wie können konkrete Integrationen der beiden Konzepte aussehen, z.B. eine Verknüpfung von Hadoop und PyTorch?
5. **Big Data in der Cloud**
   Mit welchen Technologien lassen sich Big-Data-Systeme effizient in einer Cloud-Umgebung skalieren? Wie lässt sich mit Container-Orchestrierung Kosten sparen?
6. **Maschinelles Lernen mit Spark**
   Welche Möglichkeiten für maschinelles Lernen bietet Apache Spark selbst? Wie kann die Software zu existierenden ML-Anwendungen „zuliefern“?
7. **Dokumentenorientierte Datenbanken**
   Was steckt hinter dem Schlagwort NoSQL? Wie funktionieren dokumentenorientierte Datenbanken? Sind sie ausfallsicher?

8. **Potenzial und Grenzen klassischer DBMS**
   SQL-Datenbanksysteme (Database Management Systems, DBMS) galten lange als die einzig sinnvolle Möglichkeit, große Datenmengen sinnvoll zu speichern und abfragbar zu machen. Gilt das noch für Big-Data-Anwendungen? Wo liegen die (technischen) Grenzen traditioneller DBMS und wohin geht deren Entwicklung?

9. **Klassische Hochverfügbarkeits-Cluster**
   Das Google File System und Hadoop sind für den Betrieb auf Verbünden alltäglicher Geräte ausgelegt. Was macht im Gegensatz dazu klassische, hochverfügbare Systeme aus? Wo liegen deren Vor- und, im Bezug auf Big Data, Nachteile?

10. **MQTT, ØMQ, RabbitMQ & Co**
    Wie passen Messaging-Bus-Systeme und IoT-Protokolle in einen Big-Data-Stack?

11. **Apache Cassandra**
    Wie funktioniert Apache Cassandra technisch? Was sind die Vor- und Nachteile der Datenbanklösung? Für welches Einsatzgebiet ist sie gedacht? Worin besteht der Unterschied zwischen Cassandra und HBASE?

12. **Datenmigration zu Big-Data-Systemen**
    Wie migriert man von bestehenden Systemen wie klassichen RDBMS zu HBASE oder anderen? Was muss auf dem Migrationspfad beachtet werden?

13. **Visualisierung von großen Datensätzen**
    Vertiefen Sie: Welche neuen Darstellungsmöglichkeiten sind im Zuge des Big-Data-Paradigmas aufgekommen? Welchen Wert besitzen sie? Sind sie spezifisch für Big Data?

14. **Container & Big Data**
    Welchen Nutzen haben moderne Container-Virtualisierungslösungen im Big-Data-Kontext? Lassen sich Kubernetes & Co sinnvoll einsetzen?

15. **Polyglot Persistence**
    Gibt es in Zeiten von Big Data eine einzig sinnvolle Datenbanklösung, und wenn ja, wer ist der führende Anbieter? Oder lohnen sich stattdessen mehrere Systeme im Parallelbetrieb – und falls ja, was sind die wesentlichen Softwarepakete und ihre Einsatzgebiete?

16. Data Governance

„Big Data“ und die EU-Datenschutzgrundverordnung: Unvereinbare Konzepte? Was muss in der Systemarchitektur und bei der Administration berücksichtigt werden, um der DSGVO zu entsprechen? Wie schlagen sich die juristischen Anforderungen im Code nieder?

17. Wissensmanagement mit Big Data

Welche Ansätze existieren, um, basierend auf Big-Data-Lösungen, Systeme zum Wissensmanagement zu etablieren? Gibt es Ansätze, automatisiert Informationen und Wissen aus einem Big-Data-System zu extrahieren?

18. Funktionale Programmierung

Big-Data-Stacks etablieren häufig ein funktionales Paradigma bei der Programmierung. Was unterscheidet funktionale Programmierung von imperativer oder objektorientierter? Was sind die Vor- und Nachteile? Welche Sprachen eigenen sich besonders gut?

19. Definieren Sie Ihr eigenes Thema