Distributed collection, processing and analysis of sensor data

Dr. Vladimir Korkhov
v.korkhov@spbu.ru

Department of Computer Modelling and Multiprocessor Systems
Faculty of Applied Mathematics and Control Processes,
Saint Petersburg State University
Motivation

• Sensor data generated in the field might require immediate preprocessing: *need for cheap light-weight infrastructure* e.g. based on SoC (system-on-chip) devices

• Handling groups of sensors and data processing on unreliable and low-performance SoC devices requires *workload management software*: *light-weight fault-tolerant scheduler for distributed data processing*

• Detailed sensor data on object state, conditions and behaviour can be used to detect anomalies and predict characteristics: *analysis of telemetry data* e.g. sea vessel motion, conditions and traffic data

• Building models of object usage scenarios based on historical sensor data to *control devices*: e.g. *smart lighting control based on luminance and motion sensors and prediction models*
Outline

• Scenarios of sensor data collection, processing, analysis
  • Using unreliable resource-constrained hardware for data collection and processing
  • Detection of anomalies in sea vessel movement using telemetry data
  • Using IoT-related technologies for smart lighting, water consumption control
Data collection and processing

- **Scenario 1:**
  Using unreliable resource-constrained hardware for data collection and processing

- **Workload manager** (scheduler) for orchestrating distributed data processing and general purpose near field computations on dynamic (or unreliable) infrastructure based on resource-constrained hardware (e.g. Raspberry Pi, Intel Edison or similar SoC systems)
Workload scheduling in distributed infrastructures

- Create arbitrarily nested **micro-kernels** for computational workload which modify and **restore their state** in case of a failure.
- Fault-tolerance for distributed processing: workers and **master node fault-tolerance**
- The more scalable an application is the more resilient to node failures it becomes.
- No matter a *master* or a *slave* node fails, the resulting performance roughly equals to the one without failures with the number of nodes minus one.
- When a *backup* node fails the model works as low-overhead checkpoint & restart.

**Evaluation results**
Dynamic load balancing on unreliable and resource-constrained hardware

**Goal:** orchestrate distributed computing and data processing on microcomputers with help of *Apache Spark* and custom scheduler focused on fault tolerance and dynamic rescheduling

**Hardware:** Raspberry Pi / Intel Edison / Orange Pi

**Middleware:** Apache Spark; custom scheduler

**Sample problem:** FFT (data from sensors)
Fault-tolerant scheduler

Fault-tolerant scheduler for distributed applications:

- **Fault-tolerance** support including **master node** faults
- Optimized for working on **unreliable and resource-constrained** SoC devices
- Running in heterogeneous and **dynamic** hardware and networking environment
- Integrated with **Apache Spark** with **Spark Streaming** supported
- Running FFT application (with **GPGPU support** if available) with streaming input and dynamic graphical output
- with integrated SoC and cluster **monitoring API**
- **Web-based UI** for SoC and cluster monitoring and visualizing application results
Scenario 2
Detection of anomalies in sea vessel movement using telemetry data

Telemetry data and its analysis
- Sensors: speed, course, roll, pitch, wind etc.
- Collecting data from sea vessels (NMEA channels)
- Preprocessing with Apache Spark
- Anomaly detection and prediction with ML
- Considering weather forecast for prediction
Data analysis and ML: Sea vessel telemetry data analysis

Data analysis with machine learning and big data technologies

- Collecting data from sensors on vessels (vessel speed, location, pitch, roll, wind speed, fuel flow, etc)
- Filtering, cleaning, merging data
- Analyzing data and using machine learning to:
  - discover anomalies
  - reveal events and processes, their causes
  - predict fuel consumption
- Using Python and Apache Spark
Anomaly detection in telemetry data

- Maneuvering anomalies
  - Abrupt (rough) maneuvers; turns/speed
- Prerequisites for collisions
  - Depth/speed; maneuvers with course and speed, taking into account CPA (Closest Point of Approach) – TCPA (Time to CPA); fuzzy logic for maneuvers with course and speed
- Stability
  - Detection of 1% of the most noticeable rolls; roll/pitch
- Marginal conditions
  - Drift angle/speed
- Manipulating controls
  - Shaft RPM control; Rudder control
Anomaly detection in telemetry data

- Detecting and tracking anomalies on the route (offline and online)
- Detecting abnormal situations caused by other ships approach
- Predicting fuel consumption
Scenario 3:

- Using IoT/CPS-related technologies for smart lighting, water control

- **Smart lighting control**: pilot project on cloud-based smart-home lighting control based on sensor data analysis, machine learning and prediction of light usage scenarios

- **Water sensors**: using autonomous GSM data transmitters, which periodically transmit measurements of water consumption to the data collector and analysis servers; build an hourly estimation of current urban water pipes load and produce their usage forecasts.

- **Beacons**: for indoors location and navigation
Smart lighting control

Cloud-based intelligent lighting control system

• Secure platform that collects and consolidates data on lighting and energy consumption from various sites to enable **energy-efficient lighting control**
• Distributed sensors and lighting devices **interconnect and exchange** data on light use
• **Data analysis** algorithms: electricity consumption for lighting in dynamics; monitor current status of lighting devices.
• **Machine learning**: train light control algorithms, discover light usage patterns, predict light use and automatically control light behavior based on this knowledge.
• **Cloud service**: create models of rooms by uploading room plans, make the arrangement of devices, combine the devices in groups and describe initial rules of light functioning.
Smart lighting control

Cloud-based intelligent lighting control system

- Using DEUS ME6 cloud platform to control sensors and lights

Collect sensor data

Build usage profiles

Create light usage models

Smart lighting control
Smart water consumption control

Smart infrastructure for water consumption control

• Based on **distributed sensors, water meters, GSM communication** between devices, data collection and analysis to fully automate water consumption technical billing

• **Big data analysis** allows users to build an hourly estimation of current urban water pipes load and produce their usage forecasts

• System of **customer profiles** is used to estimate sectoral water consumption

• Profiles allow users to **accurately estimate and predict water consumption** of similar objects in one class, even with incomplete data

• The system of such devices is **self-sufficient and constantly expanding**. Currently, there are about 13,000 of such devices that cover half of the city, where about 2 million people live.
Indoors location and navigation with Bluetooth Low Energy (BLE) Eddystone beacons

• Using beacons to navigate patients in hospitals

  • Eddystone is an open, scalable BLE beacon format that allows developers to create contextually aware experiences on both Android and iOS devices.
  • Eddystone supports multiple types of broadcast signals, or in BLE terms, “frames”.

• Indoors location and navigation with Bluetooth Low Energy (BLE) Eddystone beacons
Conclusion

- Variety of existing sensor data in many fields
- There’s a lot of those data
- You can do plenty of things with them:
  - Create tools and infrastructures to collect and analyze the data
  - Discover patterns, find something unusual, predict new data
  - Control other devices and processes
- It’s fun!
- Let’s collaborate on this!
Contact

Dr. Vladimir V. Korkhov, associate professor

e-mail: v.korkhov@spbu.ru

Faculty homepage: http://www.apmath.spbu.ru/en/staff/korhov/
LinkedIn: https://ru.linkedin.com/in/vladimir-korkhov-2a36181
Google Scholar: https://scholar.google.com/citations?user=TdeqrjgAAAAJ&hl=en
ResearchGate: https://www.researchgate.net/profile/Vladimir_Korkhov2