|  |  |  |
| --- | --- | --- |
| **Módulo 3 – Tema 2** | Sam Madden |  |

**Video 1:**

Los datos IoT son diferentes a otros tipos.

Restricciones 3B: batería, banda, bytes. Limitaciones que surgen de la naturaleza aproximativa de los datos IoT. Detección de atípicos y anomalías IoT. Minería ítems frecuentes (dominios no continuos): para explicar los valores atípicos a través de algoritmos.

Trabajos del profesor

* TinyOs. Para dispositivos con batería. Conectados a redes inalámbricas por saltos.
* TinyDb: tratamiento de colecciones de dispositivos como si fuera una DBMS, interactúa con lenguaje parecido a SQL,
* Cartel: telecomunicaciones entre vehículos. Para iOS iCartel app abierta de colaboración (tráfico). Antes que Waze, Google maps.
* Lutron: sistema de control de luces.
* DriveWell.

Flujo de datos: sensores, cels / puertas de enlace – cloud => **“Flujo de datos.png”**.

El procesamiento puede ocurrir en cualquier lado, solo que se va filtrando y en la cloud se tiene la dimensión real de todos los dispositivos.

Retos IoT en comparación con otros sistemas de procesamiento de datos:

* Los datos son distintos. Muestras del mundo físico. **No observan todo sino secuencia de puntos de tiempos discretos.**
* Sufren ruido.
* **Lidiar con recursos limitados (energía, banda. Almacenamiento), pérdida por ruido, anomalías (detectar y corregir).**

**Video 2:**

Caso de estudio: **“Arquitectura DriveWell.png”**. Driver score.

Requisitos:

* Etiqueta dura por varios años. Batería en el interior del auto.
* Bajo consumo del celular.
* Latencia muy pocos segundos.
* Feedback en minutos por ejemplo core del viaje.
* Precisión: aplicación se equivoca=>-.

**Video 3**

Arquitectos vs. Los que analizan datos:

Los analistas quieren captar la mayor info. fidedigna posible.

Arquitectos se ven limitados de recursos.

**3 Bs**

* Batería. La fundamental. Radio 3G, GPS utilizan muucha batería. Ejemplo de consumo en **“Ejemplo consumo batería.png” “Ejemplo consumo batería2.png”**.

Sutilizas:

* + Veces encendido apagado LTE radio demora 10 segundos por lo tanto no es eficiente enviar mensajes sin demorarlo lo suficiente y apagar la radio.
  + CPU rápida- tiempo peeeero CPU 100% consume batería, quizá utilizar CPU de menor potencia o ejecutar a menos del 100%
  + Ídem pero en potencia del ancho de banda.

+bando=>más batería peeeero +banda-tiempo-power.

Asimetría entre el consumo de cargas de subida vs. Bajada. 2 veces más.

**Video 4**

+consideraciones energía:

* Más datos => censar más => más procesamiento => más transmisión…. Colectar solo lo necesario exactamente lo que requiere el usuario.
* +CPU y –transmisión vs. –procesamiento +transmisión.
* **Wifi**/BluetothLE usa mucho menos energía que 3G/LTE.

Ejemplo reducción de datos colectados y transmitidos TinyDB (**“TinyDb cuenta.png“**):

* Base de datos de dispositivos + consulta.
* Multi salto data collection desde nodos remotos al principal conectado a Internet.
* Se recopila datos de energía eficiente + limitar ancho de banda.
* Estructura de árbol en el que están pasivos la mayor parte del tiempo. Solo activos en el caso de transmisión hijos.
* Minimizar el tiempo de activos, escuchar sus radios y cantidad de datos a transmitir.

**3 Bs**

* Ancho de banda: bluethoothLE se puede transmitir muy pocos datos, inclusive en 3G/LTE.

Consideraciones:

* + Monitoreo continuo vs. alertas (ya sea al ritmo normal o inmediato al usuario).
  + Utilización de buffers locales para atenuar clases de no conectividad / conectividad (**intermitencia** conectividad).

Se utiliza asks para garantizar que los datos hayan salido del buffer para removerlos.

Ejemplo: sensor almacena localmente->luego el enlace   
(cel) y finalmente Internet. Recién allí se eliminan las copias.

**Esto se lllama principio de extremo a extremo.**

**3 Bs**

* Bits. Storage: en muchos dispositivos, cel, flash es compacto, baja potencia y almacena muchísima información.
* Solo es preocupante si realmente hay que almacenar GB y enviarlo a algún lugar.
* **ES SECUNDARIO.**
* Se puede utilizar técnicas de compresión.

**Video 5**

Descripción aproximada de datos en IoT.

* Se miden a períodos discretos del tiempo.
* Precisión limitada.
* Los datos pueden no ser correctos.

GPS por ejemplo marca que pasa por el garaje, pérdida de datos ruido, también caídas.

Técnicas para tratar el ruido y pérdida de datos:

* Interpolación: si en una serie de datos falta uno, se une mediante línea recta. Regresión o ajustes de curvas para tratar datos faltantes. Extrapolación: predecir siguiendo tendencia luego de aplicar ajuste de curvas.
* Suavizado: ajuste de curvas eliminará picos de datos.
* Alineamiento.
* Tantas otras.

**Video 6**

Utilización de técnicas para pasar de discreto a continuo.

* FunctionDB tool **“FunctionDB.png”**. Para empleo de curvas con el objeto de mejorar el eficiencia de procesamiento de datos y responder consultas que no podíamos.
* Vtrack. Pareo de mapas, para datos imprecisos de los GPS: modelo oculto Markov.
* Nuevo método full path.

**Video 7 – Video 8**

Detección de valores atípicos y anomalías.

* Detección rápida de fallos, degradación. Otros ejemplos.
* Proceso: detector => clasificador => ranking.
* Detector: mediante estadísticas basadas en distribuciones de datos.
* Explicar de anomalías:

Clasificador: en 2 clases, típicos y atípicos.

Conjunto de elementos frecuentes: dominios discontínuos. Comparar frecuentemente los valores atípicos con los típicos.

Herramienta: scorpion.

**Enumere varias posibles aplicaciones de IoT, idealmente de su industria o dominio de experiencia. Para cada uno de éstos, piense en los requisitos de demora.**

**Caso1 - Pensando en una arquitectura Control Salud entre Paciente-Médico y diagramando 2 posibilidades: alertas automáticas y datos generales para diagnósticos procesados de forma periódica:**

Fitbit device (reloj, pulsera, dependiendo del paciente): informa datos obtenidos del paciente a los servidores fitbit. Esto ya se encuentra desarrollado y el intercambio de información de utilizando red wifi / conexión celular de forma **instantánea**.

Sistema propietario alojado en Amazon: creación de alertas que **se dispararán** hacia el paciente médico: ejemplo pico presión. Definición de **momentos en el día** de recolección de datos de control, seguimiento y diagnósticos de corresponder. Recolección de datos anteriores de la API de fitbit y publicación para la app. médico. API propietaria para detectar alertas y publicarlas para ser leídas desde las apps. **comunicándose en todo momento** con API fitbit.

App. paciente: **aviso instantáneo de alerta**. **Mensajería realtime médico-paciente**, etc.

App. médico: **aviso instantáneo de alerta**. **Mensajería realtime médico-paciente**, etc. Interfaz gráfica de análisis de datos del paciente disponibles al momento.

**Caso 2: sistema seguridad personal basado en cámaras y alarmas.**

Sistema wifi sensores + cámaras + central + alarma. **Realtime**.

Nodo central con chip 4G **informando en tiempo real** al sistema propietario.

Sistema propietario alojado en Amazon con API de **recepción de alertas de alarmas instantáneo** y habilitación de cámaras.

App. usuario con informe de alarma y visualización. Opciones de usuario de recording y almacenamiento en dispositivo externo al nodo central.