# Comparison between Local and Cloud-based Storage based on an "Internet of Things" System for Indoor Pet Monitoring

COMPUTER SCIENCE & ENGINEERING
TEXAS A&M UNIVERSITY

Henry J. Serrano henry.serrano@tamu.edu
Susana M. Caballero susana.mcl@tamu.edu
Dilma Da Silva dilma@cse.tamu.edu

TEXAS A&M UNIVERSITY

## Introduction

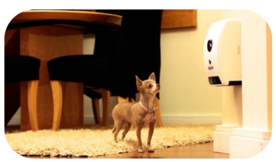
•The use of technology applied to daily life is a growing trend. Many applications monitor behavior.
•Stored data has become essential in many applications, many are based on local or cloud storage.
•Monitoring refers to a process based on gathering, observing and studying data. This data is then analyzed, based on events or time periods, to perform several actions. Analysis allows prediction of future events.
•Sensor data is often deployed to identify risk situations and to analyze how activities are performed.
•Systems for pet monitoring (letting owners now what happens in their absence) are attractive to customers and exhibit most of the challenges in processing data from IoT and Wearables.

## Objective

•Characterize differences between Local and Cloud-Based Storage based on an "Internet of Things" Modeled System for Indoor Pet Monitoring.
•Develop tools to enable benchmarking on pet monitoring workloads.

## Motivation for focusing on data services

•With the Internet of Things (IoT), more and more applications need to store data incoming from the real world. Current applications tend to store this data in the cloud. Therefore, comparisons between remote and local hosted databases need to be further examined and discussed.
•Most existing solutions use cloud servers. Is this the right approach?



## Approach

- Explore a case study based on Pet Monitoring applications.
- Develop a model behavior and tools for synthetic data generation.
- Prototype an implementation to run locally and on the cloud.
- Collection of experimental data.

## Tools and Platforms

- Leveraging previous knowldege:
  - Node.js based server.
  - Visual Studio Code and PyCharm as chosen IDEs.
- Skills acquired for this project:
  - MongoDB employed as the database.
  - API design for client interaction through HTTP requests.
  - Python programming skills.
  - Library c3.js for graph displaying.
- Platforms
  - Local on OSX; Cloud on AWS EC2 Linux.

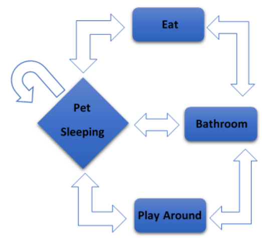



Figure 1. Pet Behavior Modeling.

## Pet Modeling

Pet Behavior was modeled accordingly to Figure 1. Then developed and executed as a script. Data generated was then uploaded to the local or cloud server.

The initial state of the model is 'sleeping' and from this one it is possible to move to the others states, and even remain in the initial one. For each state a probability was assigned. Upon this, an algorithm was coded on the script. A visual representation on it can be observed on Figure 2.

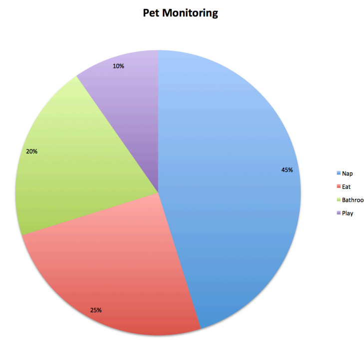



Figure 2. Actions defined by Pet Behavior Modeling.

## Methodology

- Benchmarking covers both local and cloud-based.
- Workload of 144 rows of data in JSON format.
- Upload through POST requests to the local or cloud server.
- Test scenarios varied in number of queries (1, 10, 100, 1000), same payload.
- Each test scenario execute 4 times.
- Further documentation available at http://petmonitoring.net78.net .

## Benchmark platform

- Local server and client based on a MacBook Pro Core i5 @ 2.7GHz 16GB Ram with solid state drive.
- Cloud server based on AWS EC2 instance at West-US Datacenter. Same client as the local one.

## Outcome

- Initial results indicate that deploying local server results in an average of 7ms per query. And the cloud-based server results in an average of 180ms.
- The time required to connect to the server and upload data was clearly not affected by the number of serial requests.

## Limitations

- Amount of data is not representative of real case scenarios.
- It was not explored how database scale with size and number, based on concurrent requests.
- It was not taken into consideration fluctuations in the communication backbone or load on the cloud provider.
- No diversity in queries (neither popularity or amount of data return).
- Photos and videos are relevant for pet monitoring systems but were not included in this study.
- Query latency was not broken down to identify which components have more impact on total delay.

## Conclusions

- Initial study confirms intuition that cloud-based queries take longer time to connect to than a local-based one.
- There is potentlal for batching query requests as to avoid lag.
- For continuously generated datastreams, compression can be useful.

## What's next?

- Enhance the Pet Behavior's Model and tool for synthetic generation of data.
- Perform further and more detailed benchmarkings, based on different local and cloud setup configurations.
- Gather data from physical sensors monitoring real pet behavior.
- Design and develop specialized hardware and software to monitor pet behavior, based on stsate-of-art Internet of Things devices.

## References

- Cui, H., Keeton K., Roy, I., Viswanathan, K., and Ganger, J. G. "Using Data Transformations for Low-latency Time Series Analysis" ACM SOCC (2015) – Pending Publication.
- Lutz, K., Allman, E., Wawrzynek, J., Lee, E., Kubiatowicz J., "The Cloud is Not Enough: Saving IoT from the Cloud" USENIX HotCloud, University of California, Berkeley (2015)
- Zhengming Tang, Harlan Hile, Sushil Bajracharya, Raja Jurdak, "PetTracker – Pet Tracking System Using Motes", Ubicomp 2005.
- http://www.dogmonitorapp.com/

## Acknowledgements

- CANIETI (Cámara Nacional de la Industria Electrónica, de Telecomunicaciones y Tecnologías de la Información).
- Computer Science Department of Texas A & M University.

https://github.com/TeamDStamu/PetMonitoring http://petmonitoring.net78.net