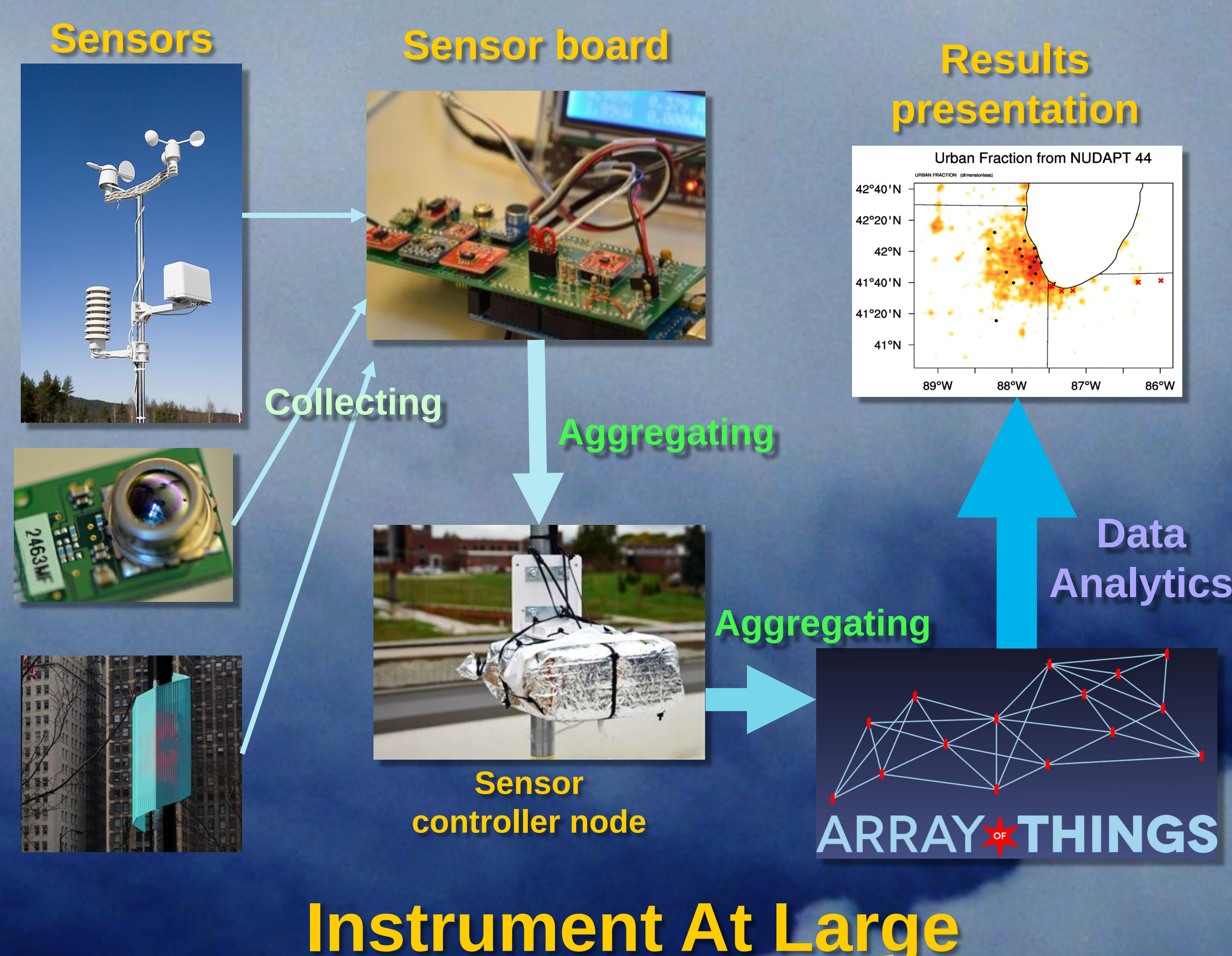




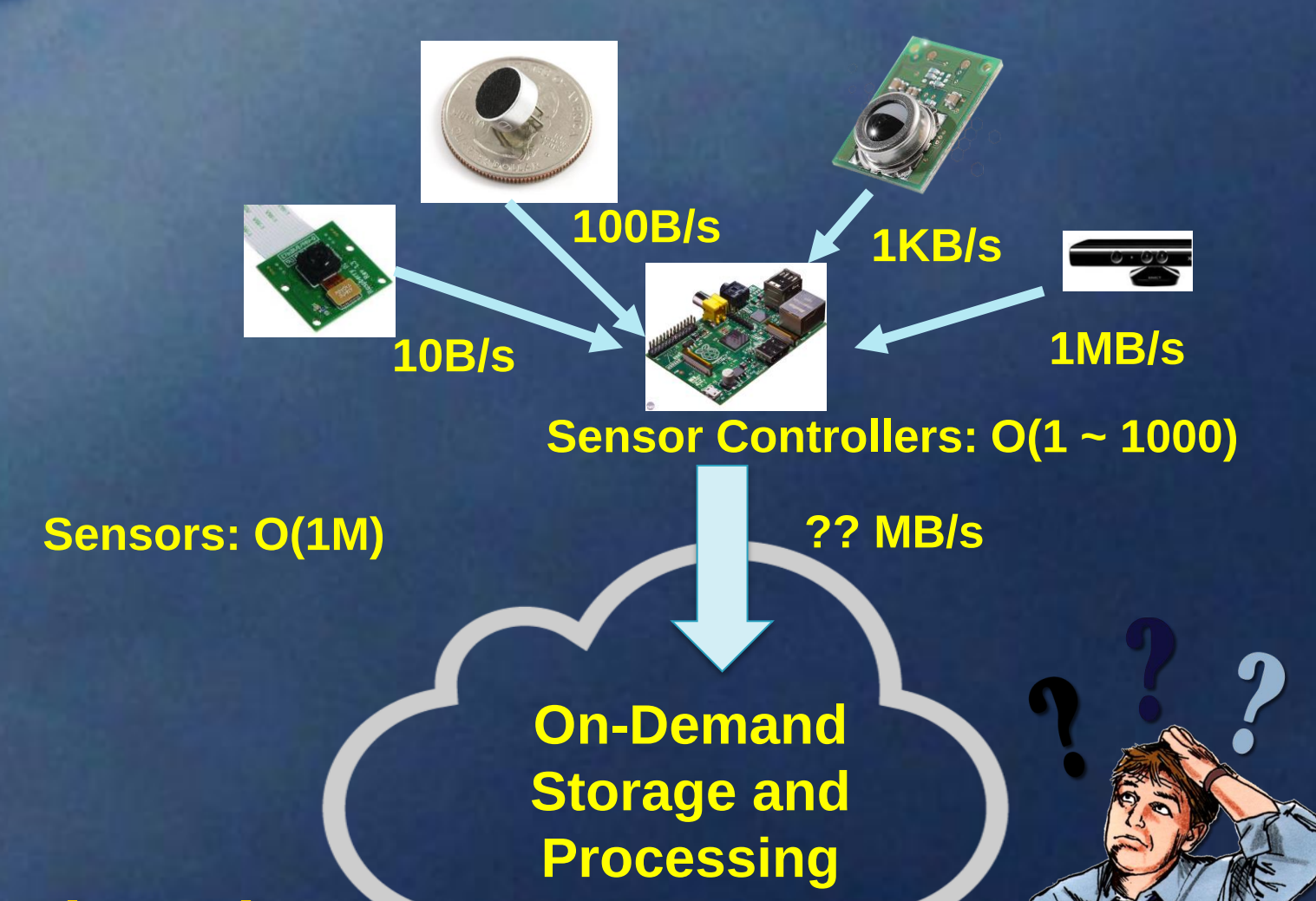
A Cloud-based Interactive Data Infrastructure for Sensor Networks

Tonglin Li, Kate Keahey, Rajesh Sankaran, Pete Beckman, Ioan Raicu

Small specialized sensor devices capable of both reporting on environmental factors and interacting with the environment are becoming increasingly ubiquitous, reliable and inexpensive. This transformation has enabled domain sciences to create "instruments at large" – dynamic and often self-organizing groups of sensors whose outputs are capable of being aggregated and correlated to support experiments organized around specific questions. This calls for an infrastructure that can support remote administration of sensors, relies on protocols that can withstand unreliable communications, and extend storage capability that can scale to support many data producing sensors, many different data types, and many end user requests. In this work we present protocols and a cloud-based data store called "WaggleDB" that address the above challenges. The system efficiently aggregates and stores data from sensor networks and enables users to query the data sets. The "WaggleDB" data store incorporates a scalable multi-tier architecture with individually scalable layers toward overcoming the challenges.



Sensor Network Data Streaming



Challenges

- Unreliable network
- Wide range of request rate
- Admins need interaction to remote sensors
- Potentially very high write concurrency
- Many sensor types: many data types
- Scalable architecture

Proposed Solution Features

- Multi-tier architecture
- Independent components in each tier
- Organize each tier as a Phantom domain for dynamic scaling
- Message queues as write buffers
- Transactional interaction by tracking all commands and responses in database
- Column-oriented database with semi-structured data for various data types

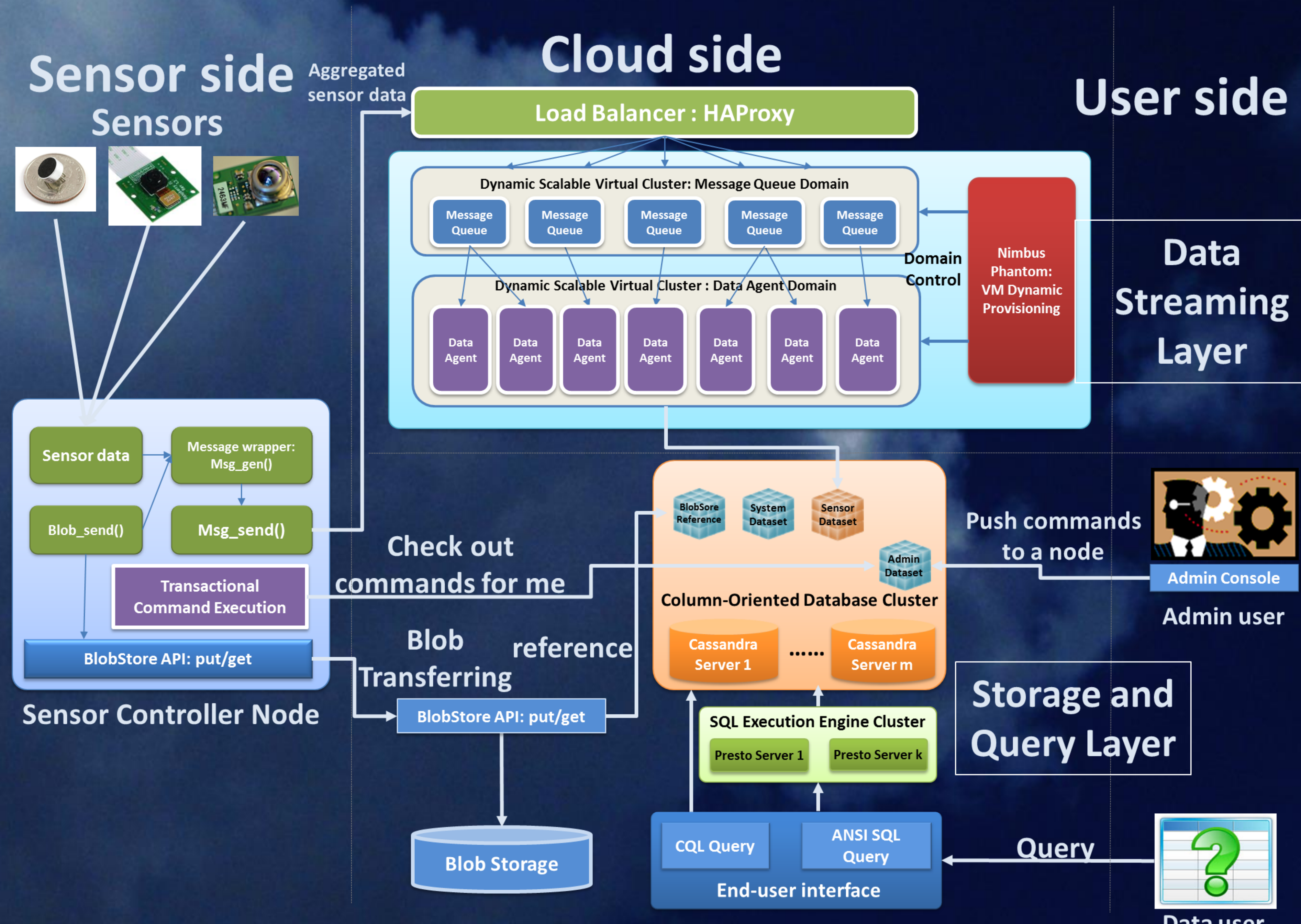
Architecture

User side

- Admin: execute commands on node controller
- Data user: query data through CQL or SQL

Cloud side

- Write buffer: message queues and consumers
- Storage: Column-Oriented database
- Data query layer: Query execution engine support standard SQL



Bandwidth

- Bandwidth increases with scales AND message sizes (within 10KB)
- Implies benefit of message batching
- Network was not saturated



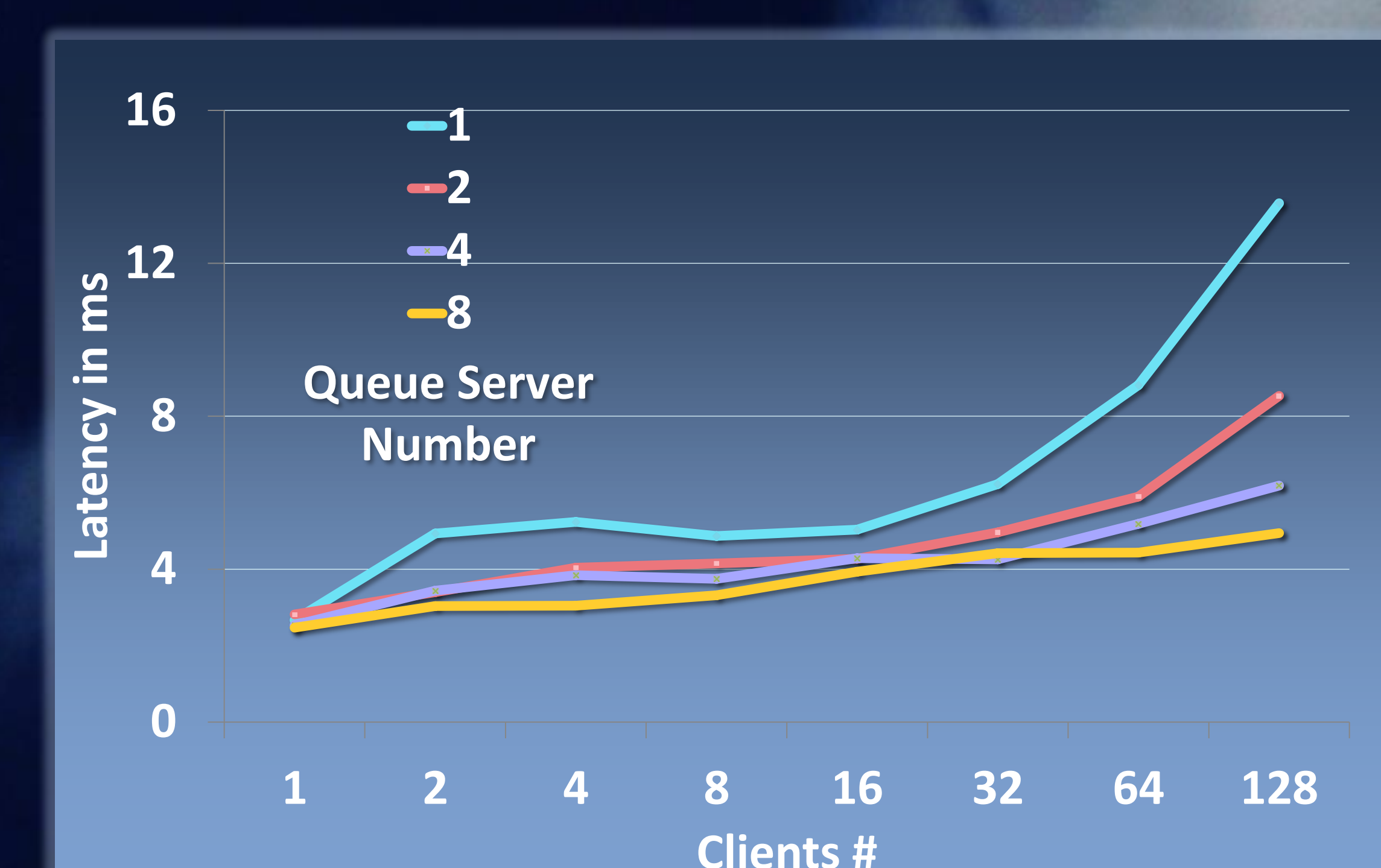
Latency

- Latency within 6ms on 32 nodes
- System scales well
- Message size doesn't impact much



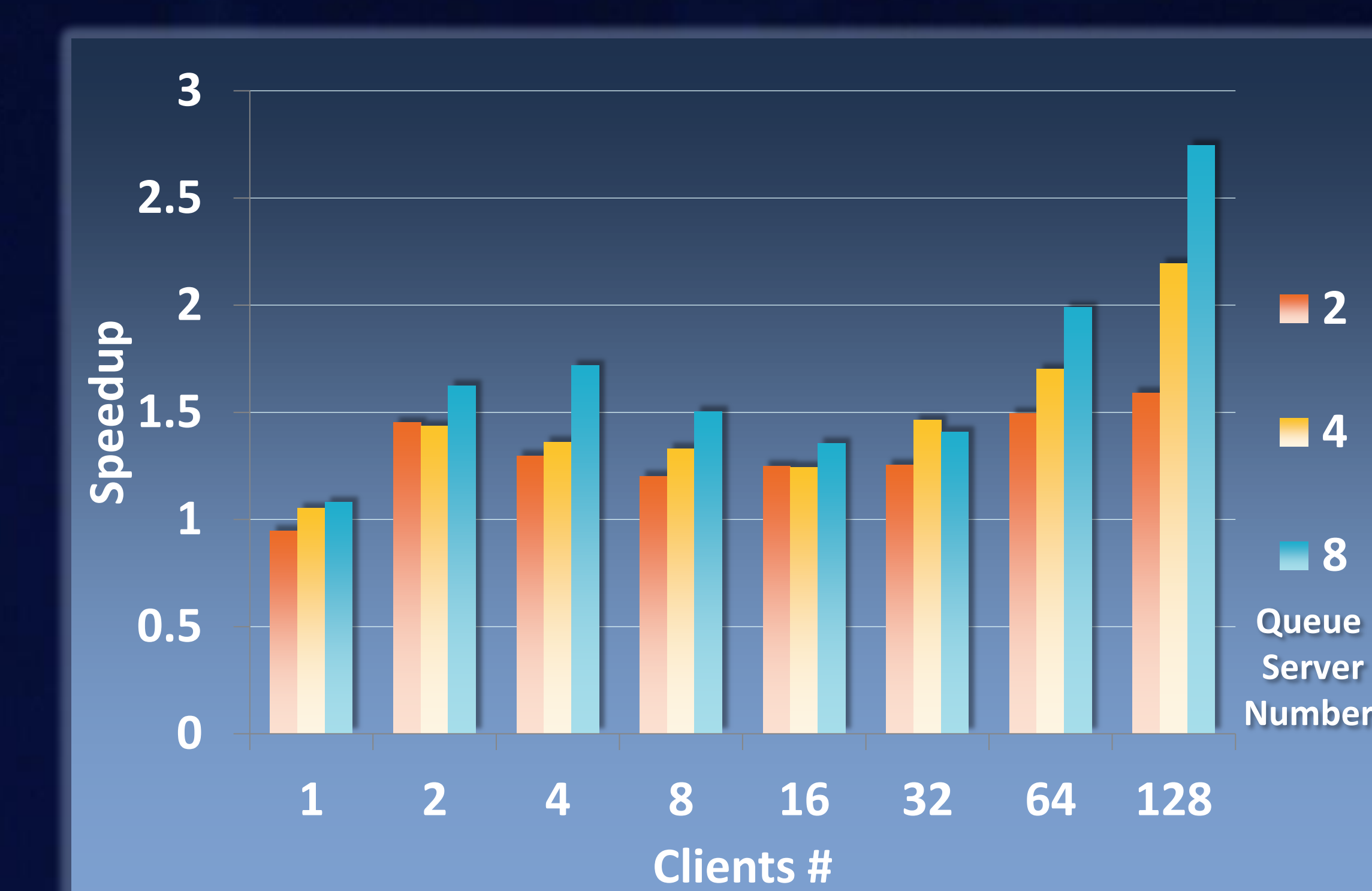
Scalability on Client and Server side

- Single queue server saturates with 32 clients scale
- System performance degrades gracefully when saturated



Speedup

- Speedup = T_1 / T_n
- Workload concurrency level associate to speedup: system not fully saturated



Conclusion

- Good performance and scalability at different concurrency levels
- Aggregating data and sending in batch has higher bandwidth

Future work

- Collecting and process data from city/state scale
- Advanced data analytics
- Integrated data portal for web apps

Acknowledgements



U.S. DEPARTMENT OF
ENERGY

Argonne
NATIONAL LABORATORY

ILLINOIS INSTITUTE
OF TECHNOLOGY

Future
Grid

Tonglin Li: Illinois Institute of Technology

Kate Keahey, Rajesh Sankaran and Pete Beckman: Argonne National Laboratory

Ioan Raicu: Illinois Institute of Technology, Argonne National Laboratory