DQBarge: Improving data-quality tradeoffs in large-scale Internet services

Michael Chow
Kaushik Veeraraghavan, Jason Flinn, Michael Cafarella
Complex Internet services

• Composed of hundreds of software components

• Requests have response time goals
Balancing response time goals

- Components have response time goals
  - Lower-level components unaware of response goals
  - Lower-level components may fail
Data-quality tradeoff

Explicit decision to return lower fidelity data
• Improve response time
• Minimize resource usage
Recommendation service

• “What should I do after grad school?”
Recommendation service

• “What should I do after grad school?”
Recommendation service

• “What should I do after grad school?”

1) Facebook
2) Google
3) Microsoft
4) Amazon
5) Academia
• “What should I do after grad school?”

1) Academia
2) Facebook
3) Google
4) Microsoft
5) Amazon
Outline

• Motivation
• Study of data-quality tradeoffs at Facebook
• DQBarge
• Evaluation of DQBarge
Study of tradeoffs at Facebook

• Systematic study of a Facebook service
  • Laser, key-value store at Facebook [2015]

• Categorized tradeoffs made by all 463 clients
>90% of clients perform tradeoffs

Top 50 clients: 90% tradeoff, 10% failure
All clients: 91% tradeoff, 9% failure

Data-quality tradeoffs are the norm, not the exception
Most tradeoffs are reactive

- Reactive → occurs on timeout/failure
- Proactive → only request what can be done
Most tradeoffs are reactive

- Reactive → tradeoff occurs on timeout/failure
- Proactive → only request data you expect to use

94% Reactive for Top 50 clients

98% Reactive for All clients

Reactive tradeoffs waste resources
Takeaways

• Data-quality tradeoffs are common
• Most are reactive, instead of proactive
• Tradeoffs only consider local information

Need global information to enable proactive, better tradeoffs
Outline

• Motivation
• Study of data-quality tradeoffs at Facebook
• DQBarge
• Evaluation of DQBarge
DQBarge

• Library for developers to help make tradeoffs

• Propagates additional data along causal path
• “What should I do after grad school?
DQBarge

• “What should I do after grad school?”
Phases of operation

• Offline phase: build models

• Online phase: use models
Quality model

Full Quality

1) Facebook
2) Google
3) Microsoft
4) Amazon
5) Academia
Quality model

Full Quality

1) Facebook
2) Google
3) Microsoft
4) Amazon
5) Academia

Work/Life
Quality model

Full Quality
1) Facebook
2) Google
3) Microsoft
4) Amazon
5) Academia

Work/Life
1) Facebook
2) Google
3) Microsoft
4) Amazon
5) Academia
Quality model

Full Quality

1) Facebook
2) Google
3) Microsoft
4) Amazon
5) Academia

Work/Life

1) Facebook
2) Google
3) Microsoft
4) Amazon
5) Academia

Teaching

1) Facebook
2) Google
3) Microsoft
4) Amazon
5) Academia
Phases of operation

• Offline phase: build models

• Online phase: use models
“What should I do after grad school?”
“What should I do after grad school?”
“What should I do after grad school?”
DQBarge

“What should I do after grad school?”
“What should I do after grad school?”

DQBarge
“What should I do after grad school?”
DQBarge

“What should I do after grad school?”
“What should I do after grad school?”

1) Facebook
2) Google
3) Microsoft
4) Amazon
5) Academia
Use cases of DQBarge
Use cases of DQBarge
Use cases of DQBarge

- Latency
- Quality
- Resource Usage
Use cases of DQBarge

Latency

Quality

Resource Usage
• Motivation
• Study of data-quality tradeoffs at Facebook
• DQBarge
• Evaluation of DQBarge
Evaluation

• Do data-quality tradeoffs improve performance?
• How much does provenance improve tradeoffs?
• How much does proactivity improve tradeoffs?
• How does DQBarge help in end-to-end scenarios?
  • Load spike
  • Utilizing spare resources
  • Dynamic capacity planning
Do proactive tradeoffs help?

Reactive vs. Proactive:
- Fraction of requests: 18% → 6%
- Quality drop: >10
Load spike scenario

3x load increase
Utilizing spare resources

![CDF graph showing the impact of utilizing spare resources on request latency](chart.png)

- Without utilizing spare resources: 15% drop
- With utilizing spare resources: 8% drop

Quality drop >10
Conclusion

• Data-quality tradeoffs are very common

• Suboptimal due to reactivity & lack of information

• DQBarge improves tradeoffs

Questions?