Jiyao Li
1. why 2PC (2-Phrase commit )is challenging in a cloud-scale distributed system?

2. As we know, Aurora is a relational database system, how does Aurora address the constraint of high throughput data processing in the side of storage of the network

Mason Pachner
1. What makes Amazon Aurora different than a standard SQL database?

2. How does the saving and recovery process work for Amazon Aurora?

Jesse McCammon
1. How does Aurora help companies or individuals achieve their goals better or more efficiently than systems that are already in place?

2. Is it possible for companies or individuals to achieve the same benchmarks as Aurora on their own without the use of the AWS platform?

Namita Raghuvanshi
1. How log helps in maintaining the consistency of the durable state, the runtime state, and the replica state and how is this log generated?

2. In general which recovery protocol is used in traditional databases and what are the tradeoffs? What approach Aurora uses to overcome it?

Matthew Richins
1. The paper focuses on all the benefits of this system. What could be some pitfalls or downsides to Aurora?

2. How is ‘self-healing’ achieved?

3. Based on this explanation, “We achieve this by replicating each data item 6 ways across 3 AZs with 2 copies of each item in each AZ.” How does Aurora scale with size?

Shubham Swami
1. How does the bottleneck for high throughput data processing move from compute and storage to Network? Is this an assumption?

2. How does a quorum-based model handle the correlated failures?

Kanak Tenguria
1. How "segmenting a storage volume and replicating each segment 6 ways with a 4/6 write quorum" going to effect performance of traditional databases like MySQL?

2. Section 3.3 states "A core design tenet for our storage service is to minimize the latency of the foreground write request." How is this achieved?