Reviewer #1

Questions

- **1. Overall Rating**
  - Weak Accept
- **2. Relevant for PVLDB**
  - Yes
- **3. Are there specific revisions that could raise your overall rating?**
  - Yes
- **4. Paper Category/Flavor**
  - Regular: Systems
- **5. Paper Summary. In one solid paragraph, describe what is being proposed and in what context, and briefly justify your overall recommendation.**
  - The paper presents a middle tier with a neural-network-based isolation level prediction approach for self-adaptive isolation level selection to achieve serializability by dynamically selecting between serializable and lower isolation levels that benefit from the performance gains of less synchronization. The graph-learned model predicts the isolation levels based on runtime workload characteristics. The approach dynamically validates run-time dependencies and schedules their commit order. Vulnerable dependencies are identified and a lightweight validation mechanism mitigates the overhead.
- **6. Three (or more) strong points about the paper. Please be precise and explicit; clearly explain the value and nature of the contribution.**
  - S1 Self-adaptive isolation level selection based on based workload characteristics is a good use case for predictive models
    S2 Articulation of performance gains on 3 differents workloads and varying skews.
    S3 Clear experimental setup (PostgesSQL, TxnSails, Benchbase) with stated goals.
- **7. Three (or more) weak points about the paper. Please clearly indicate whether the paper has any mistakes, missing related work (provide references), or results that cannot be considered a contribution; write it so that the authors can understand how to improve their paper.**
  - W1: No reasons are provided why this cannot be integrated in a OLTP TMS (see details).
    W2: A neural network is used that learns a multi-class classifier. The used model is very powerful, yet may be overkill. Comparisons with simpler models, or even rule-based approaches, for self-adaptive isolation level selection are missing.
- **8. Novelty; justify your answer in Q13. Please give a high novelty rating to papers on new topics, opening new fields, or proposing truly new ideas; give medium**

ratings to "delta" papers and those on well-known topics but still with some valuable contribution. For SDS and EA&B papers, novelty does not need to be in the form of new algorithms or models. Instead, novelty for SDS can be new understanding of issues related to data science technologies in the real world. Novelty for EA&B can be new insights into the strengths and weaknesses of existing methods or new ways to evaluate existing methods. For Vision papers, novelty can be futuristic information systems and architectures or anticipated new challenges. Submissions would describe novel insights and projects that are in an early stage, but with a strong promise of eventual high impact; they should include a broader and more compelling problem statement, with a higher novelty bar.**

    - With some new ideas
- **9. Significance**
    - Improvement over existing work
- **10. Technical Depth and Quality of Content. For Vision papers, quality depends on the convincing description of novel insights, systems and architectures, as well as the related research challenges, and future research directions.**
    - Solid work
- **11. Experiments. EA&B papers should have a higher bar for experiments. Vision papers are not expected to include experiments; you may choose "Not applicable".**
    - Very nicely support the claims made in the paper
- **12. Presentation**
    - Excellent: careful, logical, elegant, easy to understand
- **13. Detailed Evaluation (Contribution, Pros/Cons, Errors). Please number each point and provide as constructive feedback as possible.**
    - Although experimantel baselines and related work are providing are nicely provided, my biggest question for this paper is why old TP Monitors systems such as Tuxedo, Encina, or CICS are not considered? I understand that the main focus of the paper is the workload driven, self-adaptive isolation level selection using a neural network multi-class classifier, but in term of transaction scheduling the mentioned TP Monitors are fundamental.

      Typos/Grammar:
      page 4 bottom right: " ... concurrent transaction *Tj* is reading the same data item and undergoing validation." ??
      page 5 bottom right: "...where each row in A represents the feature vector of an operator..". It should be "... where each row in V ..."

- **14. Revision. If revision is required, list specific required revisions you seek from the authors (remember: the goal of a revision is not to make a paper the final and definite answer to a problem; only include the revisions necessary for the arguments of the paper to be well supported). Please number each point. If revision is not required, simply add "N/A".**

    - addrees weak points, and details.
- **15. Your Confidence in Review**
    - Knowledgeable in this sub-area

Reviewer #3

Questions

- **1. Overall Rating**
    - Weak Accept
- **2. Relevant for PVLDB**
    - Yes
- **3. Are there specific revisions that could raise your overall rating?**
    - Yes
- **4. Paper Category/Flavor**
    - Regular: Systems
- **5. Paper Summary. In one solid paragraph, describe what is being proposed and in what context, and briefly justify your overall recommendation.**
    - This paper describes TxnSails, a middleware system that provides serializable isolation to applications while executing transactions with a variety of isolation levels on the backend database (PostgreSQL in their testing). The middleware is given a set of transaction templates before the workload starts: they are analyzed to identify dependencies between transaction templates. This information, together with runtime tracking of read- and write-sets in the middleware and are used to validate that transactions are serializable before they are permitted to commit.
- **6. Three (or more) strong points about the paper. Please be precise and explicit; clearly explain the value and nature of the contribution.**
    - S1. TxnSails applies existing theory explaining when executions will be serializable when run at weaker isolation levels in a new way, showing that middleware can add validation to transactions to improve performance without sacrificing correctness.

      S2. The approach to adaptively choosing isolation levels, and reasoning about how to safely transition between isolation levels is novel, as far as I know.

      S3. The design and implementation appear to be reasonably general purpose -- i.e., it is possible to write new clients without modifying the middleware.
- **7. Three (or more) weak points about the paper. Please clearly indicate whether the paper has any mistakes, missing related work (provide references), or results that cannot be considered a contribution; write it so that the authors can understand how to improve their paper.**

    - W1. TxnSails is quite a complex solution: it needs static information about all transactions pre-supplied before the workload starts, modifies the database schema to add version information, mirrors data from the database so that validity checks can be done locally and trains two neural networks to predict the optimal isolation level for a given workloads. While the backend DBMS is unmodified, clients have to be written to the TxnSails API.

      W2. The paper claims that the techniques are general purpose, and that a benefit of the approach is that it can be easily applied to multiple DBMS backends but the only one that is evaluated is PostgreSQL. It would strengthen the claims if it could be shown working with another system (e.g., MySQL).

      W3. In terms of related work and the experimental evaluation, I was surprised that deterministic concurrency control (i.e., Calvin) was not covered. It also needs transaction templates to be pre-declared and outperforms locking and MVCC in some of the same scenarios where TxnSails claims advantages.

      W4. While the paper acknowledges in section 4.1.3 that the design does not handle range queries or prevent phantom reads, the issue is deeper than suggested there. It is not a simple matter to bolt an existing solution into this architecture because the middleware does not have access to the information it needs to know which keys or values are accessed by the backend DBMS to evaluate a predicate. The middleware would need to mirror more data (e.g., indexes) as well as reimplement complex query logic to efficiently prevent phantoms.
- **8. Novelty; justify your answer in Q13. Please give a high novelty rating to papers on new topics, opening new fields, or proposing truly new ideas; give medium ratings to "delta" papers and those on well-known topics but still with some valuable contribution. For SDS and EA&B papers, novelty does not need to be in the form of new algorithms or models. Instead, novelty for SDS can be new understanding of issues related to data science technologies in the real world. Novelty for EA&B can be new insights into the strengths and weaknesses of existing methods or new ways to evaluate existing methods. For Vision papers, novelty can be futuristic information systems and architectures or anticipated new challenges. Submissions would describe novel insights and projects that are in an early stage, but with a strong promise of eventual high impact; they should include a broader and more compelling problem statement, with a higher novelty bar.**
    - With some new ideas
- **9. Significance**
    - Improvement over existing work

- **10. Technical Depth and Quality of Content. For Vision papers, quality depends on the convincing description of novel insights, systems and architectures, as well as the related research challenges, and future research directions.**
    - Solid work
- **11. Experiments. EA&B papers should have a higher bar for experiments. Vision papers are not expected to include experiments; you may choose "Not applicable".**
    - OK, but certain claims are not covered by the experiments
- **12. Presentation**
    - Excellent: careful, logical, elegant, easy to understand
- **13. Detailed Evaluation (Contribution, Pros/Cons, Errors). Please number each point and provide as constructive feedback as possible.**
    - Mostly covered in S1-S3 and W1-W4.

      Overall this paper has some good ideas and solves some hard problems, but doesn't fully support claims that the approach is easily portable and general purpose.
- **14. Revision. If revision is required, list specific required revisions you seek from the authors (remember: the goal of a revision is not to make a paper the final and definite answer to a problem; only include the revisions necessary for the arguments of the paper to be well supported). Please number each point. If revision is not required, simply add "N/A".**
    - R1. I would prefer the paper to list the assumptions and limitations of the work in the introduction. In particular, the chosen setting only supports access by primary key (so the middleware can maintain read and write sets). It also requires all transactions to have a pre-declared template (so ad hoc transactions are not supported). In the current draft, these assumptions are buried later in the paper.

      R2. The limitations of a middleware solutions were not explored in the evaluation. For example, what is the overhead of garbage collection, what happens with larger workloads (when the metadata or lock table grow to a significant fraction of RAM)? If the backend database is elastically scalable (e.g., https://neon.tech/), is there a point at which TxnSails becomes the bottleneck?
- **15. Your Confidence in Review**
    - Expert in this problem

Reviewer #4

Questions

- **1. Overall Rating**
    - Weak Reject
- **2. Relevant for PVLDB**

    - Yes
- **3. Are there specific revisions that could raise your overall rating?**
    - Yes
- **4. Paper Category/Flavor**
    - Regular: Systems
- **5. Paper Summary. In one solid paragraph, describe what is being proposed and in what context, and briefly justify your overall recommendation.**
    - This paper introduces a system, TxnSails, that adaptively selects isolation levels for serializable transaction execution. TxnSails sits between applications and databases so it can be easily used together with existing databases. TxnSails includes three techniques: 1) user-level concurrency control for executing transactions under lower isolation levels while maintaining serializability; 2) an ML-based policy for selecting the right isolation level; and 3) cross-isolation validation when switching the isolation level. The experiments show that TxnSails outperforms existing approaches significantly.
- **6. Three (or more) strong points about the paper. Please be precise and explicit; clearly explain the value and nature of the contribution.**
    - S1. The paper selects an interesting and important problem.
      S2. This paper has articulated interesting insights for motivating their techniques.
      S3. The experiments are somehow comprehensive and show promising results.
- **7. Three (or more) weak points about the paper. Please clearly indicate whether the paper has any mistakes, missing related work (provide references), or results that cannot be considered a contribution; write it so that the authors can understand how to improve their paper.**
    - W1. The overhead of TxnSails is not fully discussed or evaluated.
      W2. It is unclear how to track the dataset to be accessed by a given SQL statement.
      W3. The correctness of cross-isolation validation is not complete.
      W4. The proof is somewhat confusing.
- **8. Novelty; justify your answer in Q13. Please give a high novelty rating to papers on new topics, opening new fields, or proposing truly new ideas; give medium ratings to "delta" papers and those on well-known topics but still with some valuable contribution. For SDS and EA&B papers, novelty does not need to be in the form of new algorithms or models. Instead, novelty for SDS can be new understanding of issues related to data science technologies in the real world. Novelty for EA&B can be new insights into the strengths and weaknesses of existing methods or new ways to evaluate existing methods. For Vision papers, novelty can be futuristic information systems and architectures or anticipated new challenges. Submissions would describe novel insights and projects that are in an early stage, but with a strong promise of eventual high impact; they should**

include a broader and more compelling problem statement, with a higher novelty bar.
    - Novel
- **9. Significance**
    - Improvement over existing work
- **10. Technical Depth and Quality of Content. For Vision papers, quality depends on the convincing description of novel insights, systems and architectures, as well as the related research challenges, and future research directions.**
    - Excellent work
- **11. Experiments. EA&B papers should have a higher bar for experiments. Vision papers are not expected to include experiments; you may choose "Not applicable".**
    - OK, but certain claims are not covered by the experiments
- **12. Presentation**
    - Reasonable: improvements needed
- **13. Detailed Evaluation (Contribution, Pros/Cons, Errors). Please number each point and provide as constructive feedback as possible.**
    - D1. TxnSails requires accessing the tuples in the database to do the validation, which could be costly. Therefore, it proposes caching hot tuples to reduce this cost. However, it does not evaluate the performance impact of accessing tuples in the database. The experiments did not show the percentage of data tuples being cached in TxnSails and the performance impact by a higher number of accesses to the database.

      D2. To roll back uncommitted transactions under TxnSails failures, does TxnSails need to log the uncommitted transactions? If so, this overhead should be evaluated.

      D3. TxnSails needs to track the dataset that is accessed by an SQL statement to do the validation. It is unclear how TxnSails supports it. If the SQL is on a primary key, it might be easy. However, if the SQL is not on a primary key (e.g., update the scores for all students with a given name), then tracking the dataset is challenging. Will TxnSails modify the SQL to return the tuples the SQL has accessed? Another challenging case is when the SQL is an aggregation, which does not tell the accessed data tuples.

      D4. The biggest issue for me is that the correctness of cross-isolation validation is not fully proven.

      The correctness of cross-isolation validation depends on Definition 7. It proves that the cross-isolation validation can detect and avoid cross-validation vulnerable dependencies in Definition 7. However, are these the only cross-validation vulnerable dependencies? This paper did not give a proof.

D5. The detailed steps of the proof in 5.2 are hard to follow.

For example, it mentions “Without loss of generality, we assume T_0 is the first transaction committed” I am not sure why it does not lose generality. T_0/T_1/T_2 have asymmetric dependencies. Assuming T_1 is the first transaction committed will be different from the case for T_0.

Another example that I do not understand is how the text leads to the highlighted conclusions (e.g., T_1 must operate under SER and in other words …).

- **14. Revision. If revision is required, list specific required revisions you seek from the authors (remember: the goal of a revision is not to make a paper the final and definite answer to a problem; only include the revisions necessary for the arguments of the paper to be well supported). Please number each point. If revision is not required, simply add "N/A".**
  - D1-D5
- **15. Your Confidence in Review**
  - Knowledgeable in this sub-area