Q1. What is the problem the authors are trying to solve?

File Prediction

Q2. What other approaches or solutions existed at the time that this work was done?

1. Griffioen and Appleton use probability graphs to predict future file accesses
2. Lei and Duchamp use pattern trees to record past execution activities of each program
3. Vitter, Curewite, and Krishnan adopt the technique of data compression to predict next required page
4. Kroeger and Long predict next file based on probability of files in contexts of FMOC
5. Patterson et al. develop TIP to do prediction using hints provided from modified compilers
6. Chang and Gibson design a tool which can transform UNIX application binaries to perform speculative execution and issues hints
7. Mowry et al. use modified compiler to provide future access patterns for out-of-core applications

Q3. What was wrong with the other approaches or solutions?

Over a period of time, prediction accuracy will decrease

Q4. What is the authors' approach or solution?

Develop a Program-based Last Successor model

Q5. Why is it better than the other approaches or solutions?

PLS records and predicts program-specific last successors for each file that is accessed

Q6. How does it perform?

Collect the metadata in terms of <program name, successor> for each file. Use this kind of metadata to records and predicts program-specific last successors for each file that is accessed.

Q7. Why is this work important?

Disk operations have become more expensive in terms of the number of CPU cycles spent waiting for them to complete. File prediction can mitigate this problem by prefetching files into cache before they are accessed

Q8. Can any improvement be done?

If this algorithm is suit for many program is excellent, because it suit for fewer program.