

Process Switches and Branch Prediction Accuracy

ELEC 525 – Adv Microprocessor
Architecture – Spring 2005

David Chen, Bennett Lau, Jeff Shafer

Hypothesis

- Running multiple processes pollutes the history in branch predictors due to aliasing. This decreases accuracy
- Branch prediction accuracy can be improved over existing techniques
 - The history table can be partitioned to allow each process (or a subset of the most frequently executed processes) to have their own history

Obtaining Instruction Traces

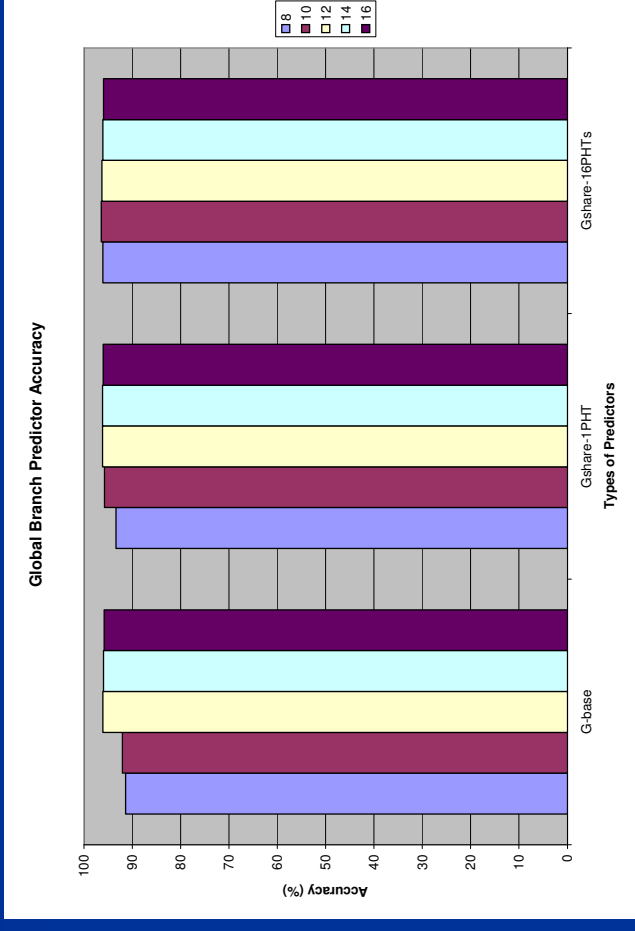
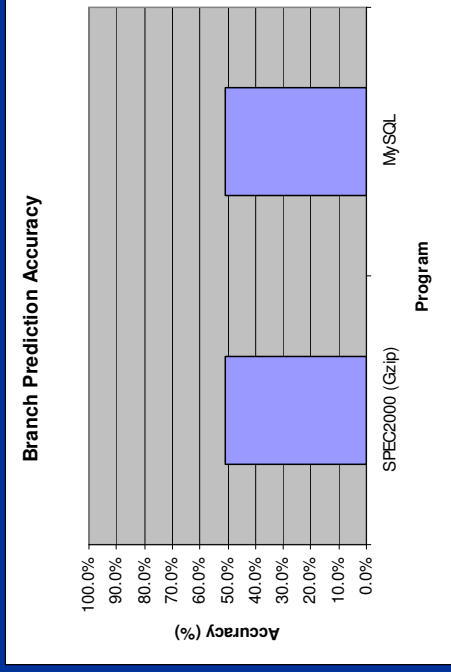
- Used Simics – Whole system simulator
 - Simulated OS: RedHat Linux 7.3
 - Modified OS kernel task scheduler
 - Calls Simics “magic function” with new process ID
 - Modified Simics trace module to catch magic function and output ID inline with instruction trace
- Installed application suite: Apache web server, MySQL database, SPEC 2000

Branch Predictor Simulator

- Written in Perl
- Parses instruction trace, predicts branches, and compares to actual result in trace file
- Two algorithms
 - Two-Level Global Predictor
 - 12 branch history window, 4K level two history, GShare, 2-bit counter
 - Same predictor plus separate history tables for each process
- Instrumented for history table comparison before & after process switch

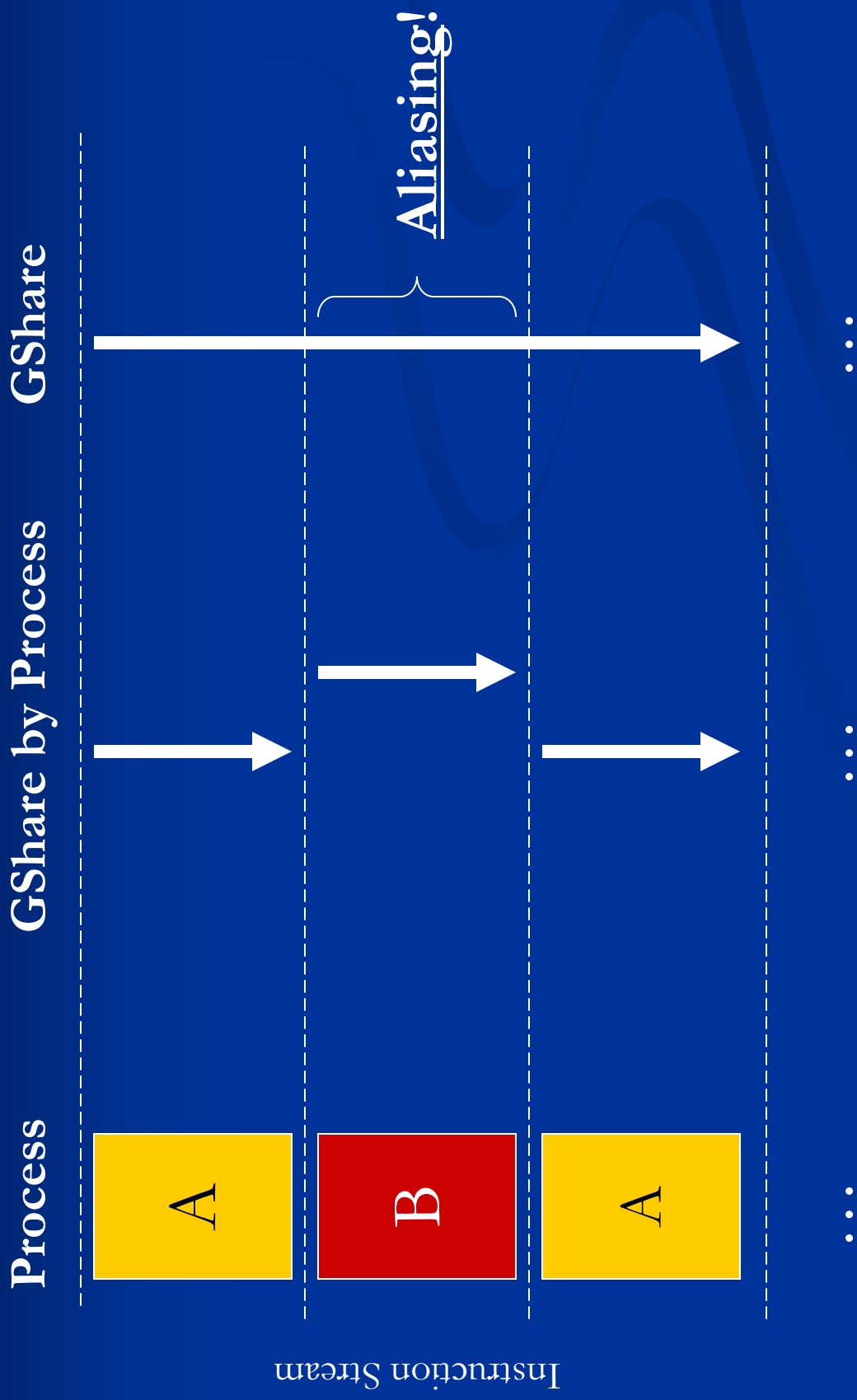
Branch Predictor Simulator

- Original results from prior presentation:
- Corrected Simulator:

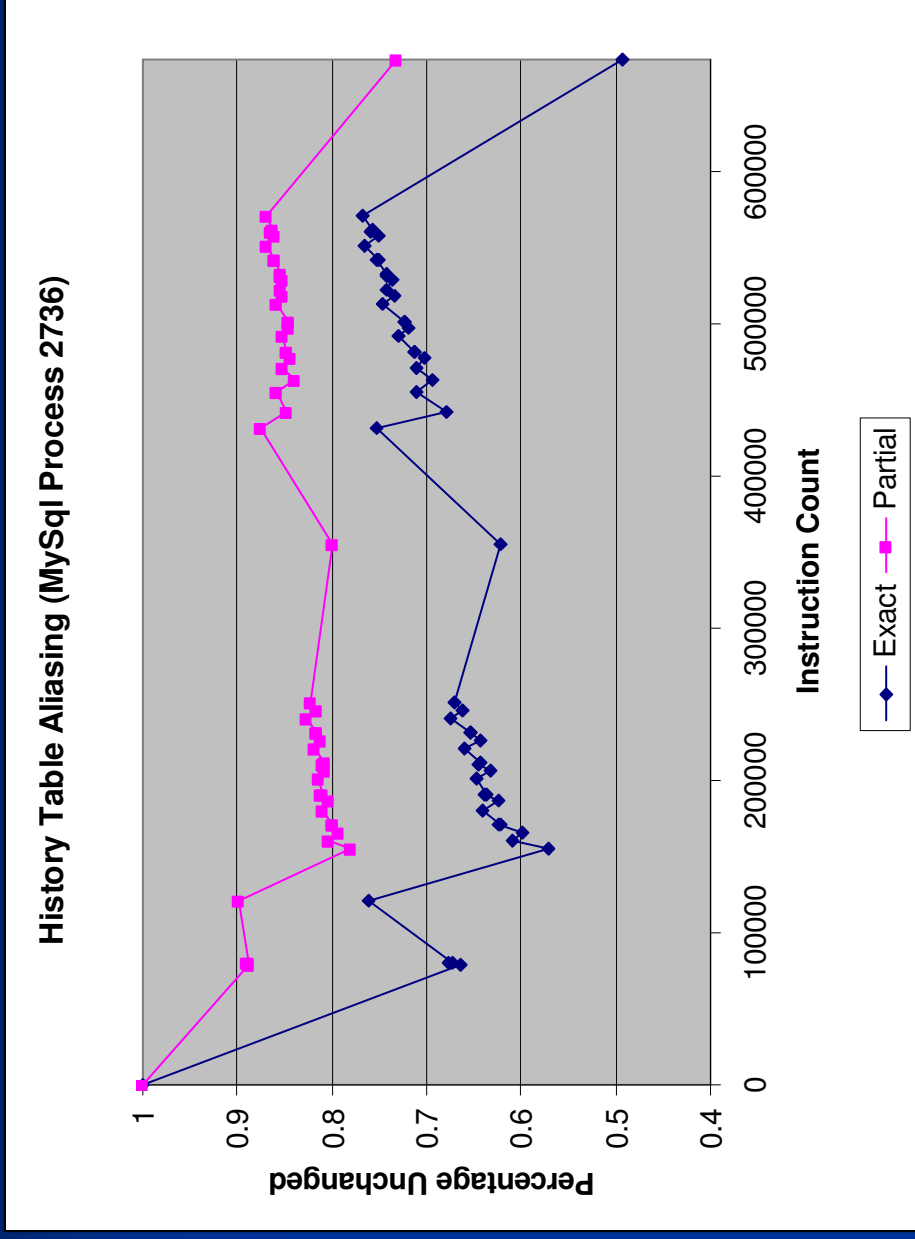


- Experimented with history table size, window length, and initial history value to optimize accuracy

What is Aliasing?



Does Aliasing Exist?



- Clearly Aliasing Exists – Is it destructive?

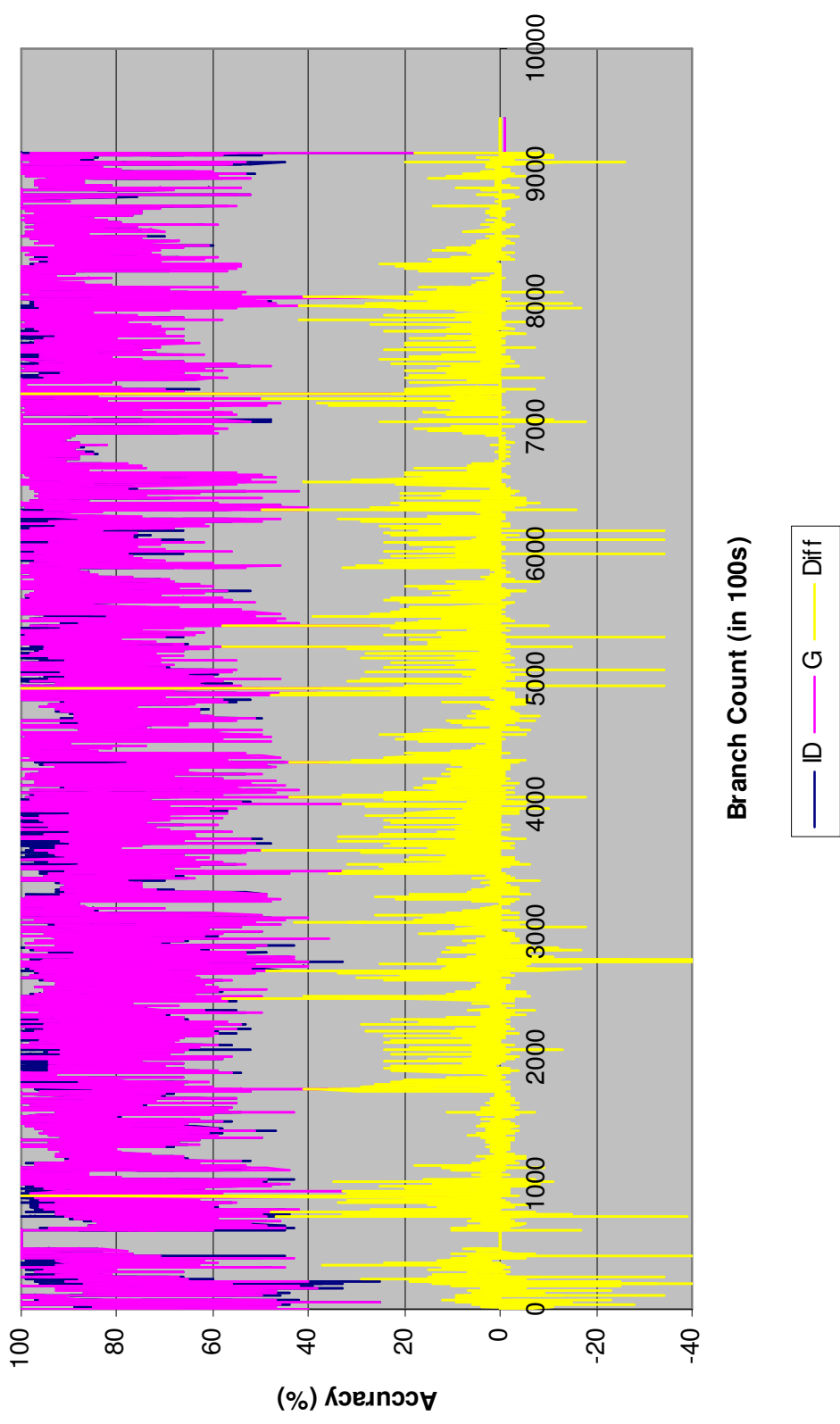
Accuracy and Utilization

	ID Branch Predictor Accuracy (%)	G Branch Predictor Accuracy (%)	Accuracy Difference (%)
Apache 1	88.68	88.48	0.20
Apache 2	88.18	87.92	0.26
Apache 3	88.20	87.79	0.41
KDE 1	86.85	86.10	0.75
KDE 2	90.48	88.99	1.49
KDE 3	90.34	89.70	0.68
KDE 4	91.16	90.73	0.43
MySQL	87.62	84.76	2.86
Pine	86.33	85.77	0.56
SPEC2000 Gzip	87.04	86.93	0.11
SPEC2000 Mgrid	86.31	86.16	0.15
VI	87.94	86.04	1.90

- Removing aliasing improves prediction accuracy (Thus, the aliasing was destructive overall)

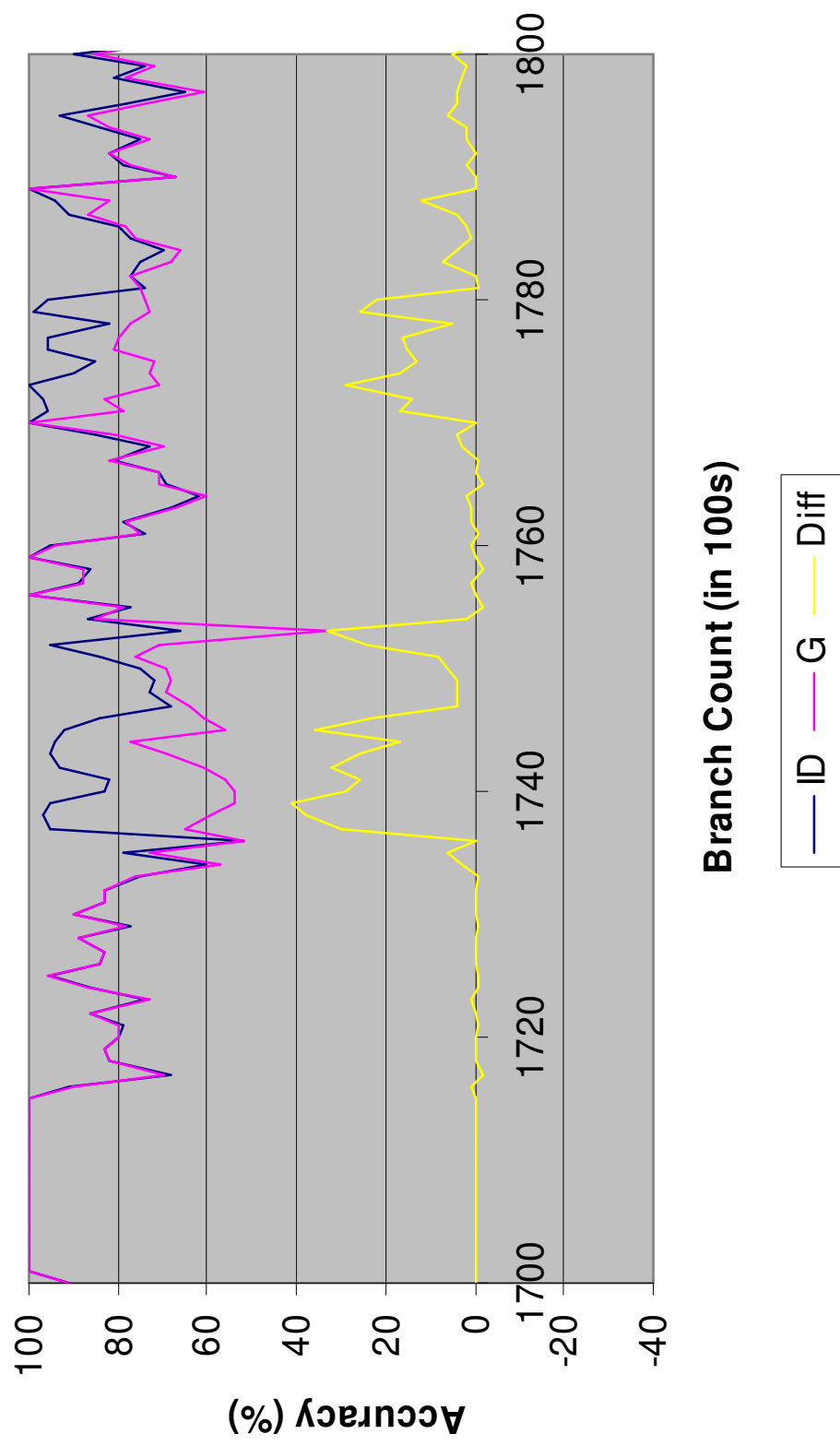
Why Does Performance Improve?

MySQL Accuracy Graph



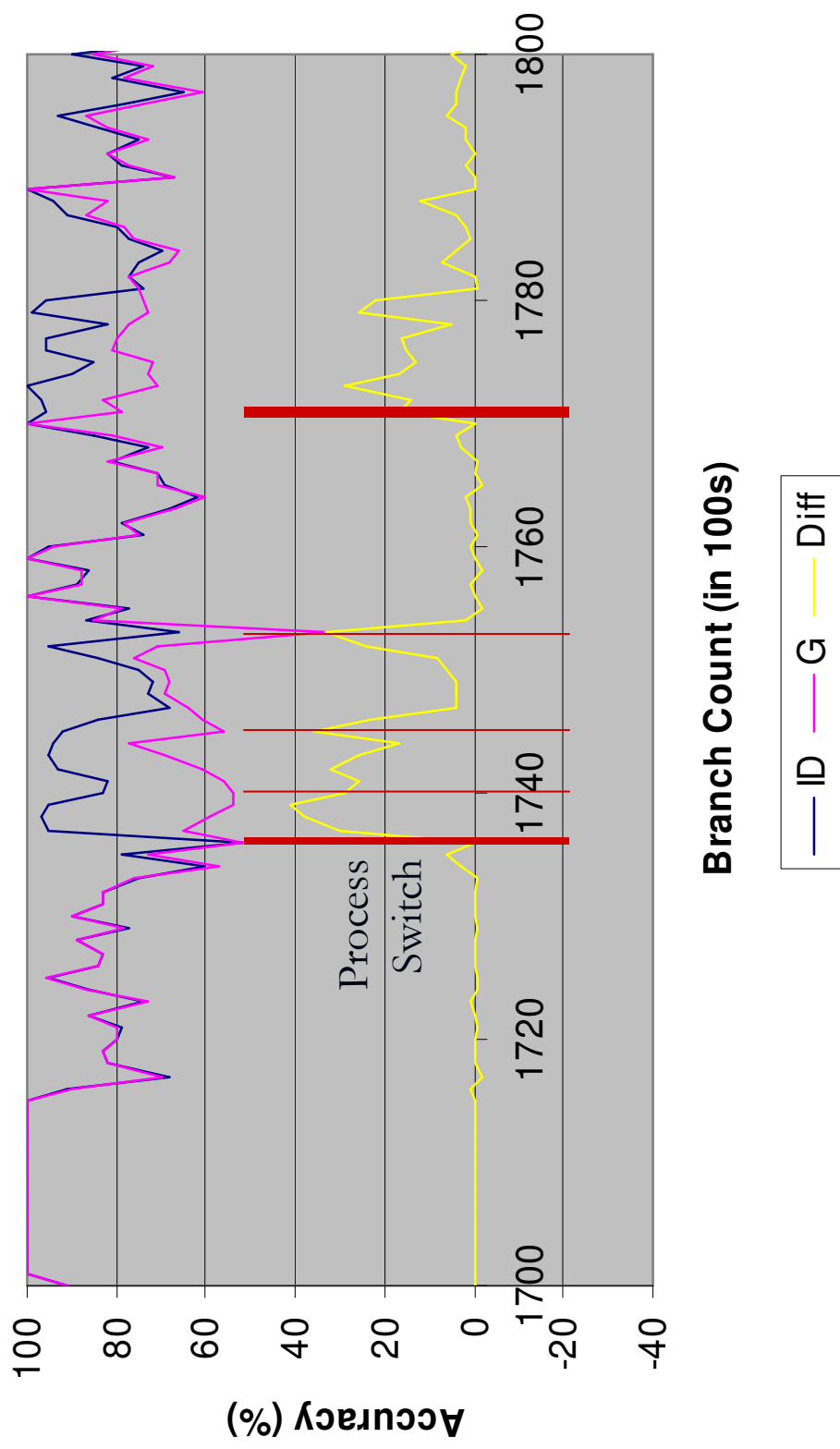
Where Do We Improve Performance?

MySQL Accuracy Graph

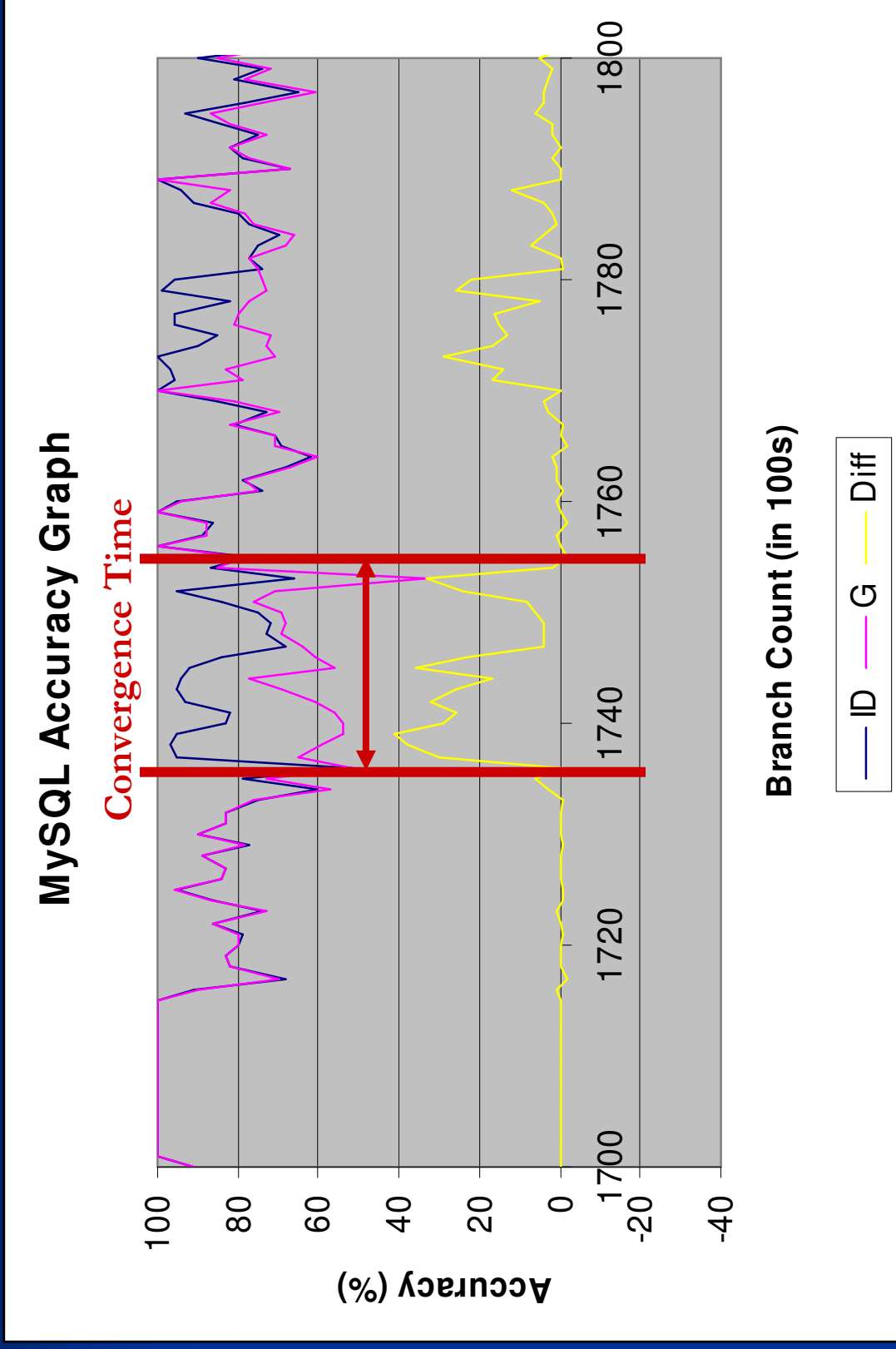


Why Does Performance Change?

MySQL Accuracy Graph

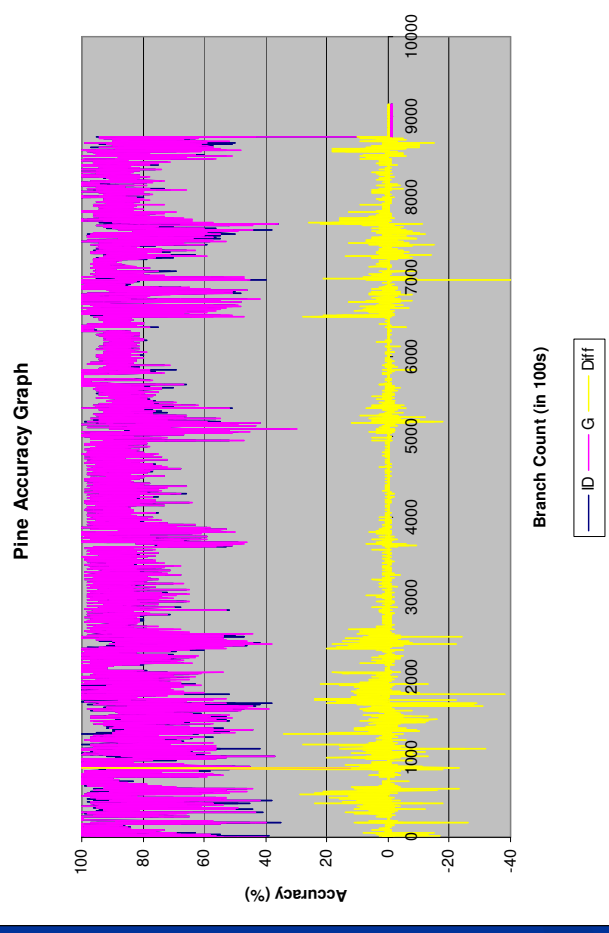
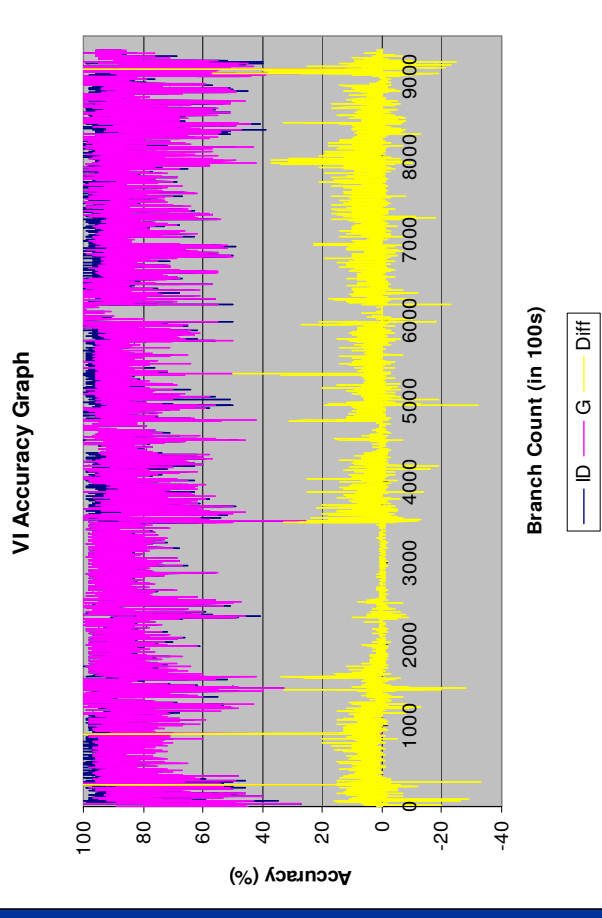


Why Does Performance Change?



Do Other Traces Show Improved Performance?

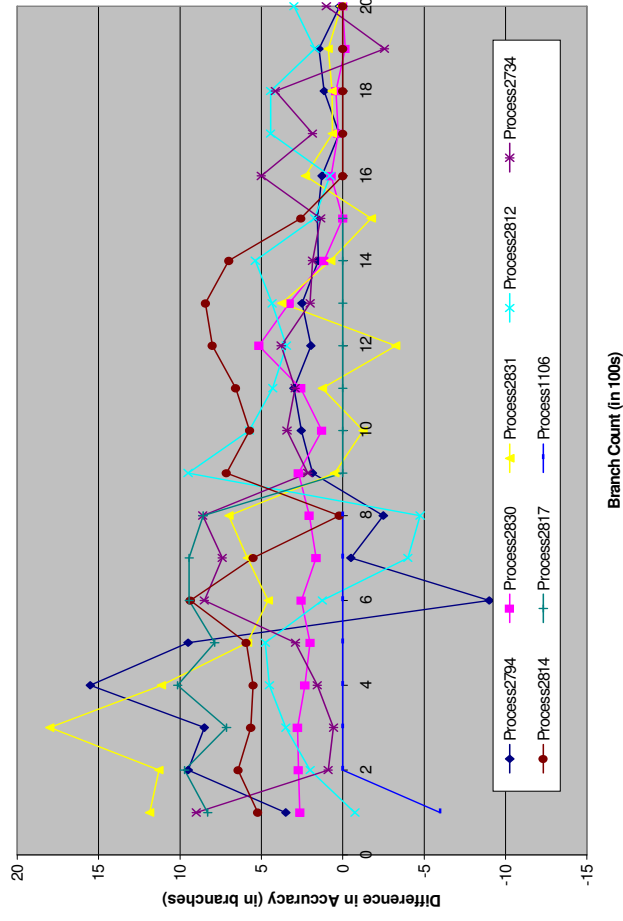
- This behavior was present on all traces
- Some show more periods of convergence
 - Directly related to task switching frequency



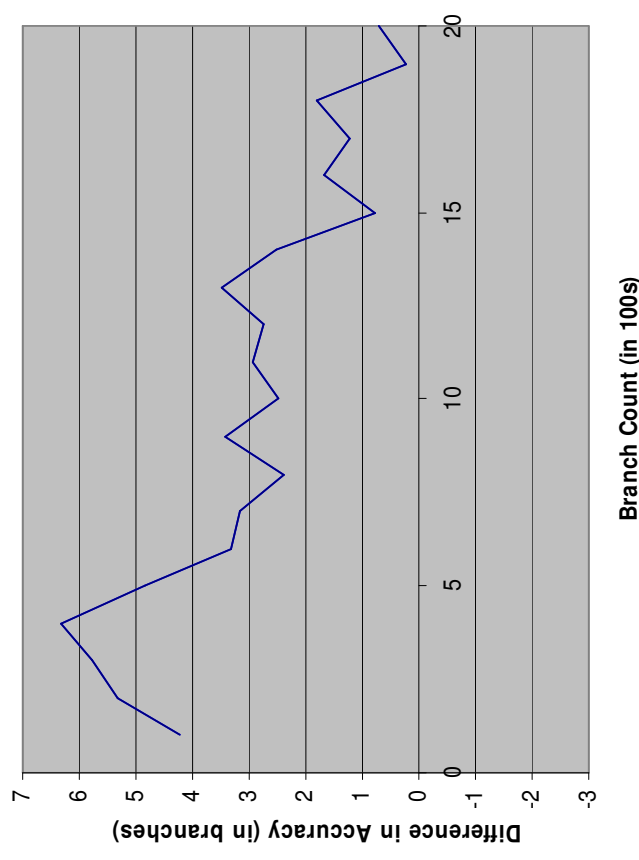
How Long Does Convergence Take?

- Longer convergence time leads to greater potential performance improvement

Performance Comparison of Branch Predictors for KDE trace by Process

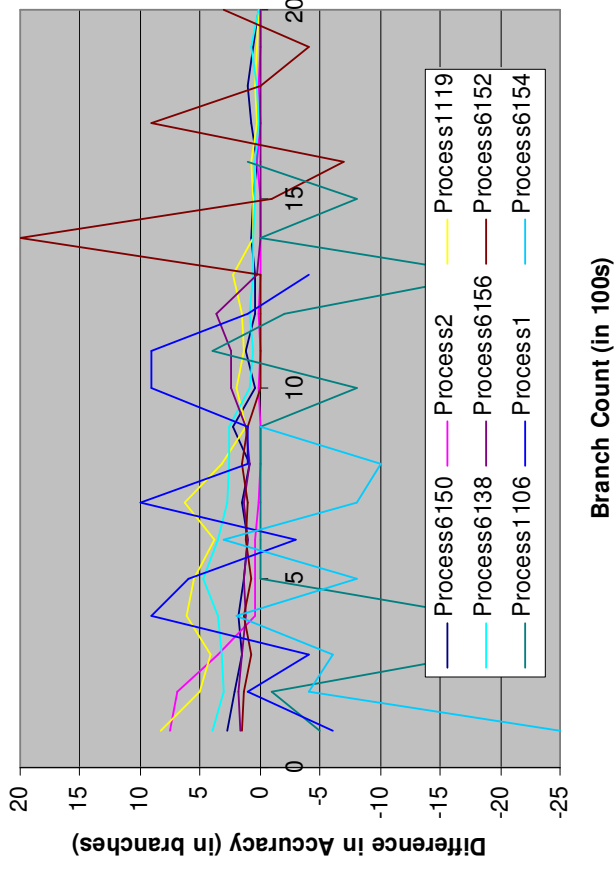


Average Performance Comparison of Branch Predictors for KDE trace

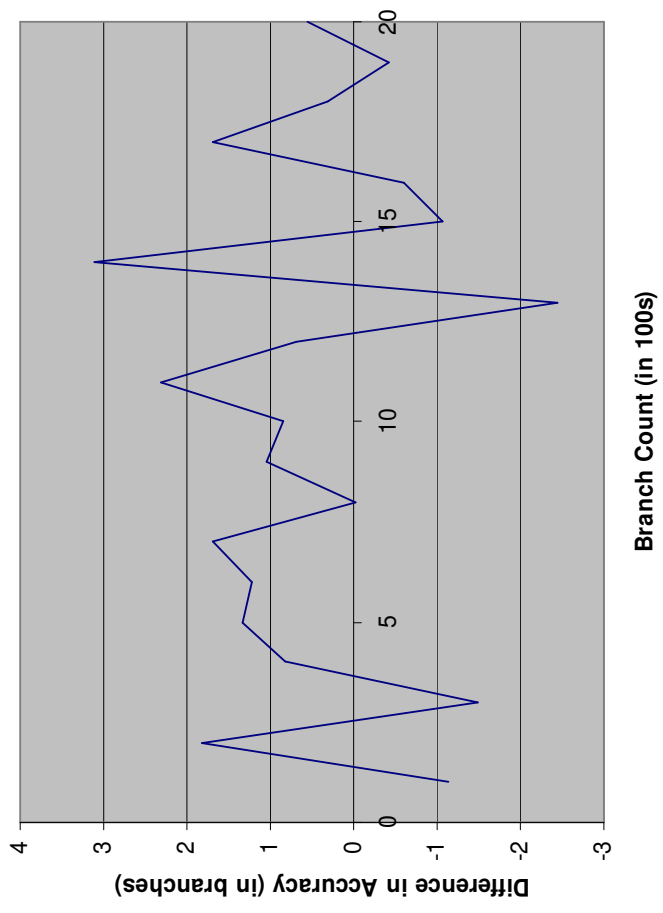


Not All Traces Converge

Performance Comparison of Branch Predictors for VI trace by process

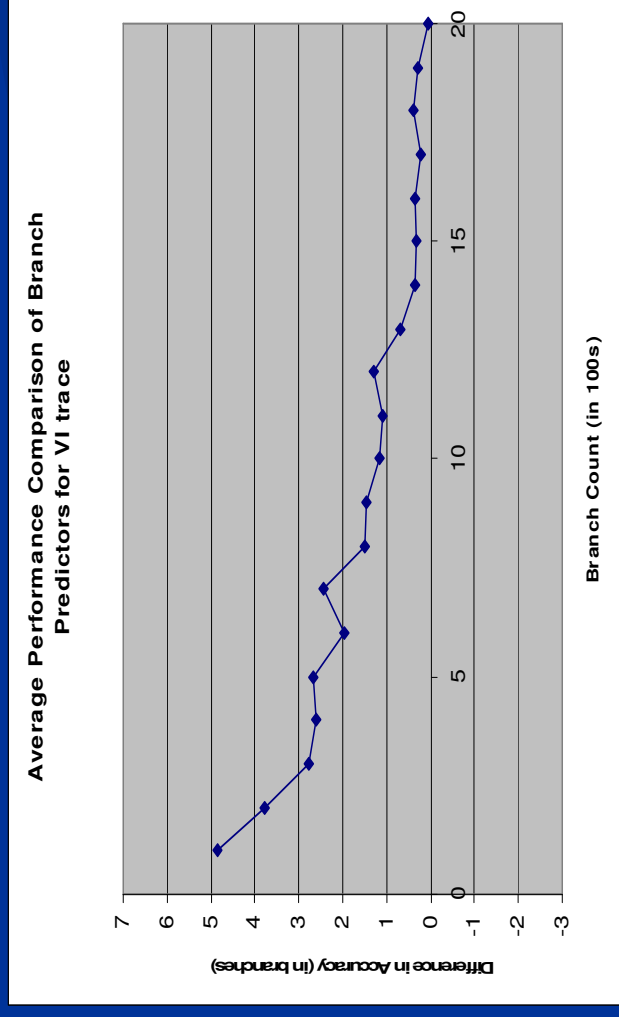


Average Performance Comparison of Branch Predictors for VI trace



Optimized View

- Some processes have poor branch predictor performance and dominate the averaged view
- Eliminating those processes shows convergence more clearly



Conclusions

- Modern machines do frequent process switches
- Traditional global branch prediction accuracy suffers after process switches
- Convergence period is short (~2000 branches)
- Overall accuracy improvement is dependent on frequency of task switches
- All accuracy improvements are on top of the existing branch predictor accuracy