Towards Adaptive Resource Allocation for Database Workloads

Cong Guo and Martin Karsten
David R. Cheriton School of Computer Science

UNIVERSITY OF WATERLOO

OUTLINE

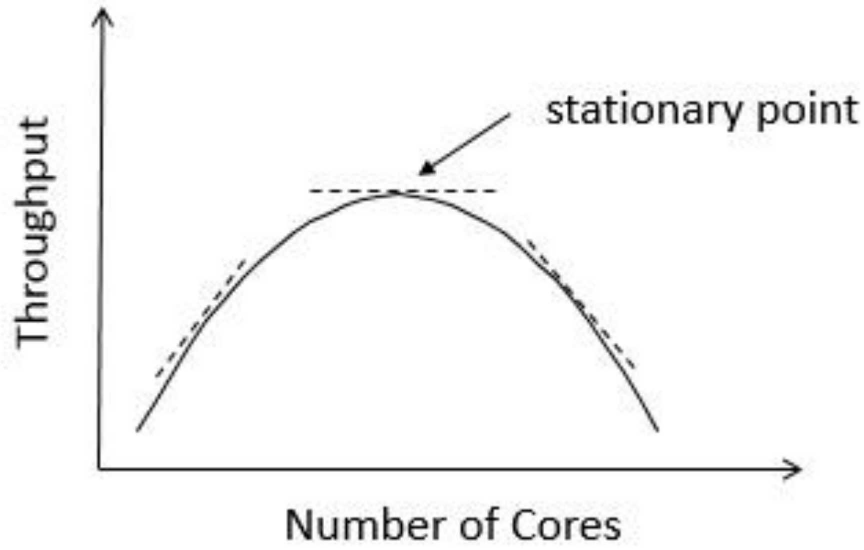
- Motivation
- Background
- Performance Measurement
- Controller Design
- Evaluation
- Conclusion

MOTIVATION

- Adaptive resource allocation
  - » more resources ≠ better performance
  - » improve resource utilization
- Feedback-based control
  - » used for optimization rather than regulation
  - » a fine-grained metric for online measurement

BACKGROUND

- Study case: CPU cores allocation
- Model: a concave performance curve
- Problem: search for the stationary point

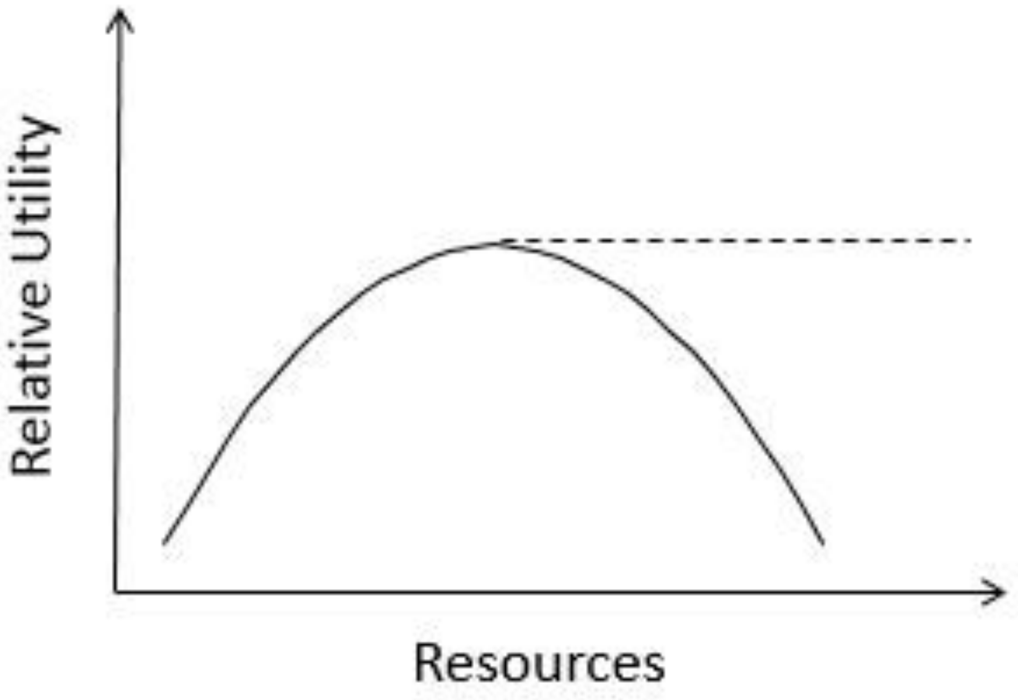


UNIVERSITY OF WATERLOO

BACKGROUND

- Fuzzy control for optimization problems
  - » no specific system model required
  - » no reference input required
  - » handle uncertainties and disturbances
  - » incorporate human knowledge via qualitative rules

GENERALIZATION



PERFORMANCE MEASUREMENT

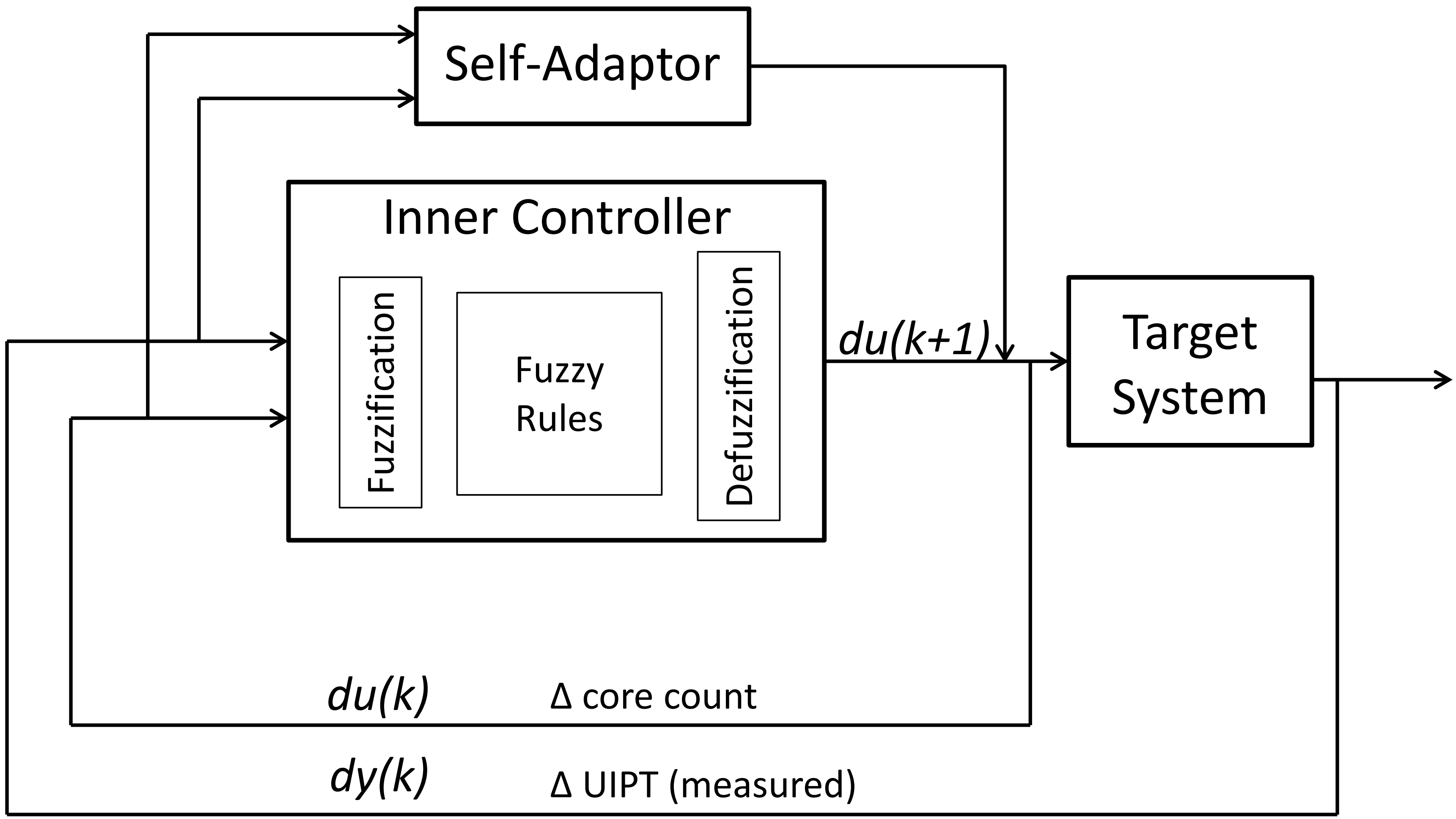
- Online performance measurement
  - » various application-specific metrics
  - » no one suitable for long-running analytical workloads
  - » User-level Instruction Per Time

USER-LEVEL INSTRUCTION PER TIME

- User-level instructions
  - » estimate the number of productive instructions
  - » measured by hardware performance counters during runtime
- Wall-clock time instead of CPU cycles
  - » CPU frequencies may change
  - » idle cycles are not covered
  - » total or average cycles?

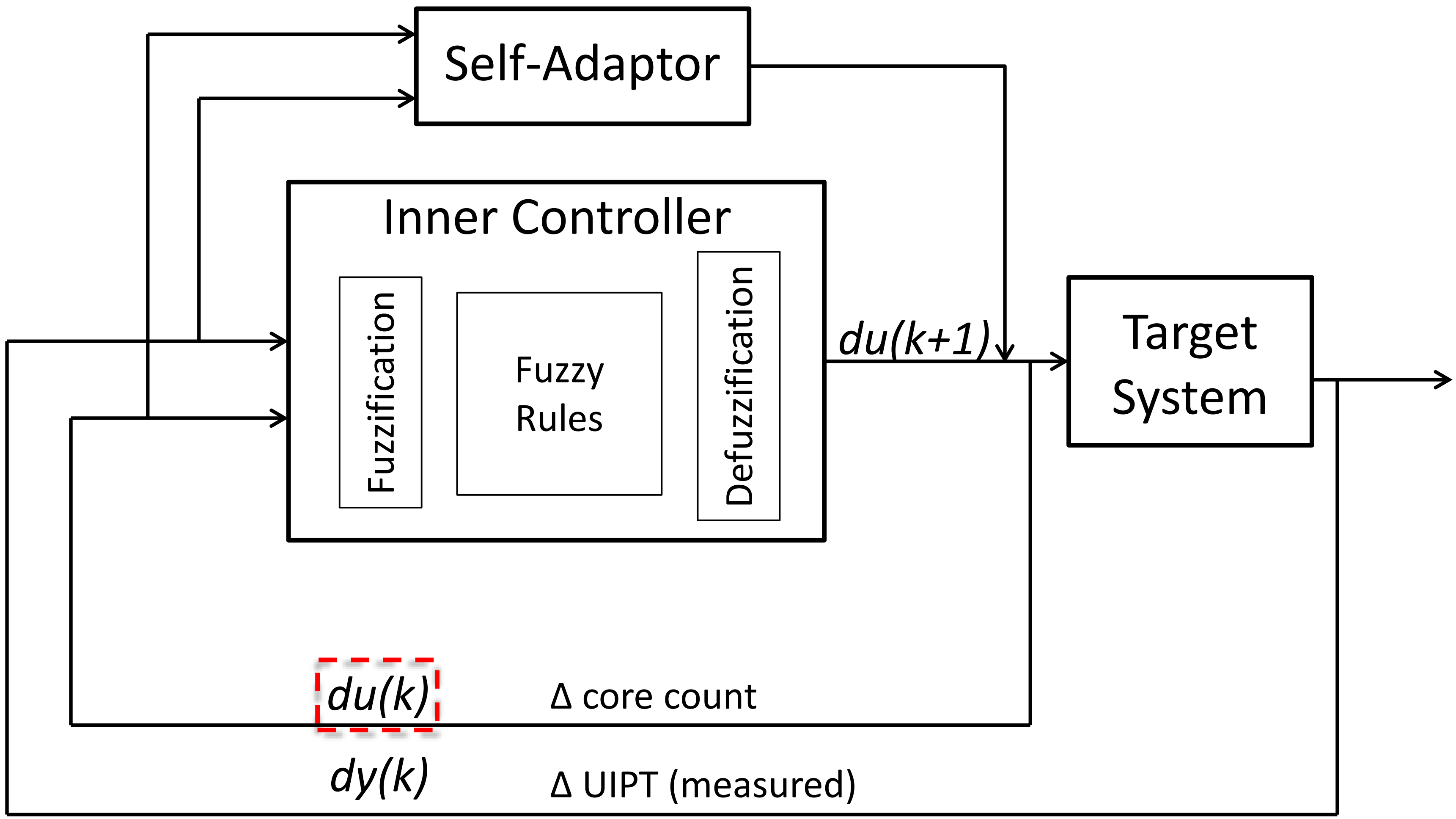
CONTROLLER DESIGN

- Overview



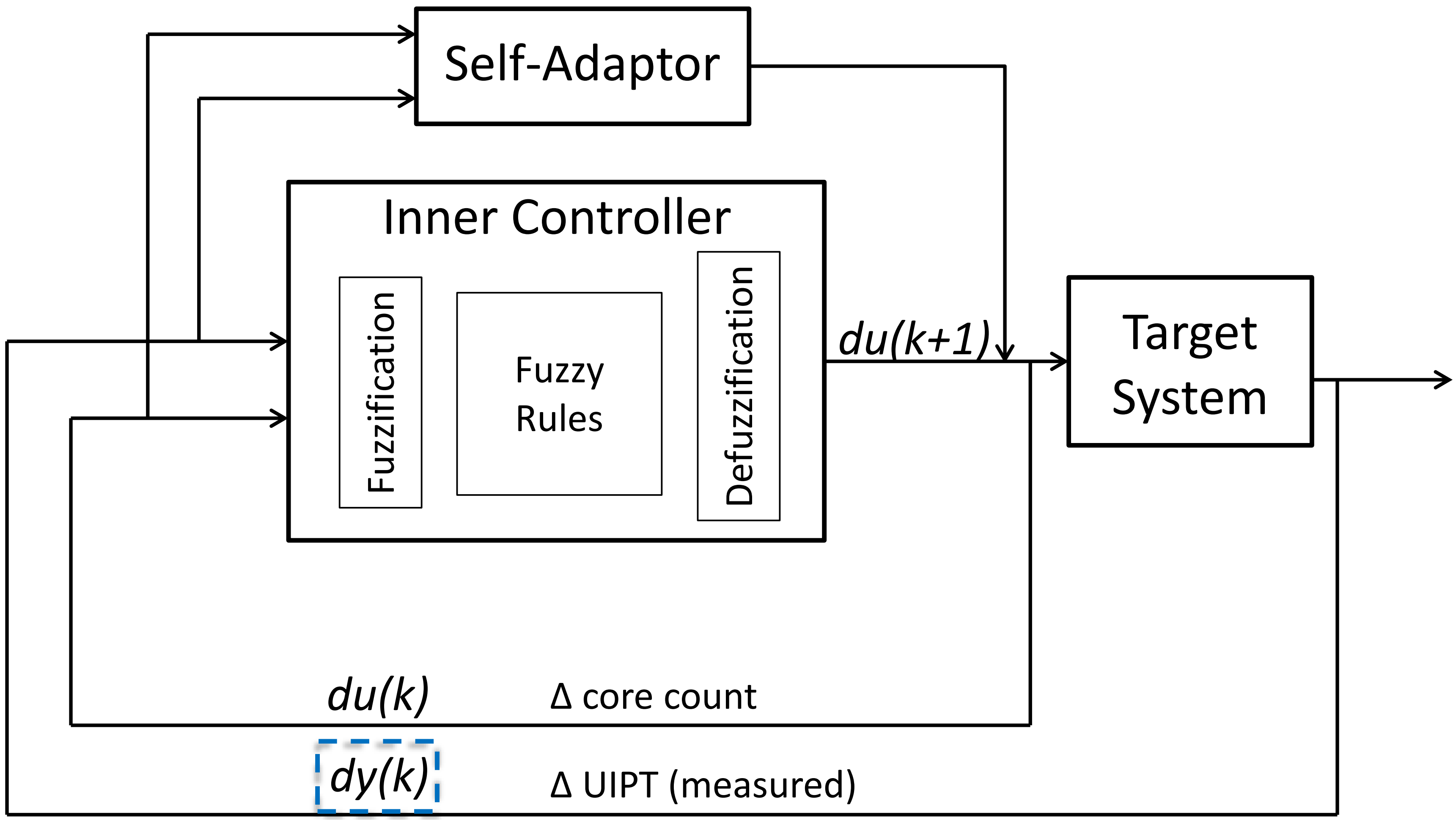
CONTROLLER DESIGN

- Overview



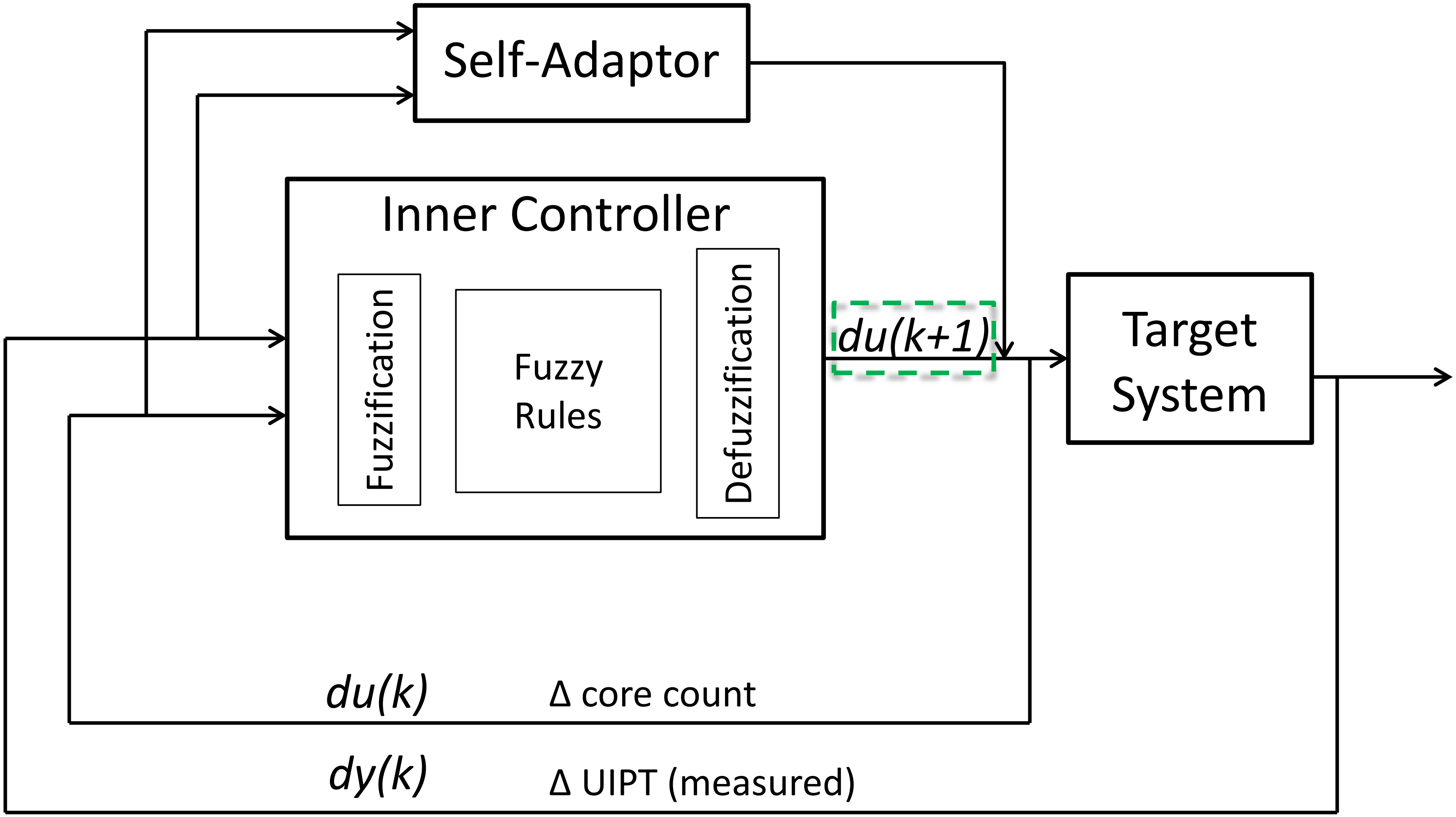
CONTROLLER DESIGN

- Overview



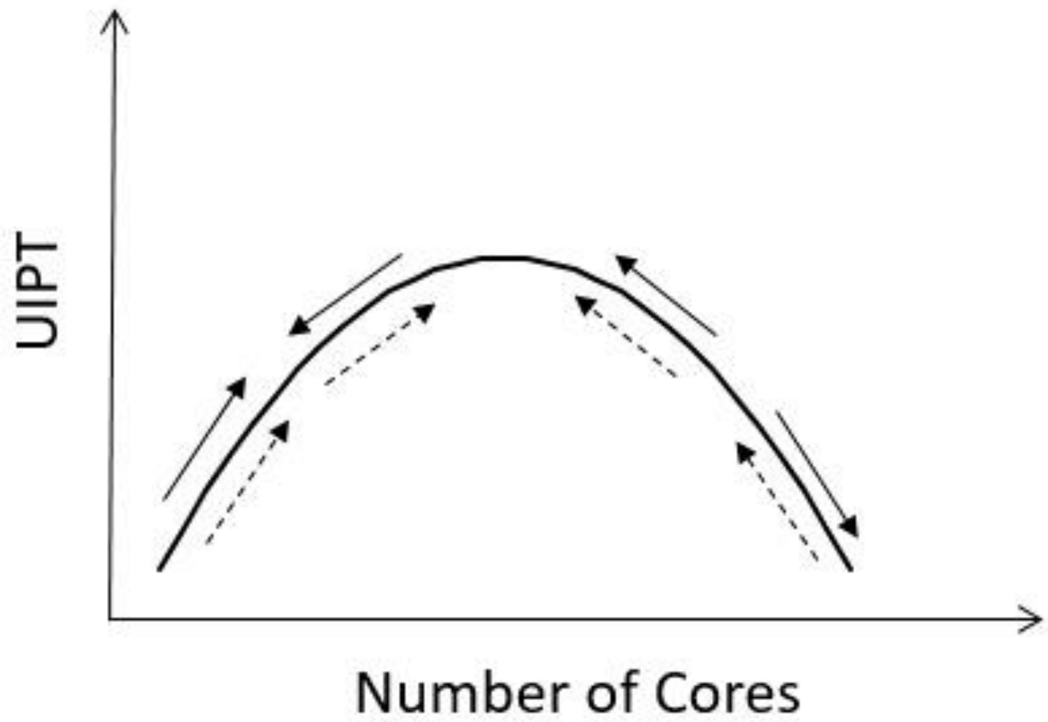
CONTROLLER DESIGN

- Overview



INNER FUZZY CONTROLLER

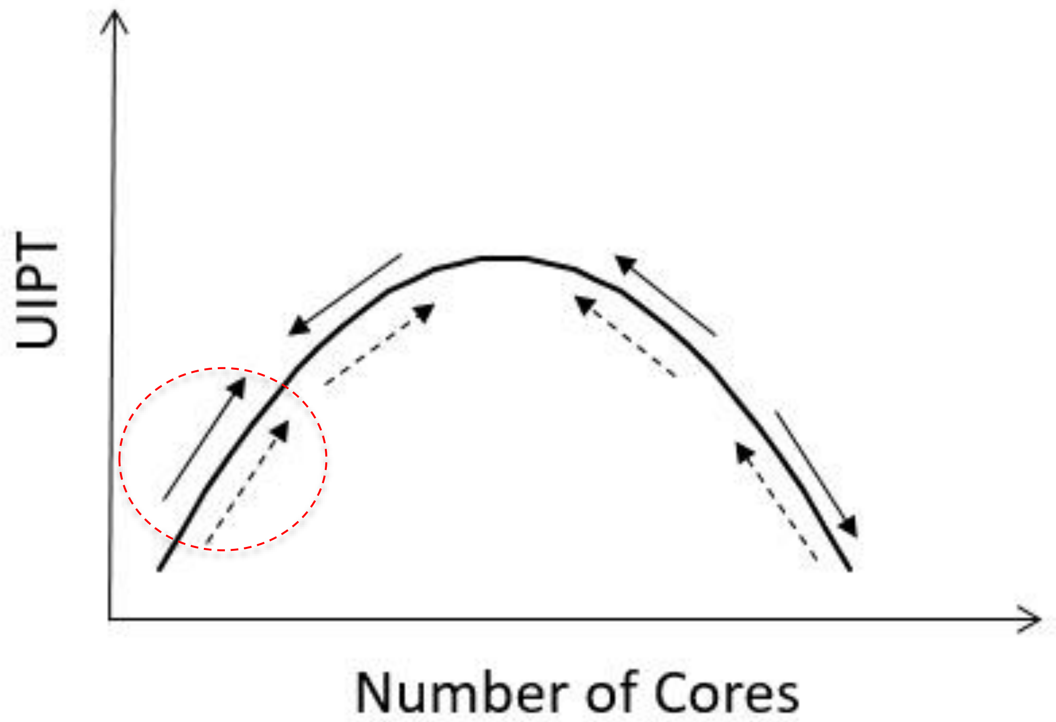
- Basic Control Rules



UNIVERSITY OF WATERLOO

INNER FUZZY CONTROLLER

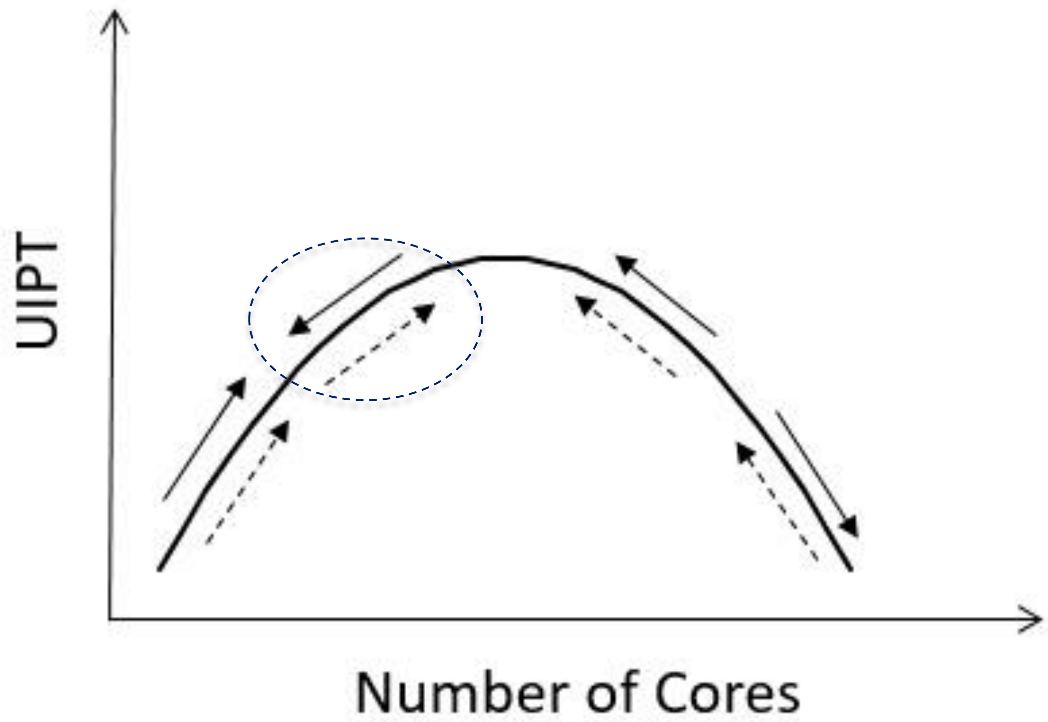
- Basic Control Rules



UNIVERSITY OF WATERLOO

INNER FUZZY CONTROLLER

- Basic Control Rules



SELF-ADAPTOR

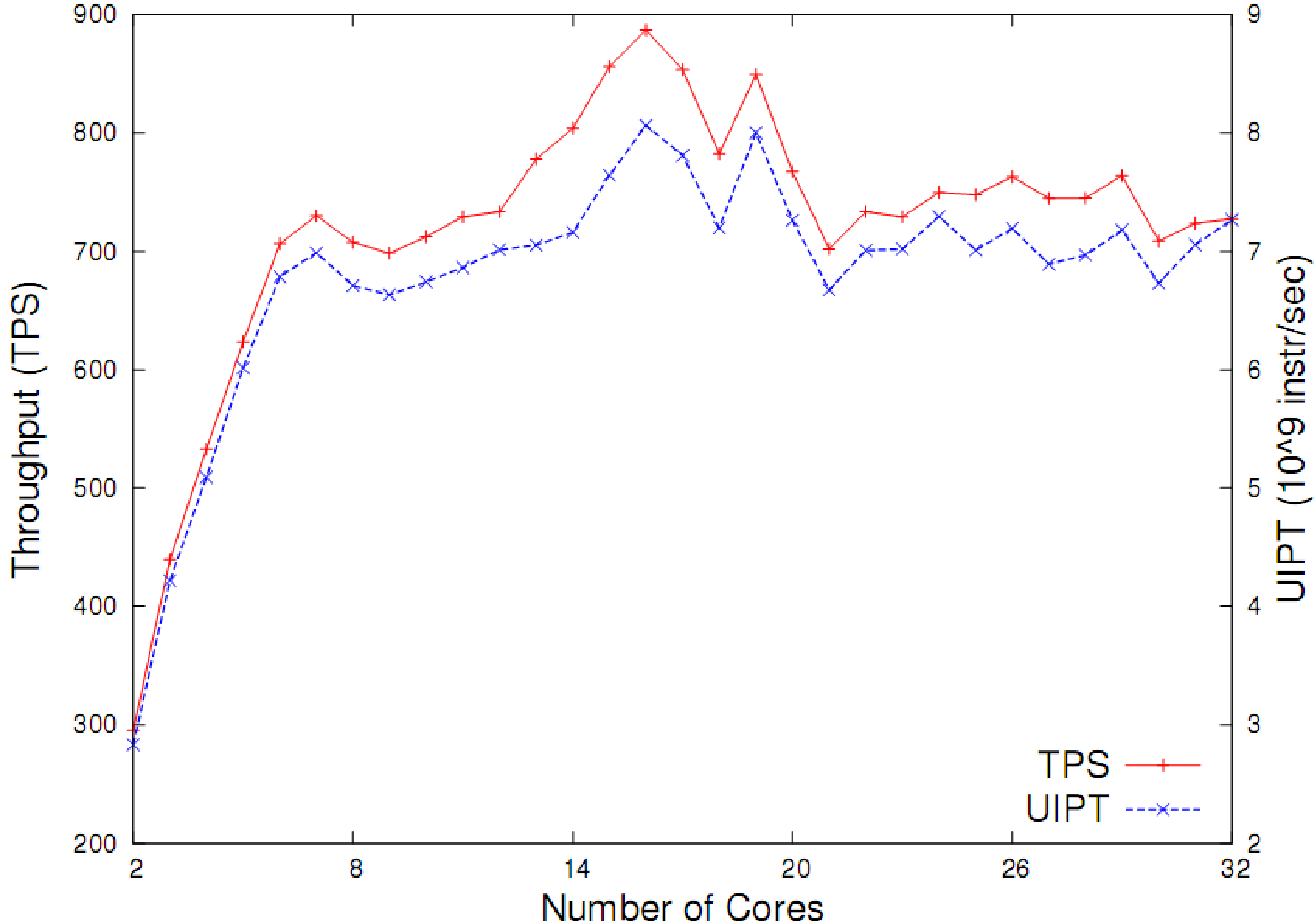
- Detect workload changes
  - » observe UIPT changes for a window of time
  - » reduce cores if UIPT decreases more than a threshold
  - » detect workload increment via a probing allocation
- Find the minimum allocation

EVALUATION

- Resource allocation
  - » Linux control groups
  - » no assignment of queries to cores explicitly
- CPU-bound workload
  - » sufficient buffer pool size
  - » disable synchronous logging

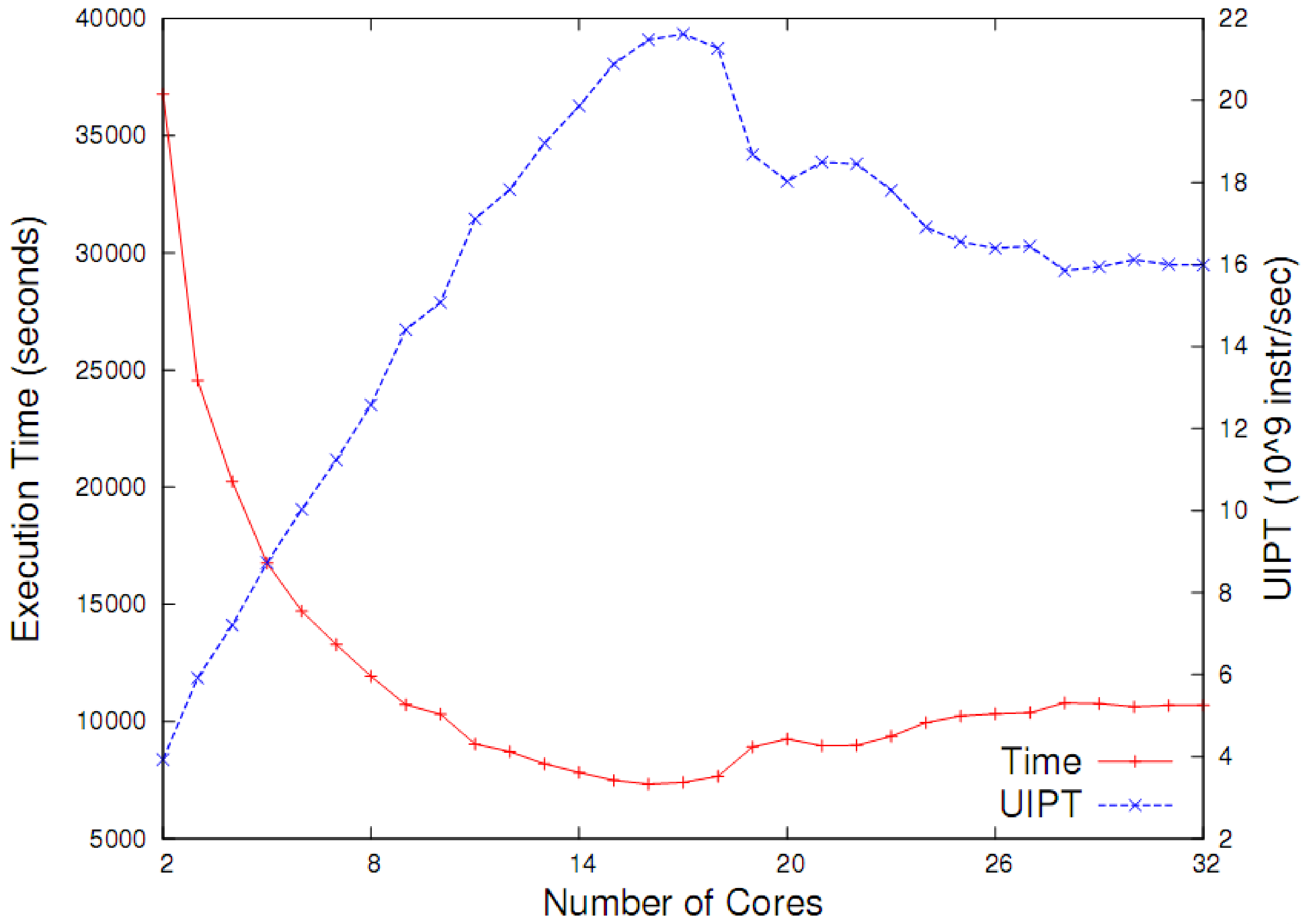
EVALUATION - UIPT

- Database Workloads – TPC-E



EVALUATION - UIPT

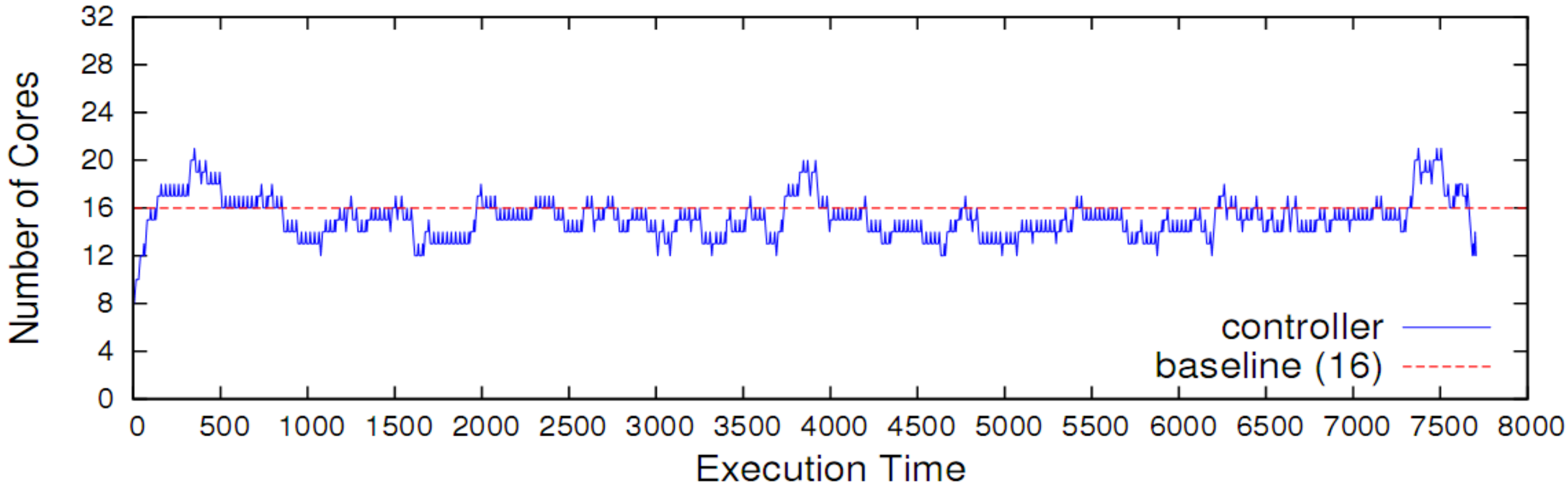
- Database Workloads – TPC-H



UNIVERSITY OF WATERLOO

EVALUATION – CONTROLLER

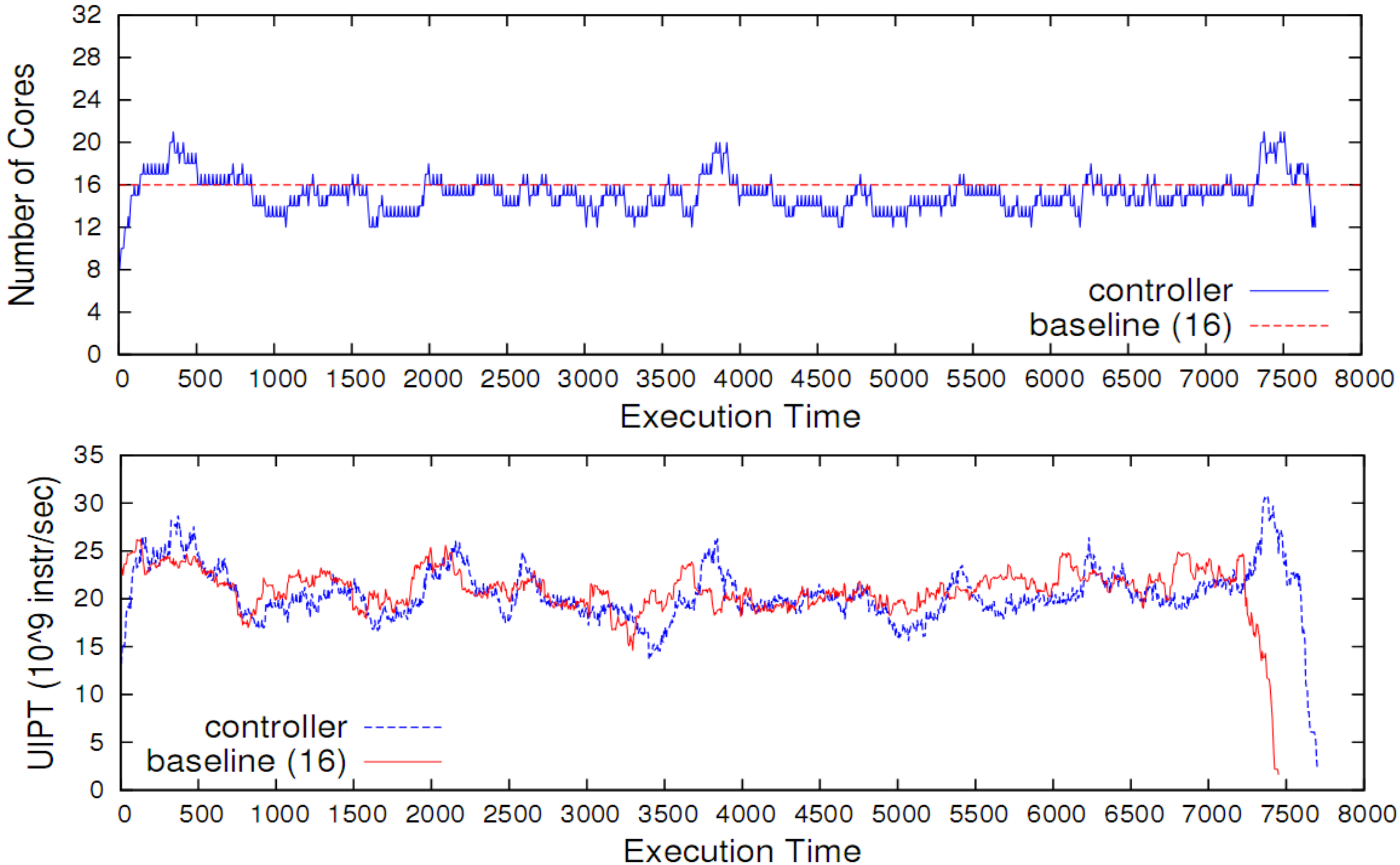
- TPC-H Workload – start from 8 cores



UNIVERSITY OF WATERLOO

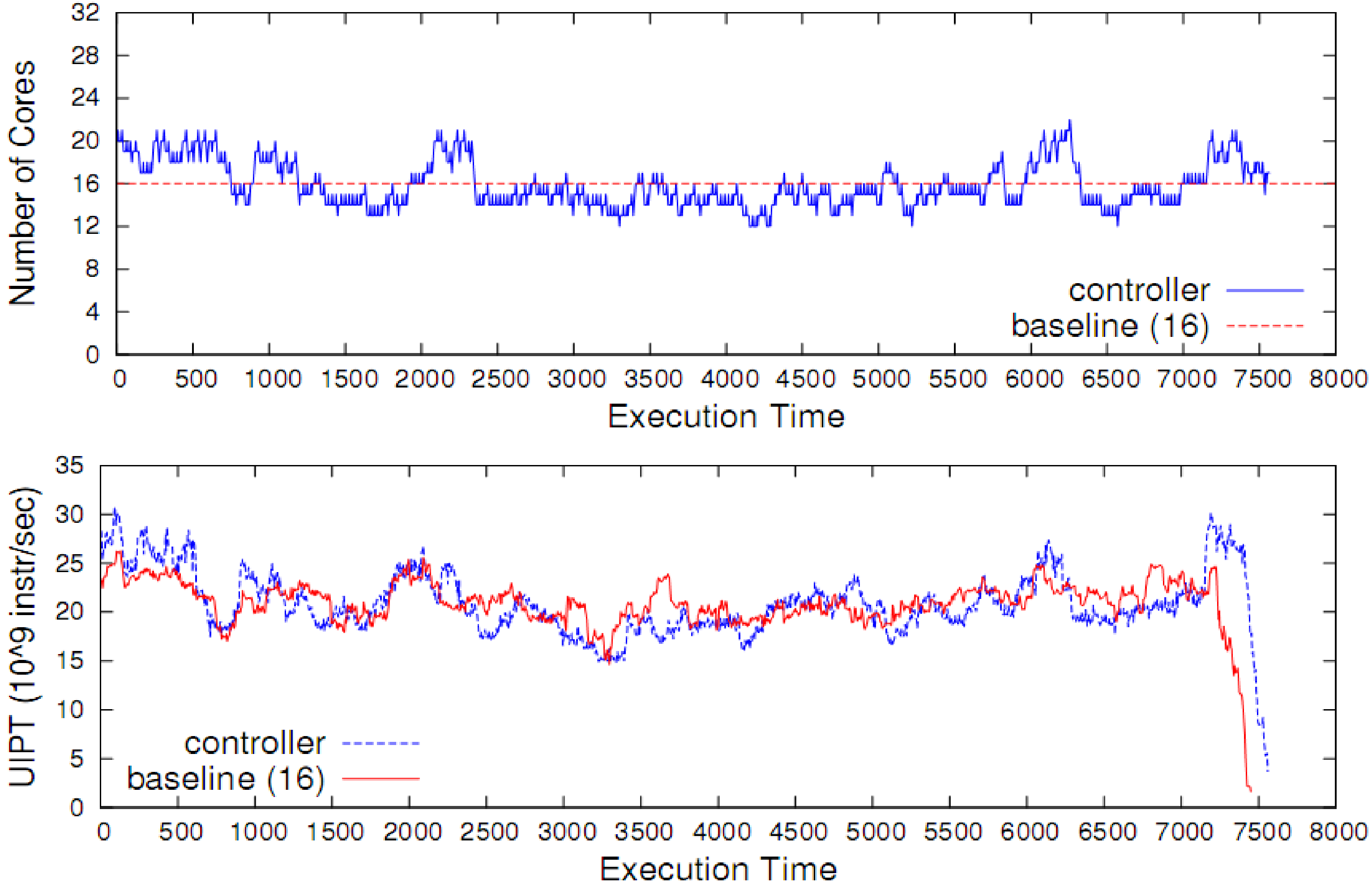
EVALUATION – CONTROLLER

- TPC-H Workload – start from 8 cores



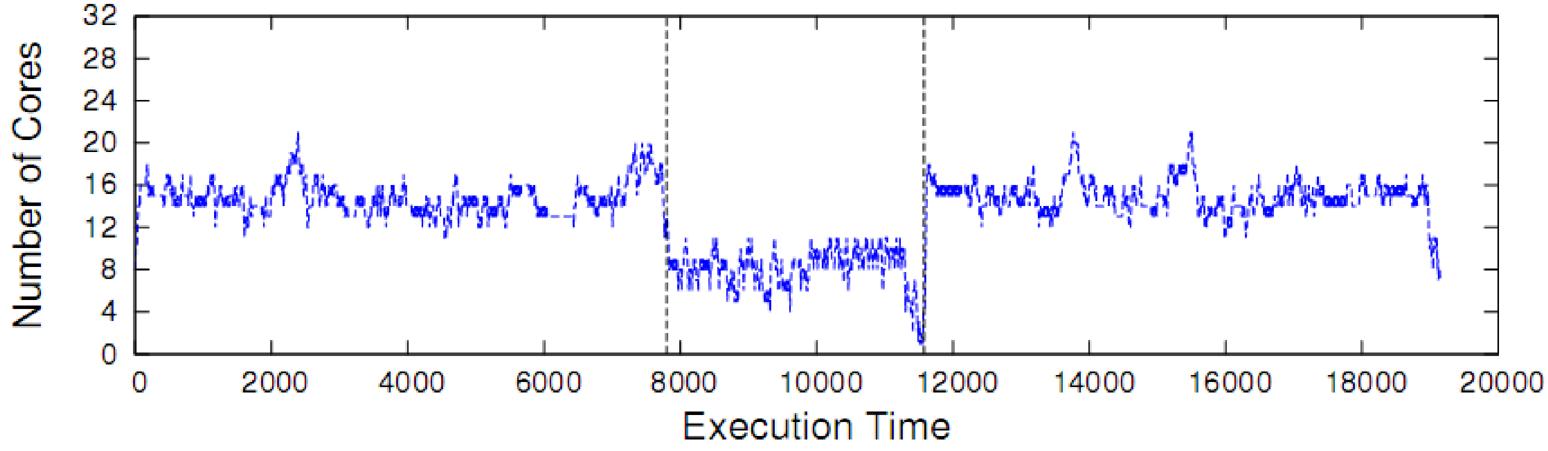
EVALUATION – CONTROLLER

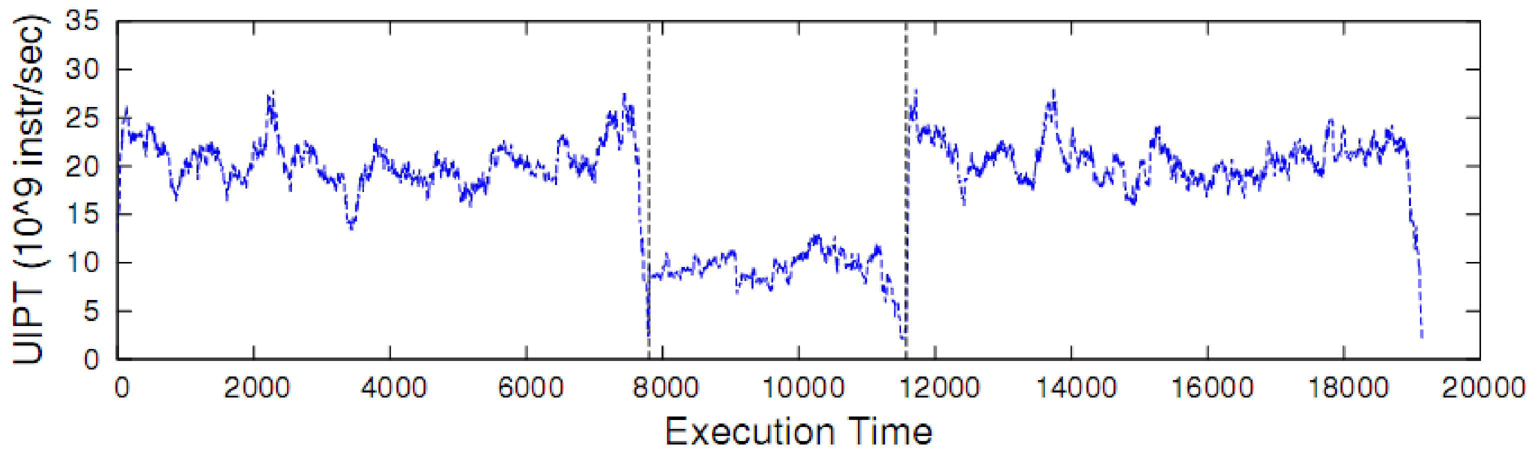
- TPC-H Workload – start from 20 cores



EVALUATION – CONTROLLER

- Adaptive to Changing TPC-H Workloads



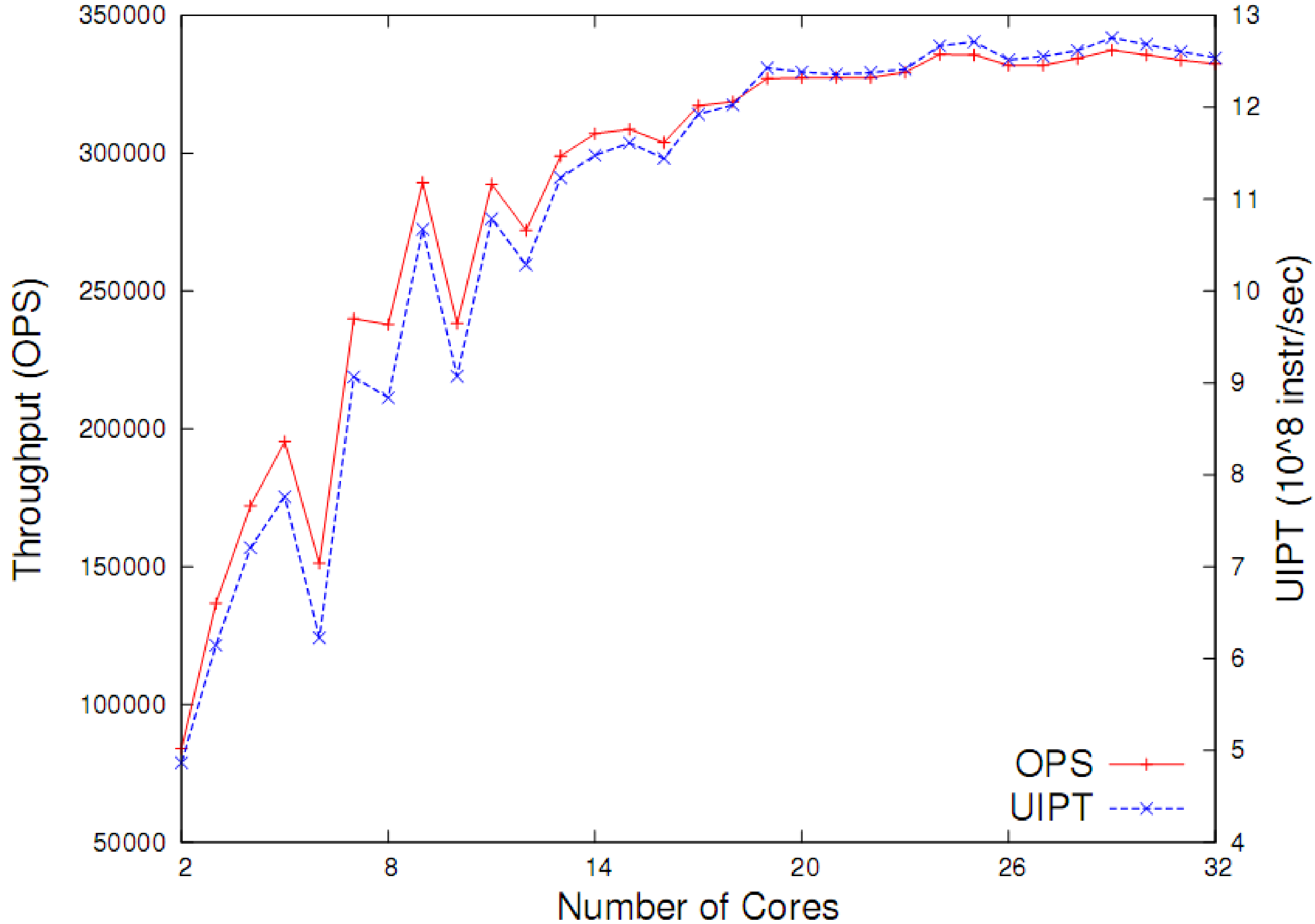


CONCLUSION

- A fine-grained performance metric
  - » agile feedback for long-running workloads
  - » a good proxy for application-level performance metrics
- An allocation algorithm based on fuzzy control
  - » no accurate system model required
  - » comparable performance
  - » further improvement
- Apply to other problems
  - » power efficiency
  - » software management problems

EVALUATION - UIPT

- Memcached



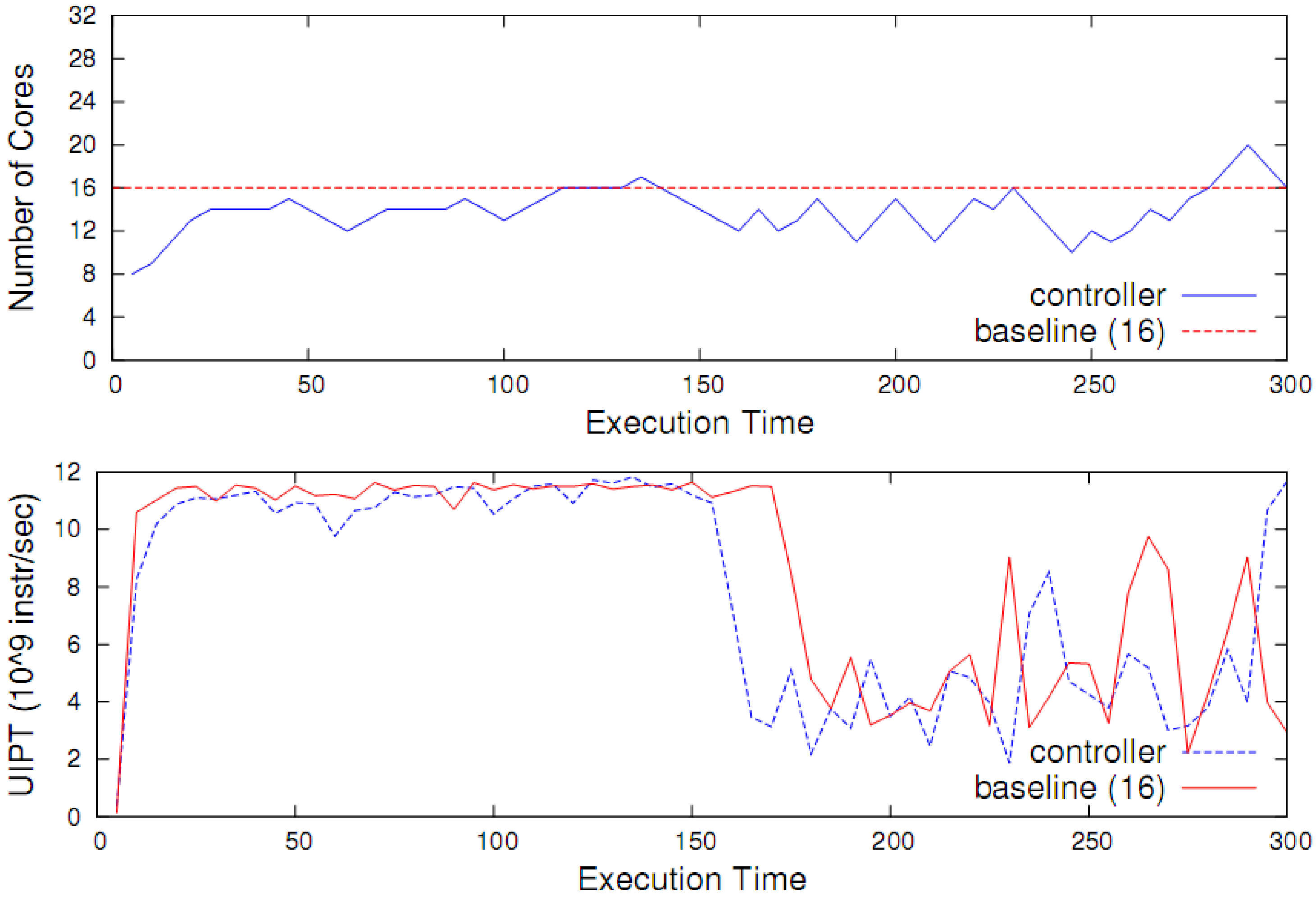
EVALUATION - UIPT

- PARSEC Benchmark

Program	Correlation between Time and UIPT	Correlation between Relative Changes
blackscholes	-0.9273	-0.9946
bodytrack	-0.9531	-0.9939
canneal	-0.9764	-0.9886
dedup	-0.9944	-0.9987
facesim	-0.9927	-0.9919
ferret	-0.8073	-0.9931
fluidanimate	-0.8994	-0.9907
freqmine	-0.8425	-0.9873
raytrace	-0.9664	-0.9979
streamcluster	-0.9604	-0.9875
swaptions	-0.9945	-0.9910
vips	-0.8635	-0.9911
x264	-0.8924	-0.9891

EVALUATION – CONTROLLER

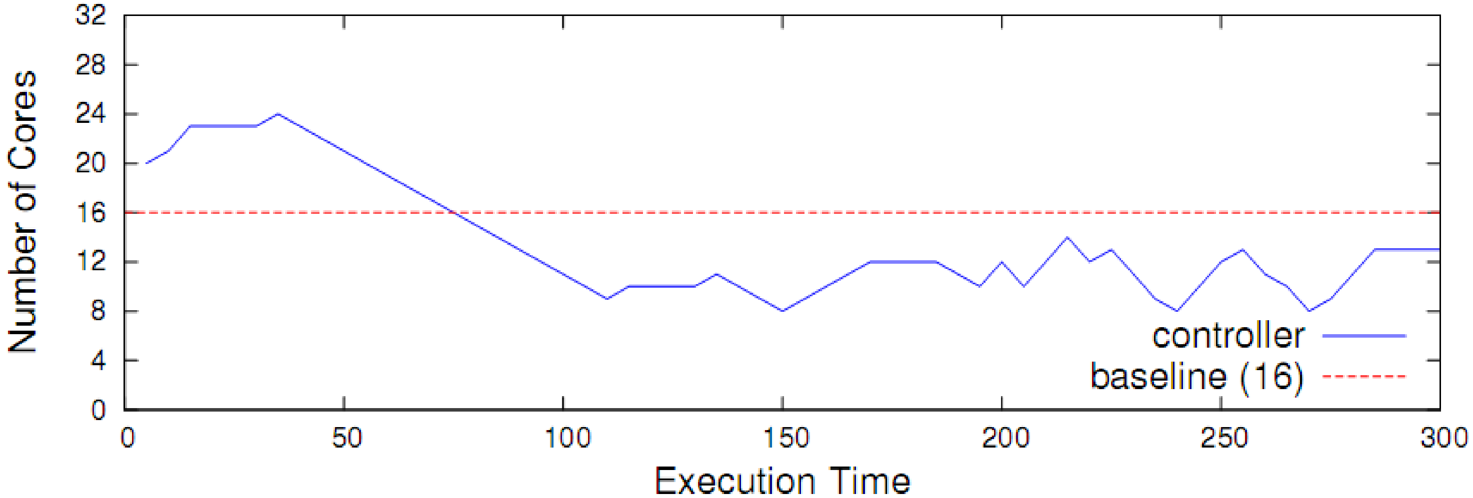
- TPC-E Workloads – start from 8 cores

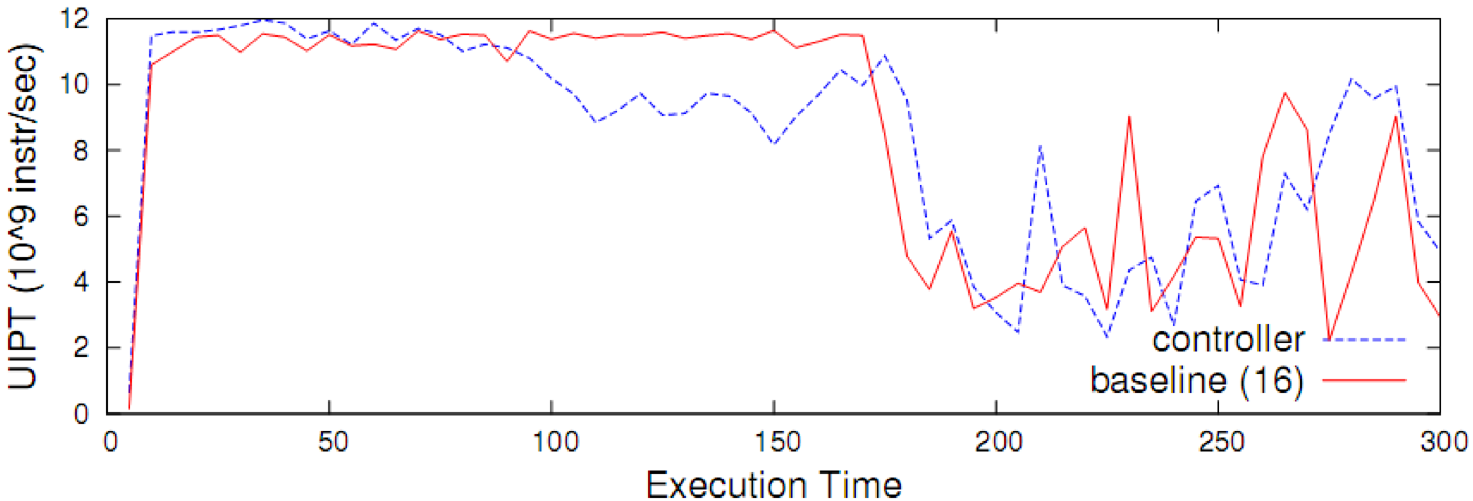


UNIVERSITY OF WATERLOO

EVALUATION – CONTROLLER

- TPC-E Workloads – start from 20 cores





EVALUATION – CONTROLLER

- Adaptive to Changing TPC-E Workloads