

3η ΑΣΚΗΣΗ ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ
Ακ. έτος 2018-2019, 8ο Εξάμηνο, Σχολή ΗΜ&ΜΥ



Όνομ/νυμο: Χαρδούβελης Γεώργιος-Ορέστης
Α.Μ: 03115100
Εξάμηνο: 8ο

Στόχος της άσκησης είναι η μελέτη των χαρακτηριστικών των σύγχρονων superscalar, out-of-order επεξεργαστών και ο τρόπος με τον οποίο επηρεάζουν την απόδοση του συστήματος, την κατανάλωση ενέργειας καθώς και το μέγεθος του chip του επεξεργαστή.

PIN

Στα πλαίσια της παρούσας άσκησης χρησιμοποιήσαμε τα εργαλείο “PIN” (όπως και στις προηγούμενες ασκήσεις) Χρησιμοποιώντας το PIN μπορούμε και εισάγουμε κώδικας κατά τη διάρκεια της εκτέλεσης των μετροπρογραμμάτων ανάμεσα στις εντολές της εφαρμογής αυτής και συλλέγουμε την επιθυμητές πληροφορίες που αφορούν την εκτέλεση τους.

Ο προσομοιωτής που θα αξιοποιήσει το εργαλείο PIN είναι ο Sniper Multicore Simulator.

BENCHMARKS

Στα πλαίσια της παρούσας άσκησης χρησιμοποιήσαμε ορισμένα από τα SPEC_CPU2006. Συγκεκριμένα χρησιμοποιήσαμε τα παρακάτω 12:

- 403.gcc
- 429.mcf
- 434.zeusmp
- 436.cactusADM
- 445.gobmk
- 450.soplex
- 456.hmmer
- 458.sjeng
- 459.GemsFDTD
- 471.omnetpp
- 473.astar
- 483.xalancbmk

McPAT

Το McPAT (Multi-core Power, Area, Timing) είναι ένα εργαλείο που συμπεριλαμβάνεται στο Sniper και θα το χρησιμοποιήσουμε με στόχο να πάρουμε τα ζητούμενα στατιστικά από τις προσομοιώσεις. Ουσιαστικά μοντελοποιεί τα χαρακτηριστικά ενός επεξεργαστή, όπως η κατανάλωση ενέργειας και το μέγεθος που καταλαμβάνουν στο τσιπ οι διαφορετικές δομικές μονάδες του επεξεργαστή

9. Πειραματική αξιολόγηση

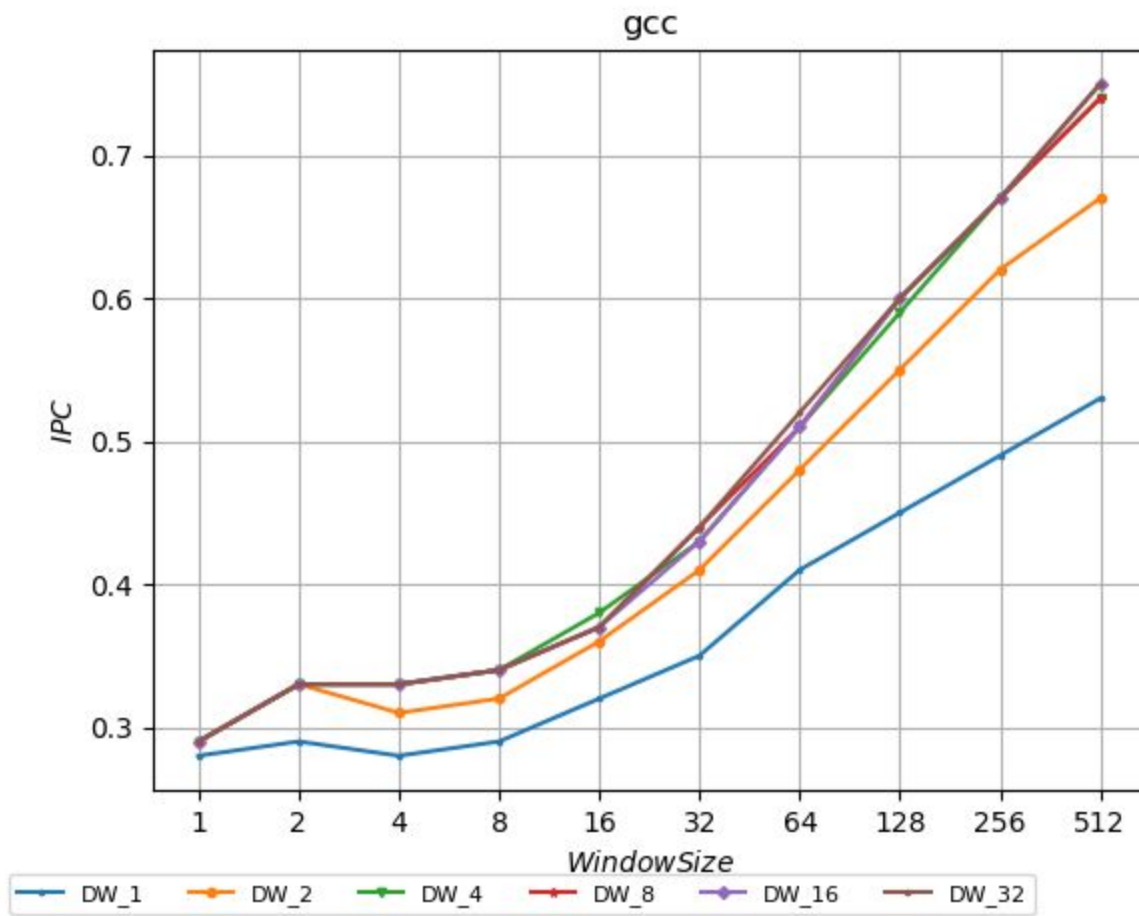
i) Χρειάζεται πραγματικά να προσομοιώσετε και τους 60 διαφορετικούς επεξεργαστές που προκύπτουν με βάση τις παραπάνω τιμές; Αν όχι, εξηγήστε ποιές περιπτώσεις μπορείτε να παραλείψετε και γιατί. Δικαιολογήστε την απάντησή σας όχι μόνο θεωρητικά αλλά και αποδεικνύοντας για ένα μικρό αριθμό αυτών των περιπτώσεων ότι καλώς τις παραλείψατε.

Συνολικά για κάθε μετροπρόγραμμα προκύπτουν 60 προσομοιώσεις. Παρόλα αυτά, δεν χρειάζεται πραγματικά να προσομοιωθούν όλοι οι συνδυασμοί. Συγκεκριμένα, οι επεξεργαστές που το window size είναι μικρότερο του dispatch width. Άρα οι απαραίτητες προσομοιώσεις θα ήταν 45 για κάθε μετροπρόγραμμα.

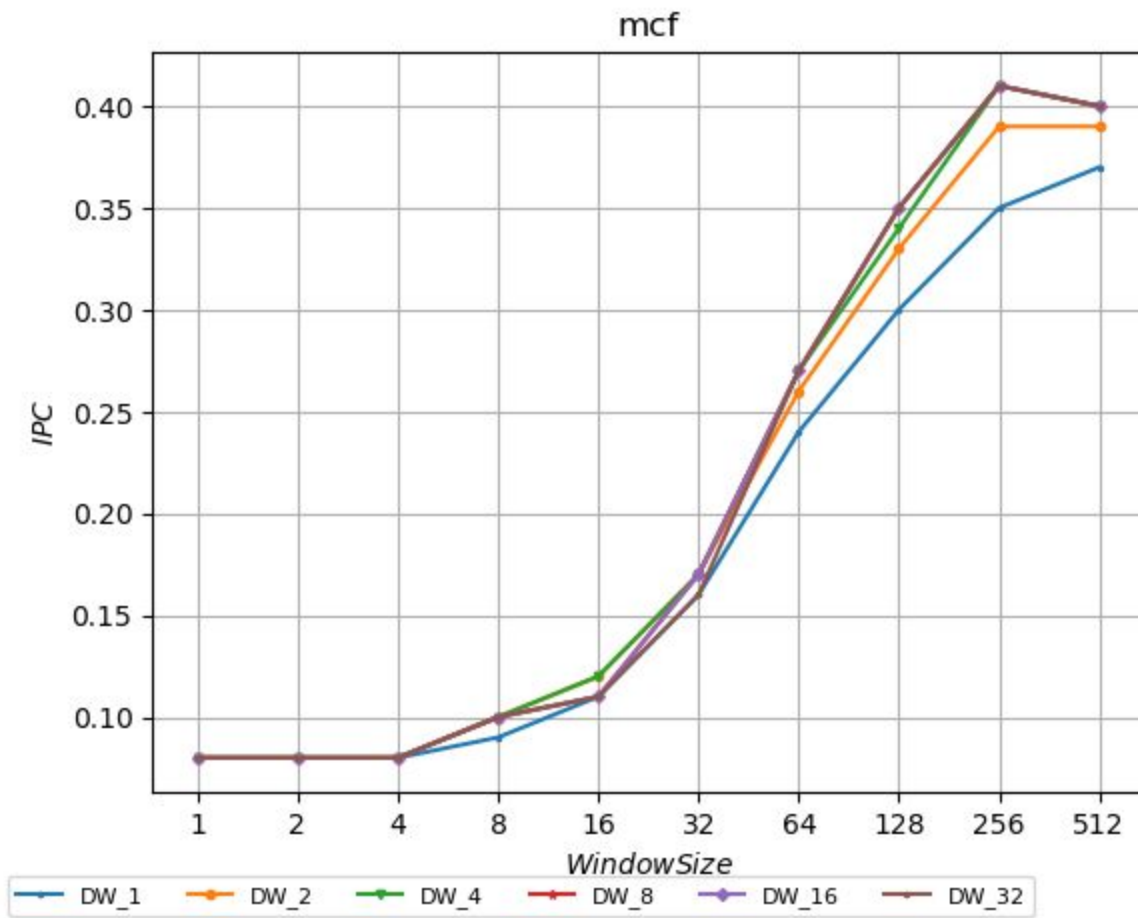
ii) Πώς επηρεάζει η κάθε παράμετρος την απόδοση του επεξεργαστή; Σε ποιά συμπεράσματα μπορείτε να καταλήξετε ως προς το σχεδιασμό ενός superscalar, out-of-order επεξεργαστή;

Για να απαντήσουμε στην ερώτηση, θα παρουσιάσουμε αρχικά για κάθε benchmark τα αποτελέσματα των προσομοιώσεων σε γραφικές παραστάσεις.

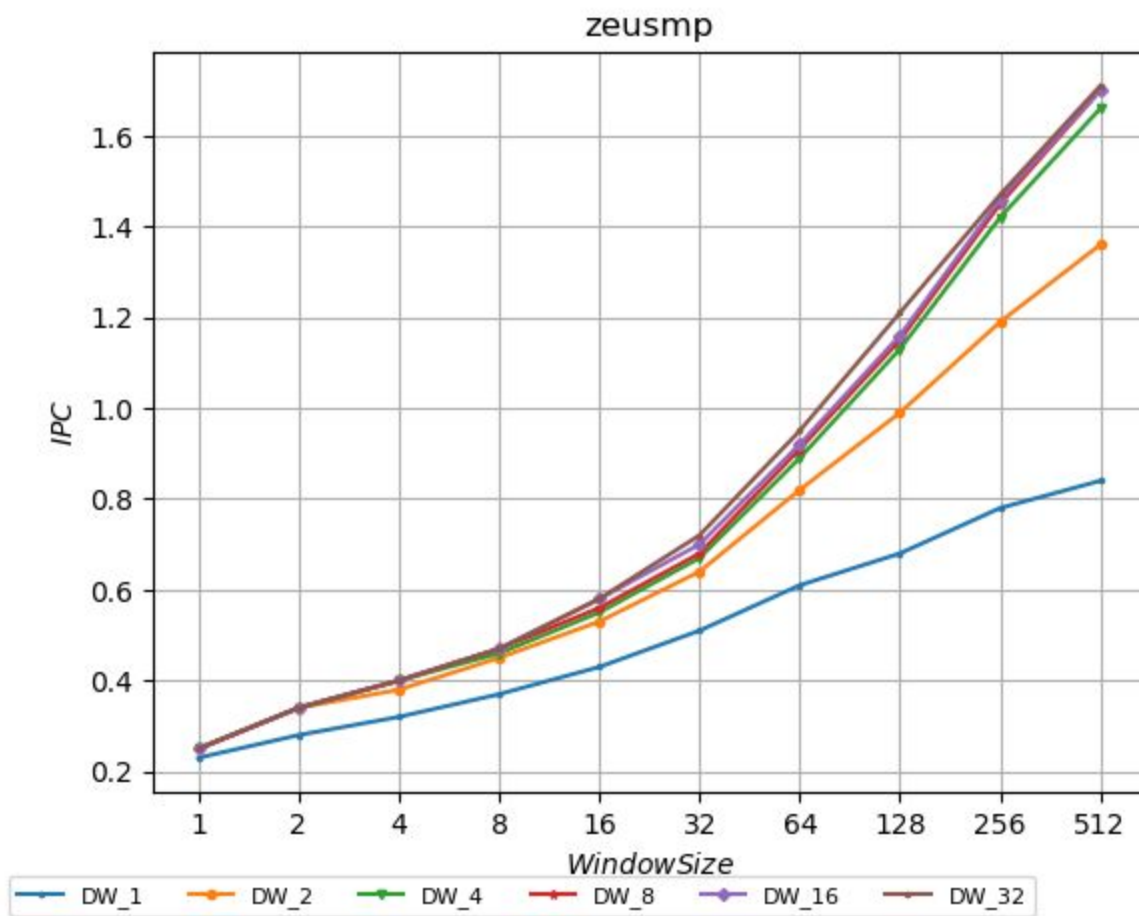
➤ 403.gcc



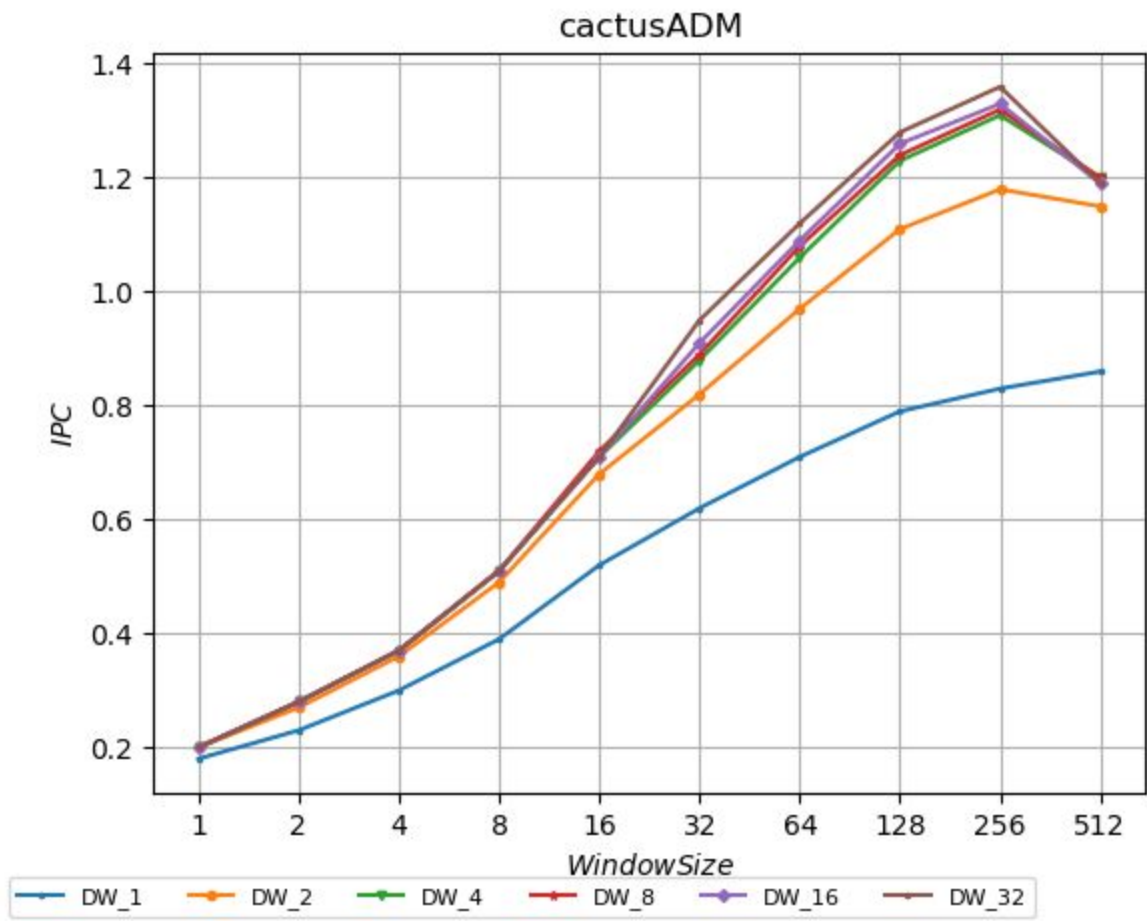
➤ 429.mcf



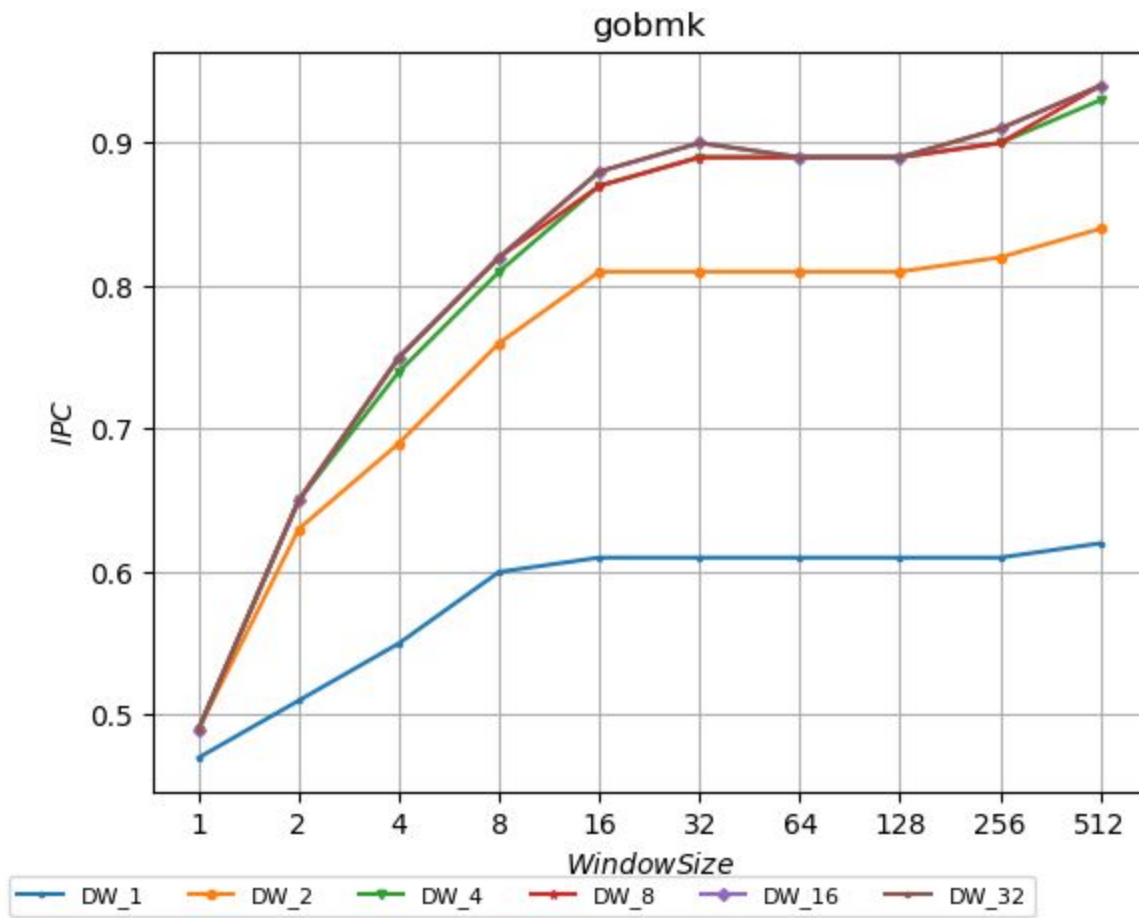
➤ 434.zeusmp



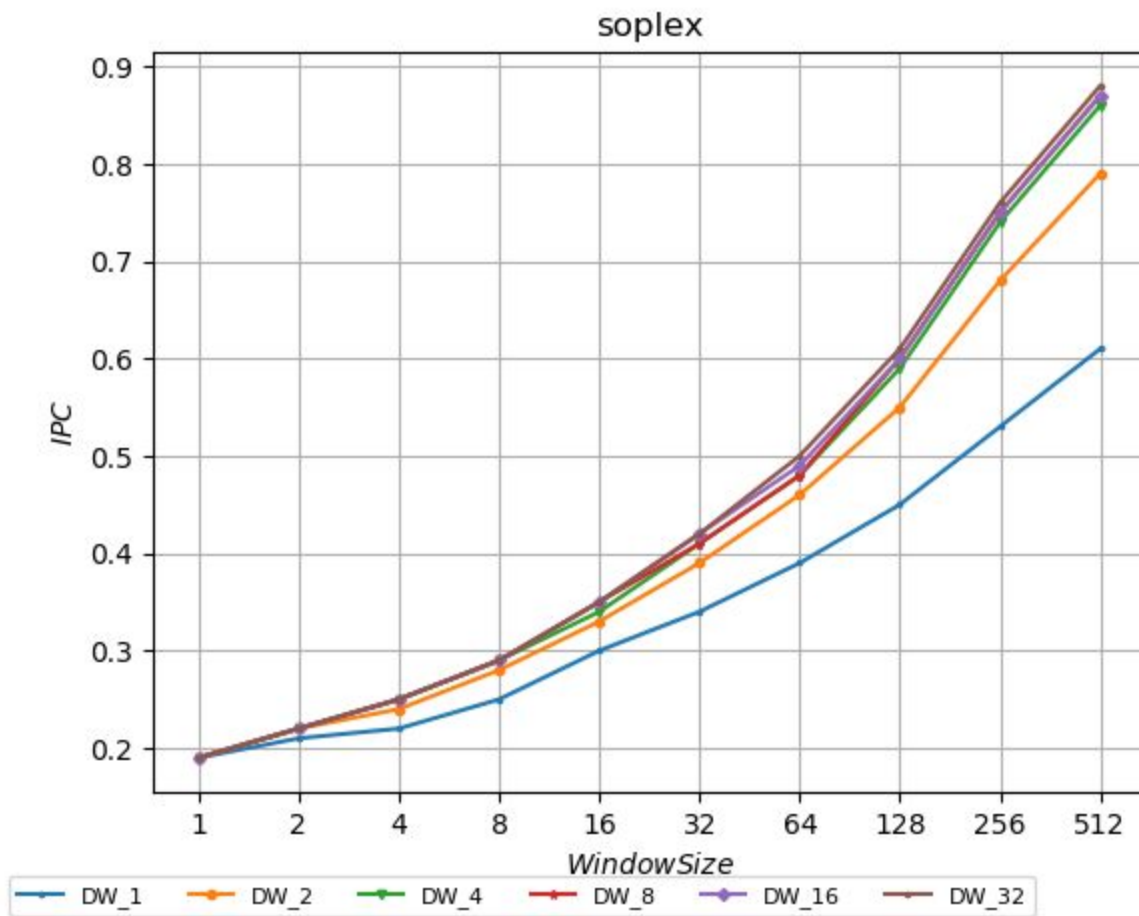
➤ 436.cactusADM



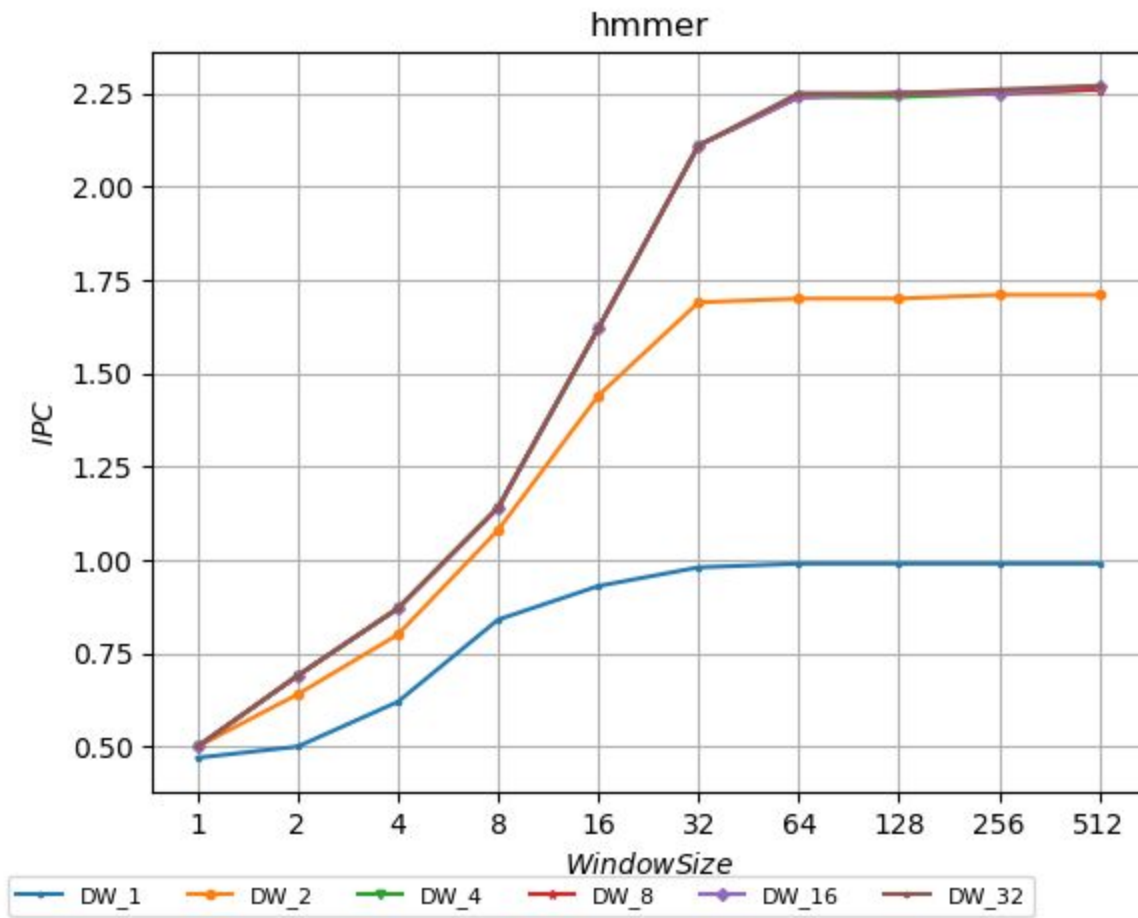
➤ 445.gobmk



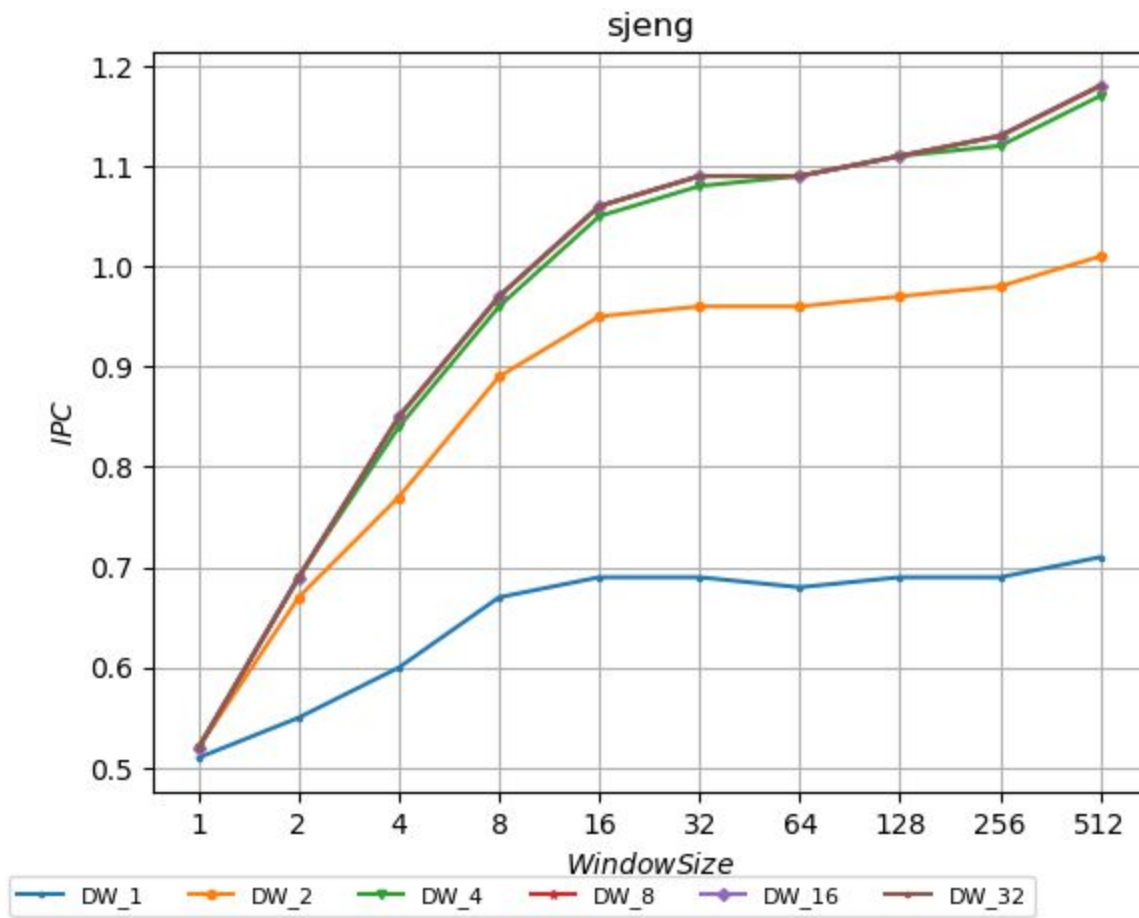
➤ 450.soplex



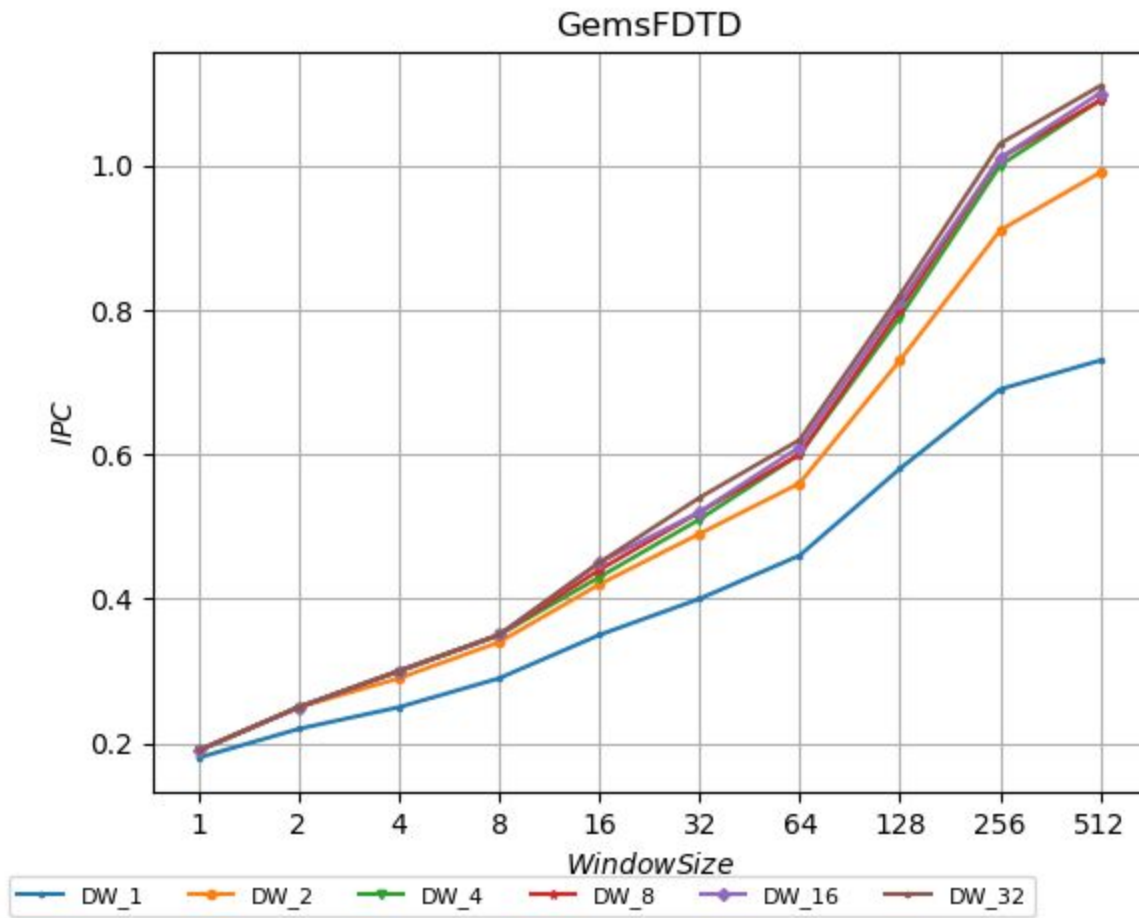
➤ 456.hmm



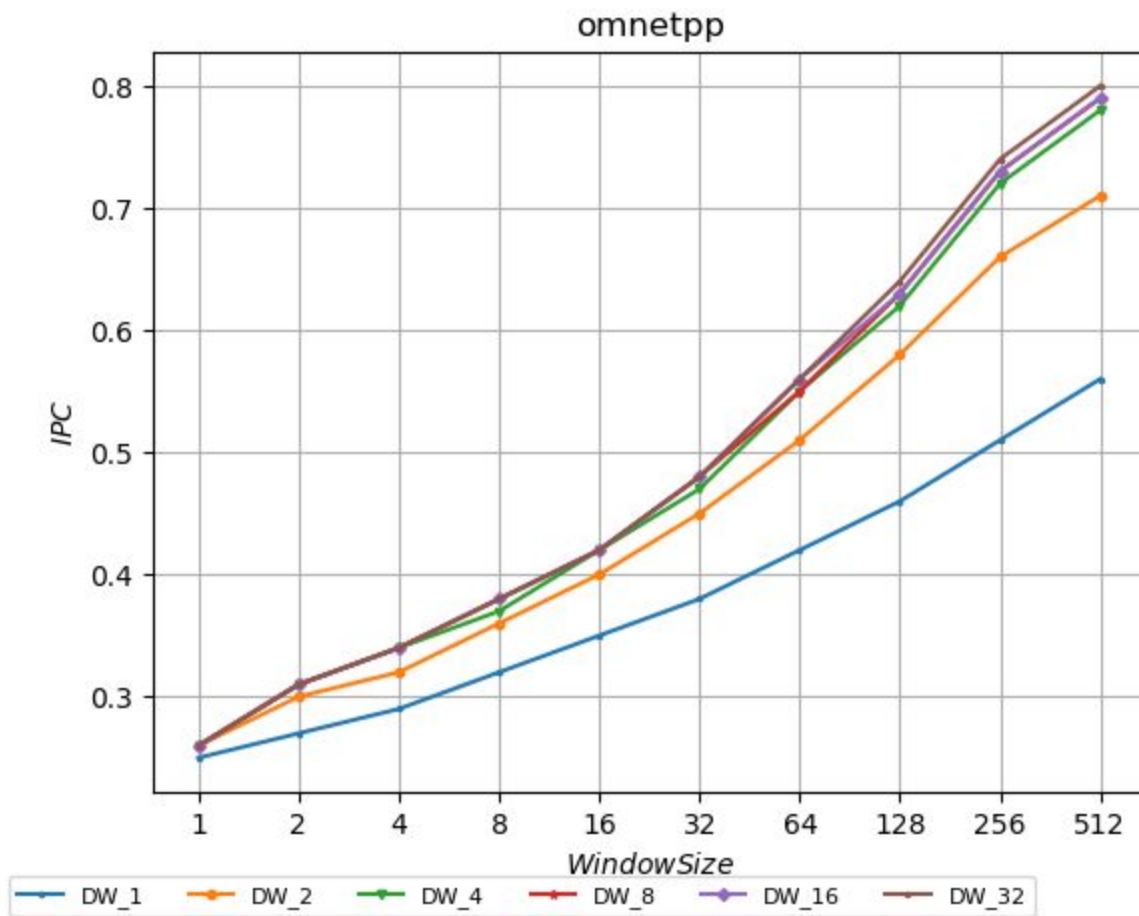
➤ 458.sjeng



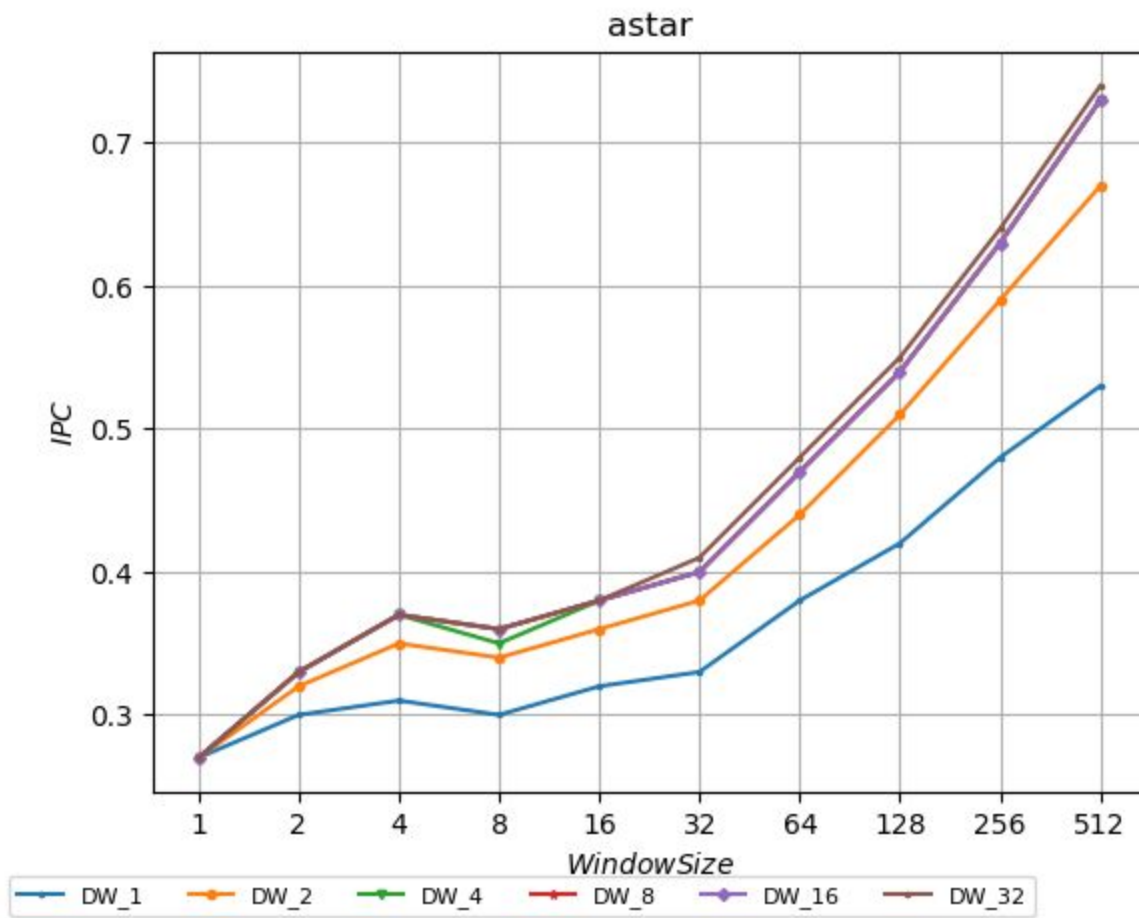
➤ 459.GemsFDTD



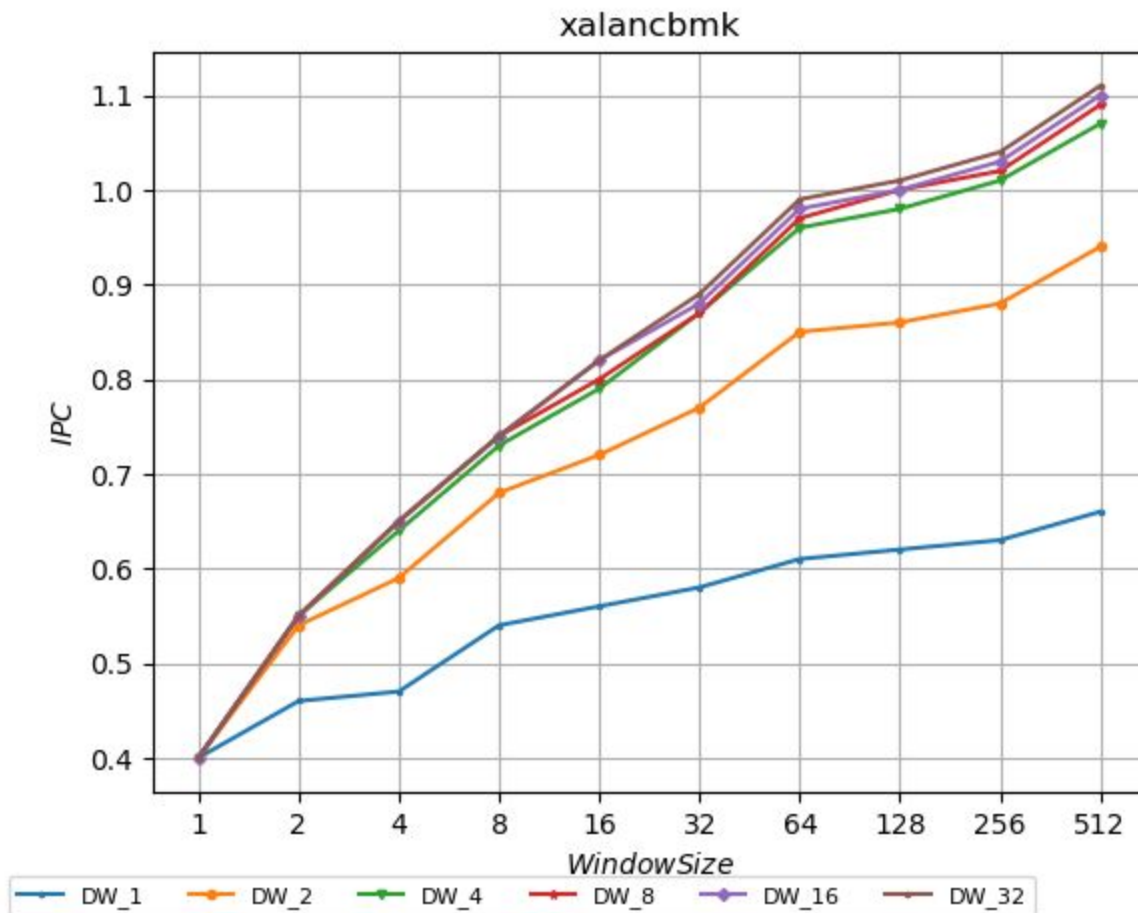
➤ 471.omnetpp



➤ 473.astar



➤ 483.xalancbmk



Συμπεράσματα

Εξετάζοντας το window size παρατηρούμε πως όσο αυξάνεται σε γενικές γραμμές αυξάνεται και η απόδοση.

Στις περισσότερες περιπτώσεις, όσο αυξάνεται το ένα μέγεθος τόσο αυξάνεται και το άλλο.

Σε κάποια μετροπρογράμματα (mcf, gcc) για πολύ μικρά window sizes και ορισμένα display widths, έχουμε σταθερό ή ελαφρώς μειούμενο IPC.

Σε ορισμένα μετροπρογράμματα από την άλλη, όπως τα mcf και cactus, αυξάνονται μέχρι και window size 256 και για 512 μένουν σταθερά είτε μειώνονται ελαφρώς.

Τέλος σε άλλα μετροπρογράμματα (sjeng, htmmer, gobmk) η απόδοση σταματάει να αυξάνεται σημαντικά ξεκινώντας από window size 8 ή 16. Ταυτόχρονα είναι σημαντικό να παρατηρήσουμε πως συχνά υπάρχουν διαφορές στις μεταβολές της απόδοσης σε σχέση με το window size ανάλογα και με το display length, οπότε υπάρχει συσχέτιση μεταξύ τους.

Μελετώντας το dispatch width βλέπουμε πως όταν είναι ίσο με 1 η απόδοση είναι αρκετά μικρότερη από τις άλλες περιπτώσεις, για dw=2 υπάρχει μεγάλη βελτίωση, ενώ για 4 και πάνω υπάρχει σχεδόν μηδαμινή βελτίωση στις περισσότερες περιπτώσεις. Δηλαδή τα παρόντα μετροπρογράμματα δεν εκμεταλλεύονται στο έπακρο το μεγάλο dispatch width.

iii) Πώς επηρεάζει η κάθε παράμετρος την κατανάλωση ενέργειας και το μέγεθος του τσιπ;

Για το ερώτημα αυτό θα χρησιμοποιήσουμε το μέγεθος Energy-Delay Product (EDP) που συνδυάζει το χρόνο εκτέλεσης ενός benchmark με την ενέργεια που δαπανήθηκε.

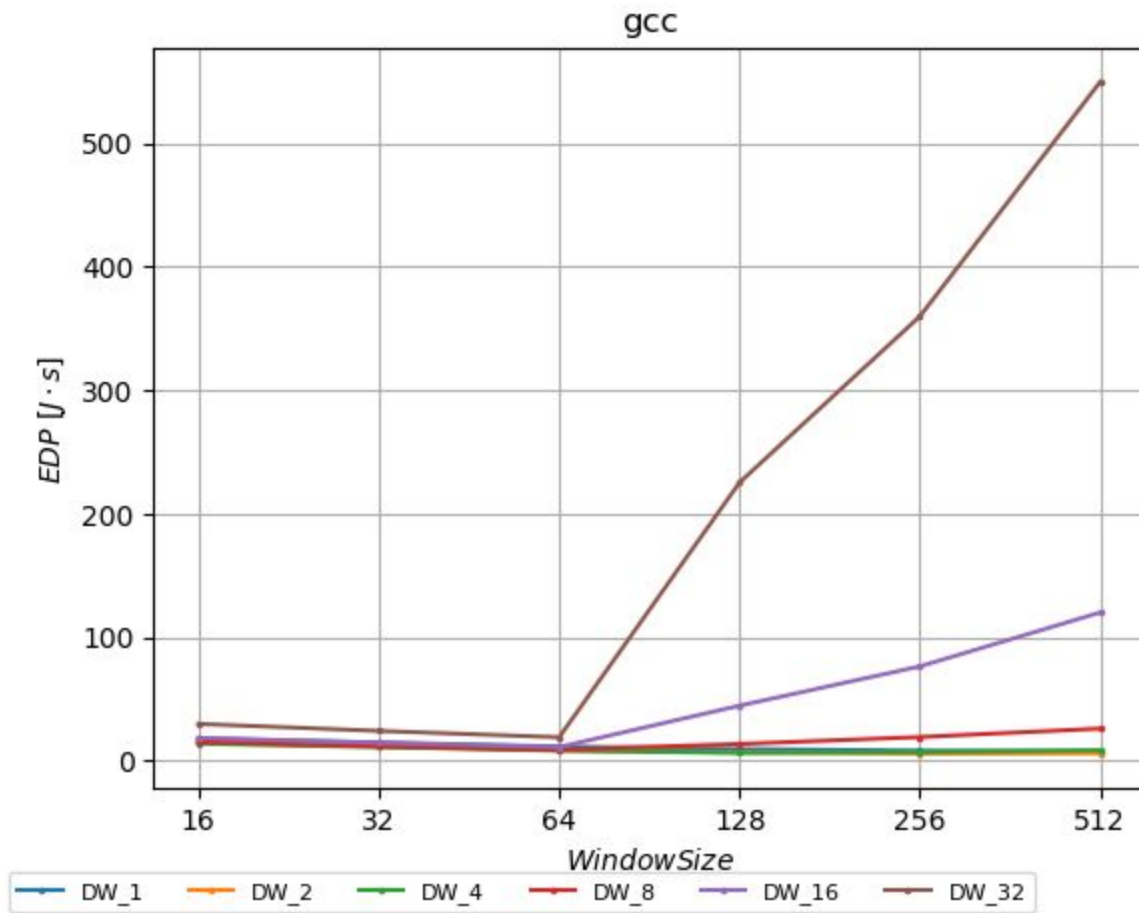
Για την αξιολόγηση της κατανάλωσης ενός επεξεργαστή χρησιμοποιείται ως μετρική η συνολική κατανάλωση ενέργειας σε Joules. Όμως, επειδή πολλές φορές απαιτείται η μελέτη της επίδρασης διαφόρων χαρακτηριστικών του επεξεργαστή όχι μόνο στην κατανάλωση αλλά ταυτόχρονα και στην επίδοσή του, θα χρησιμοποιηθεί η μετρική energy-delay product (EDP), το οποίο για την εκτέλεση ενός benchmark ορίζεται ως το γινόμενο της ενέργειας επί τον χρόνο εκτέλεσης του benchmark: $EDP = \text{Energy(J)} * \text{runtime(sec)}$

Επίσης, για να δώσουμε περισσότερο βάρος στον χρόνο εκτέλεσης υψώνουμε το runtime στο τετράγωνο και στον κύβο και έχουμε τα $ED^2P = \text{Energy(J)} * \text{runtime}^2(\text{sec})$ και $ED^3P = \text{Energy(J)} * \text{runtime}^3(\text{sec})$.

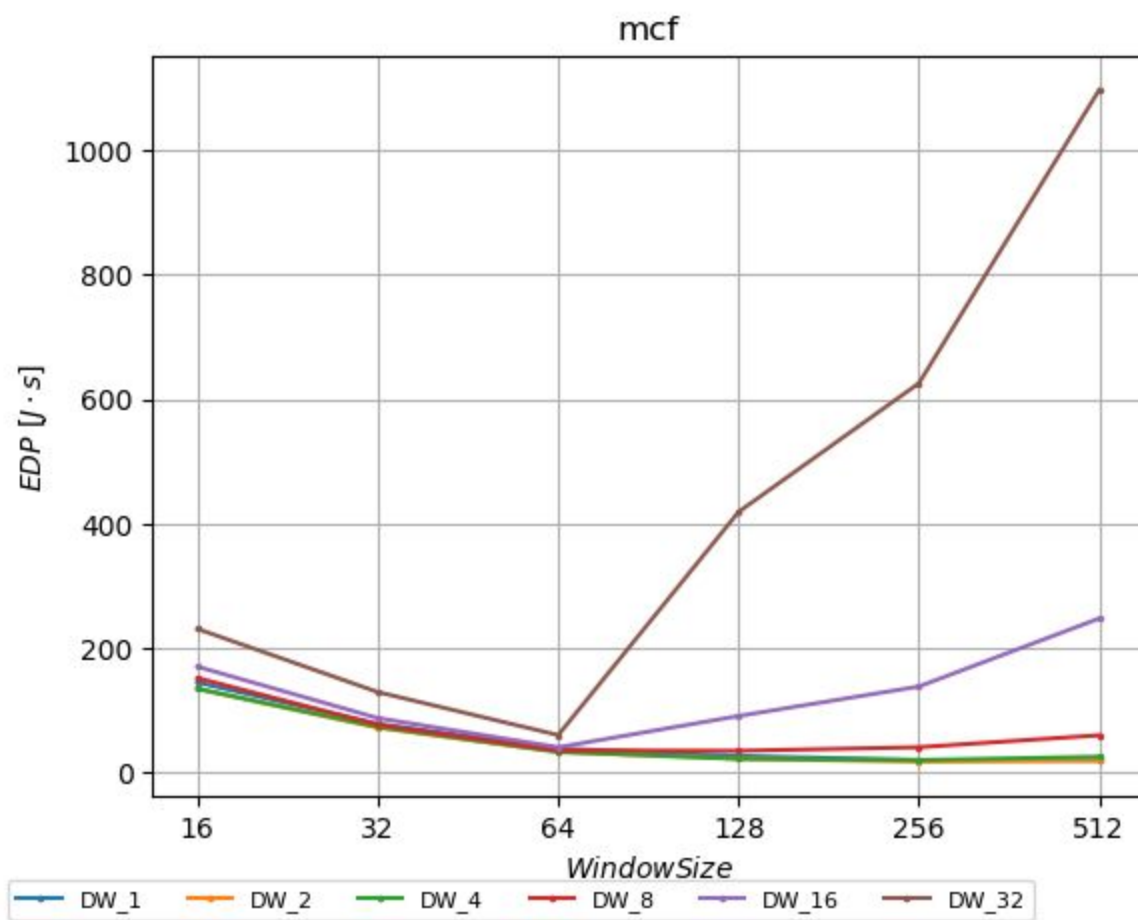
Παρακάτω παρουσιάζονται τα αποτελέσματα για τα EDP, ED^2P , ED^3P , Energy και Area σε γραφικές παραστάσεις.

EDP

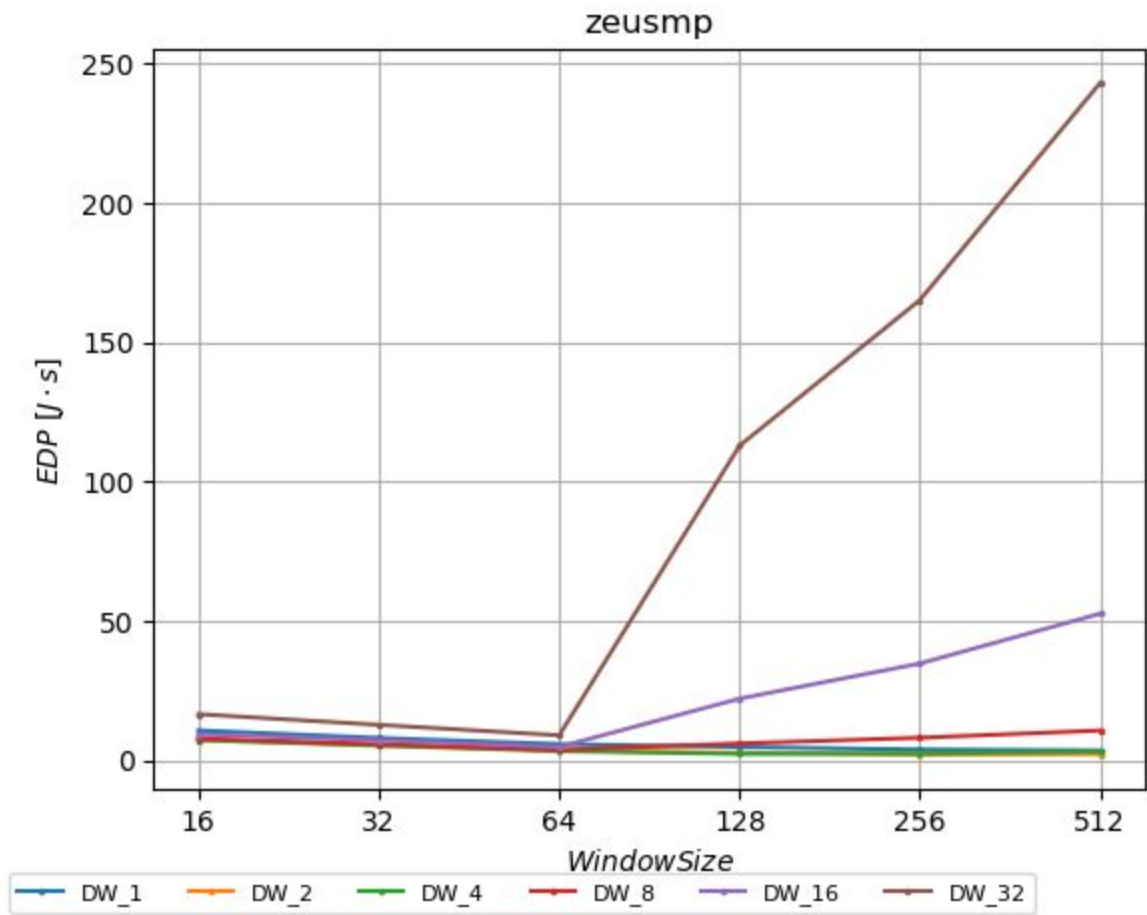
➤ 403.gcc



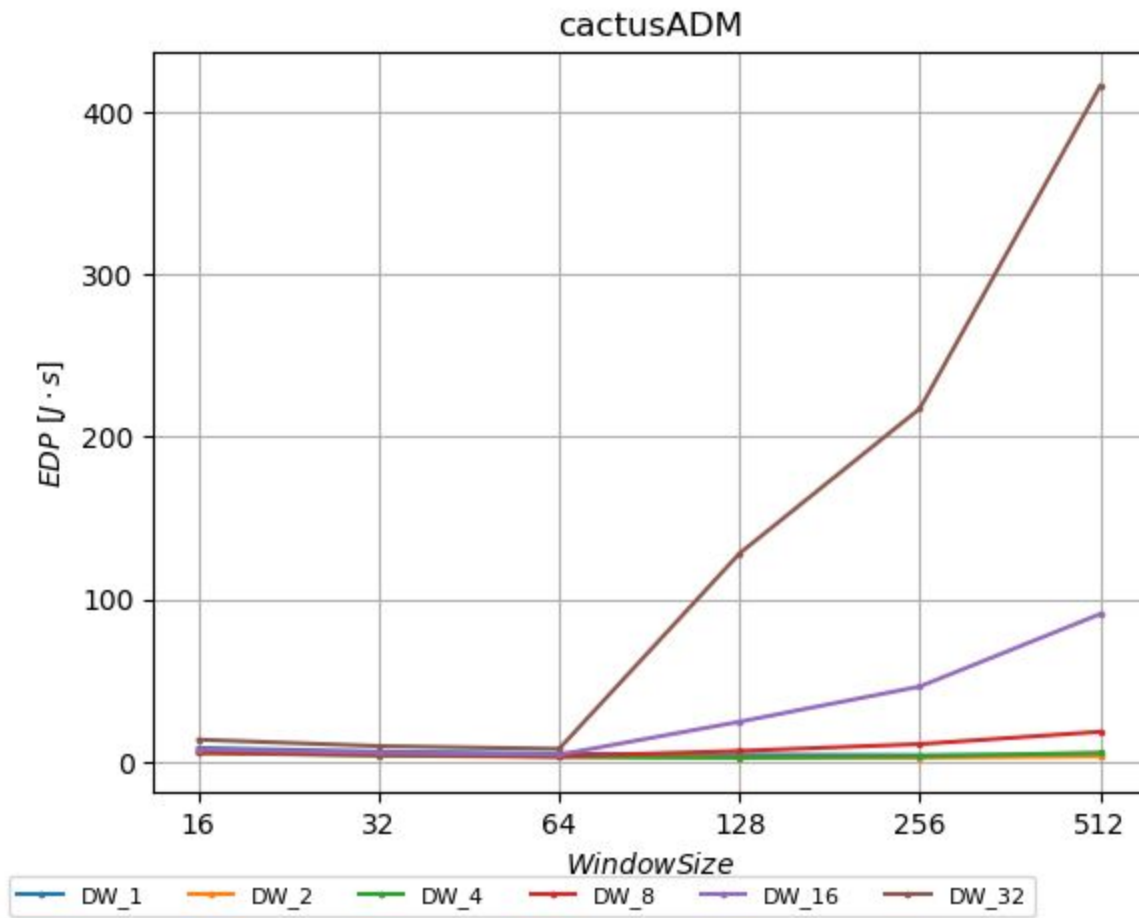
➤ 429.mcf



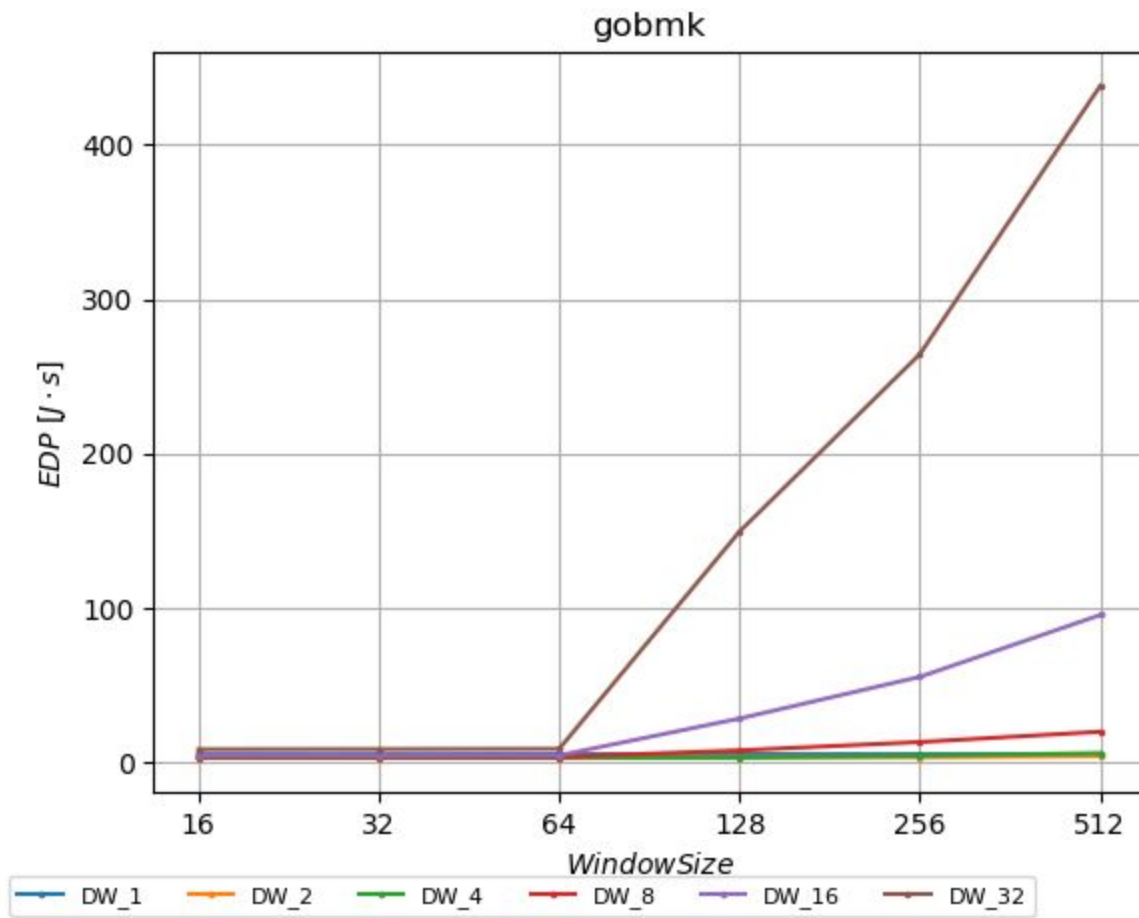
➤ 434.zeusmp



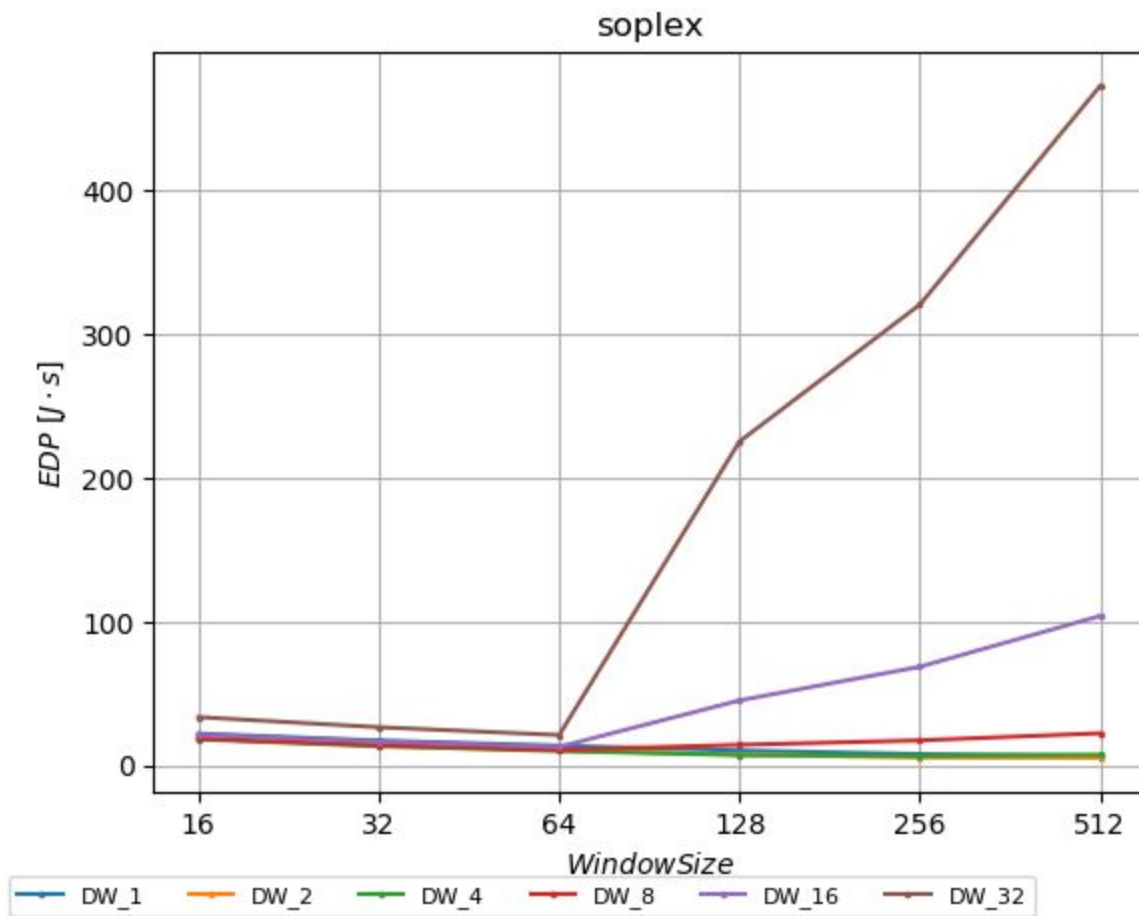
➤ 436.cactusADM



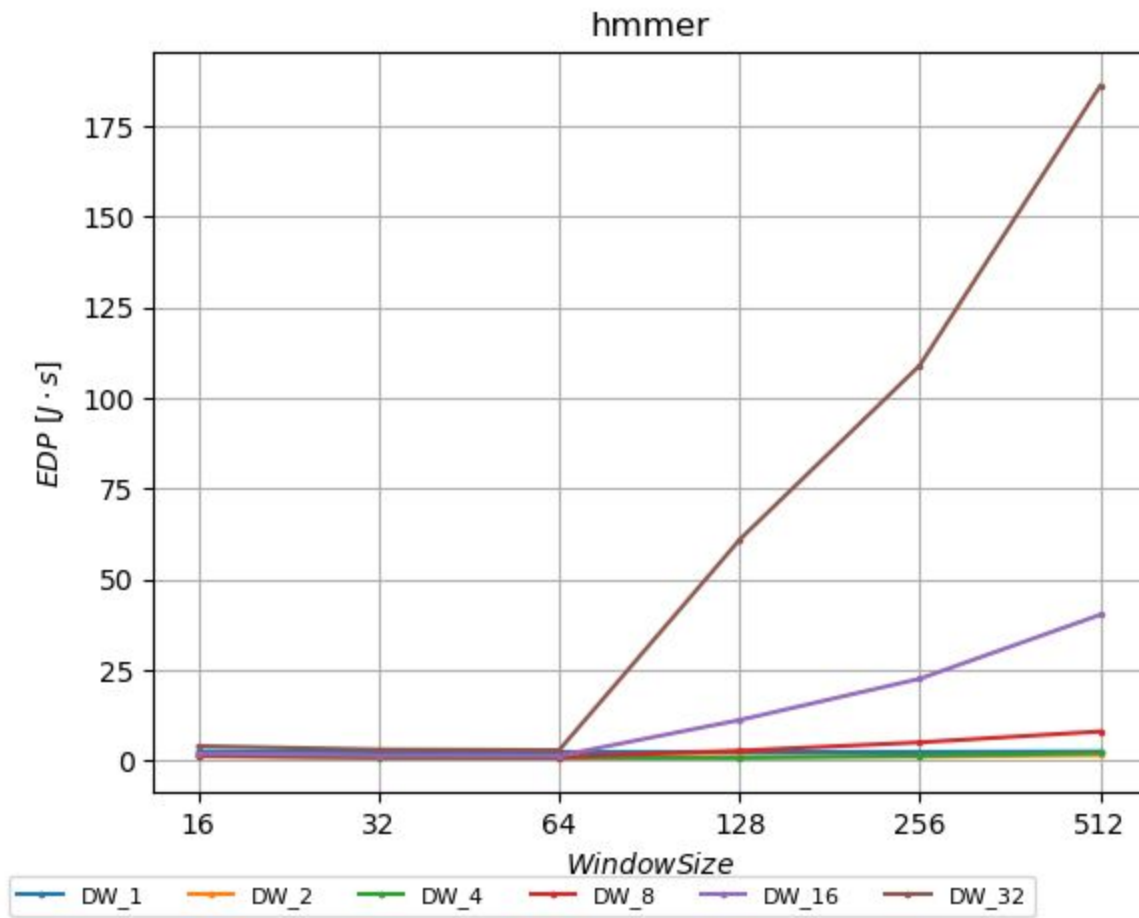
➤ 445.gobmk



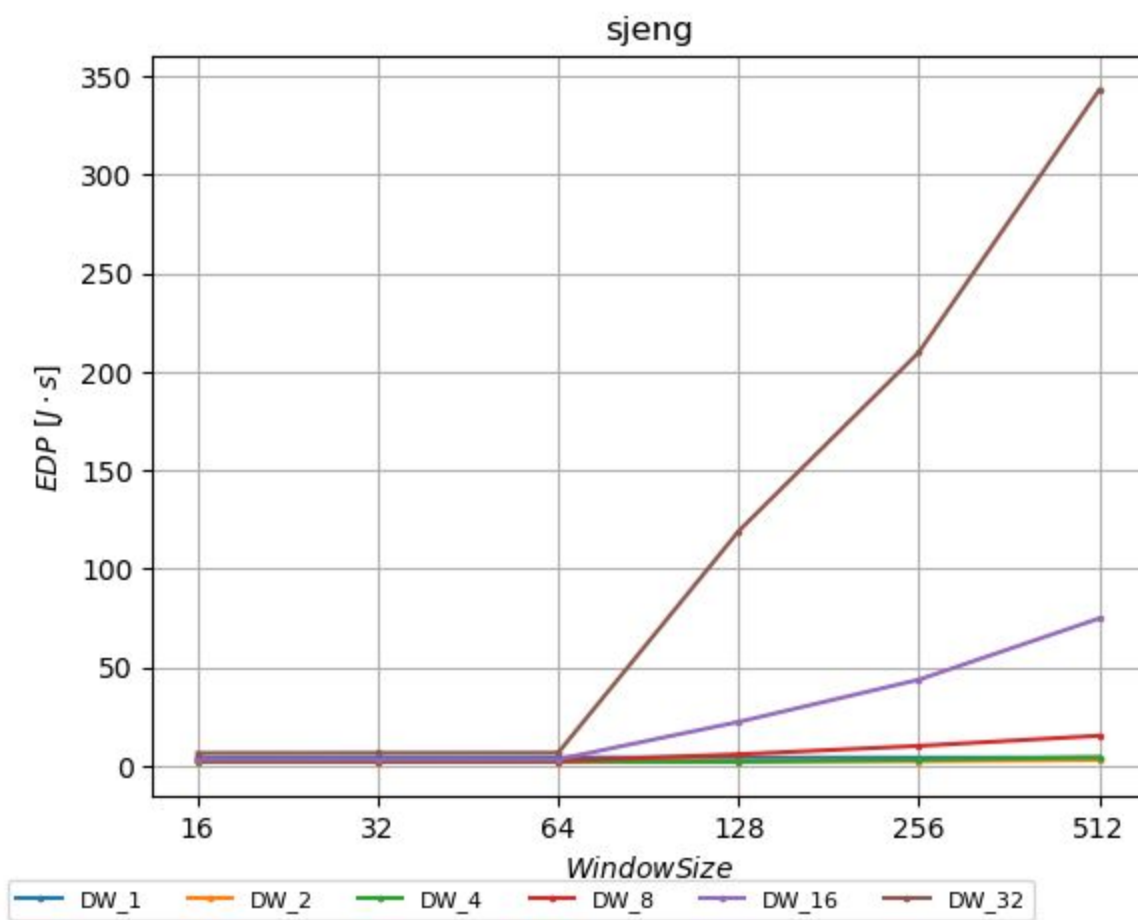
➤ 450.soplex



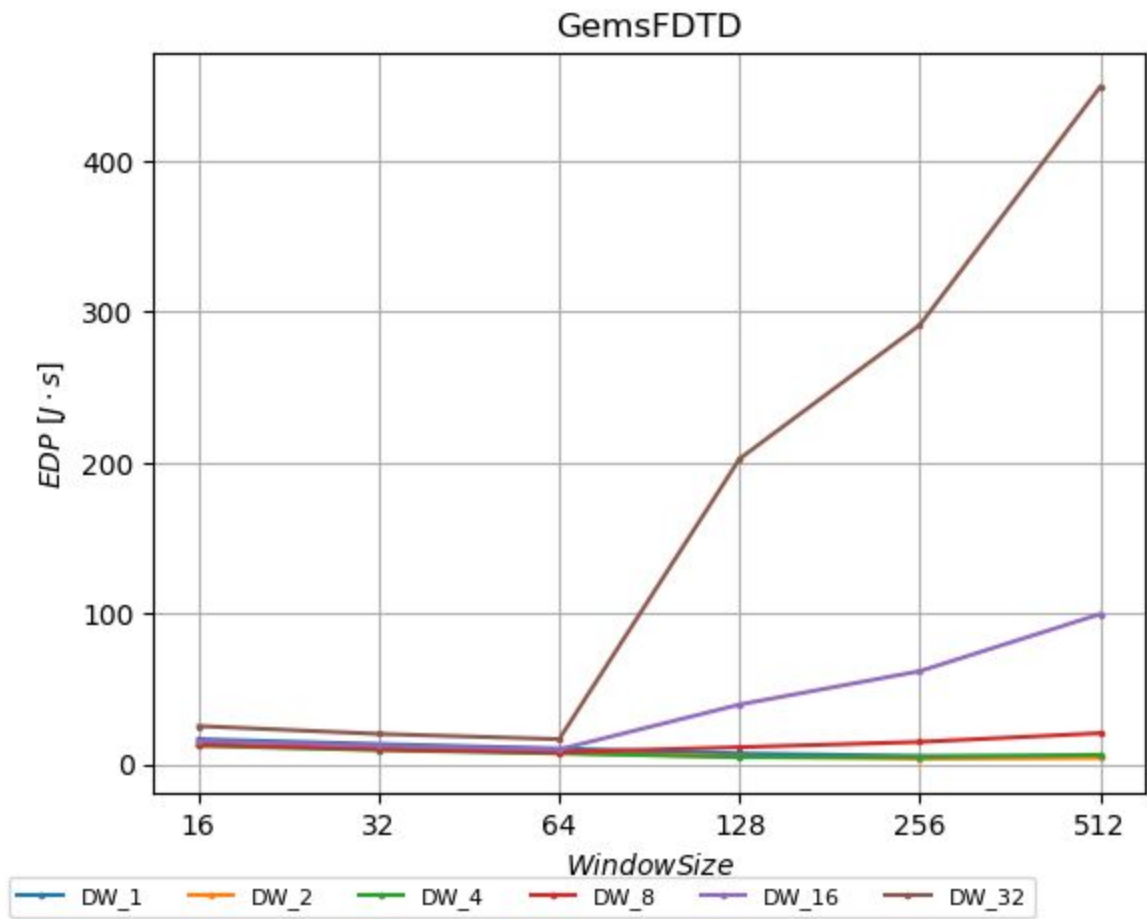
➤ 456.hmmmer



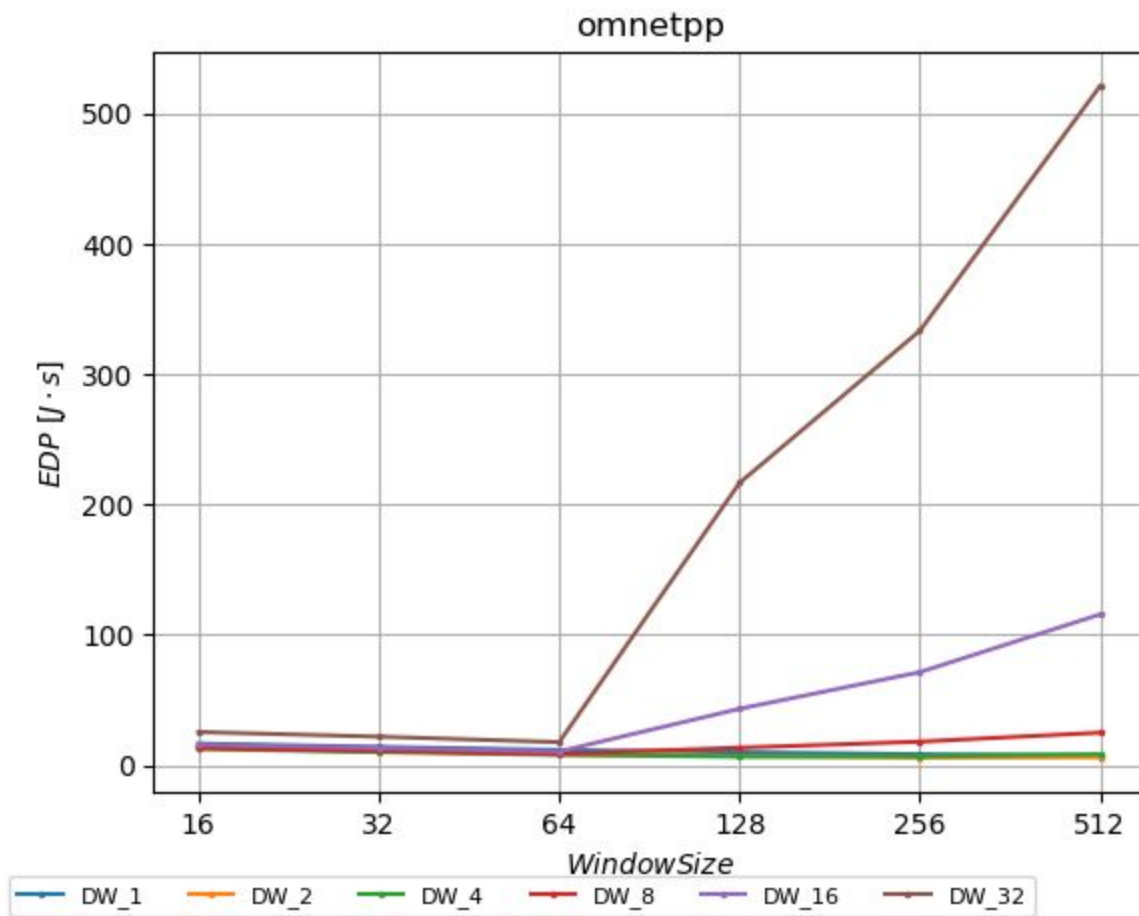
➤ 458.sjeng



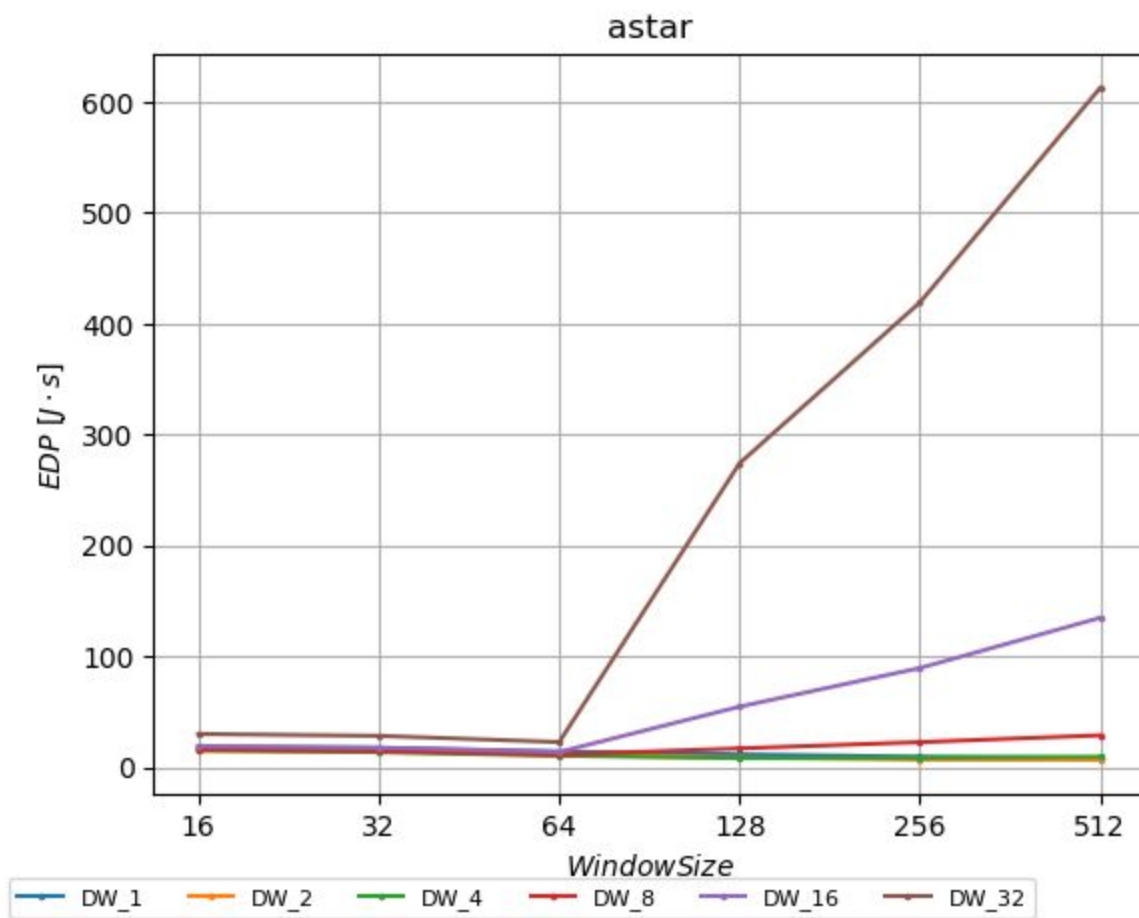
➤ 459.GemsFDTD



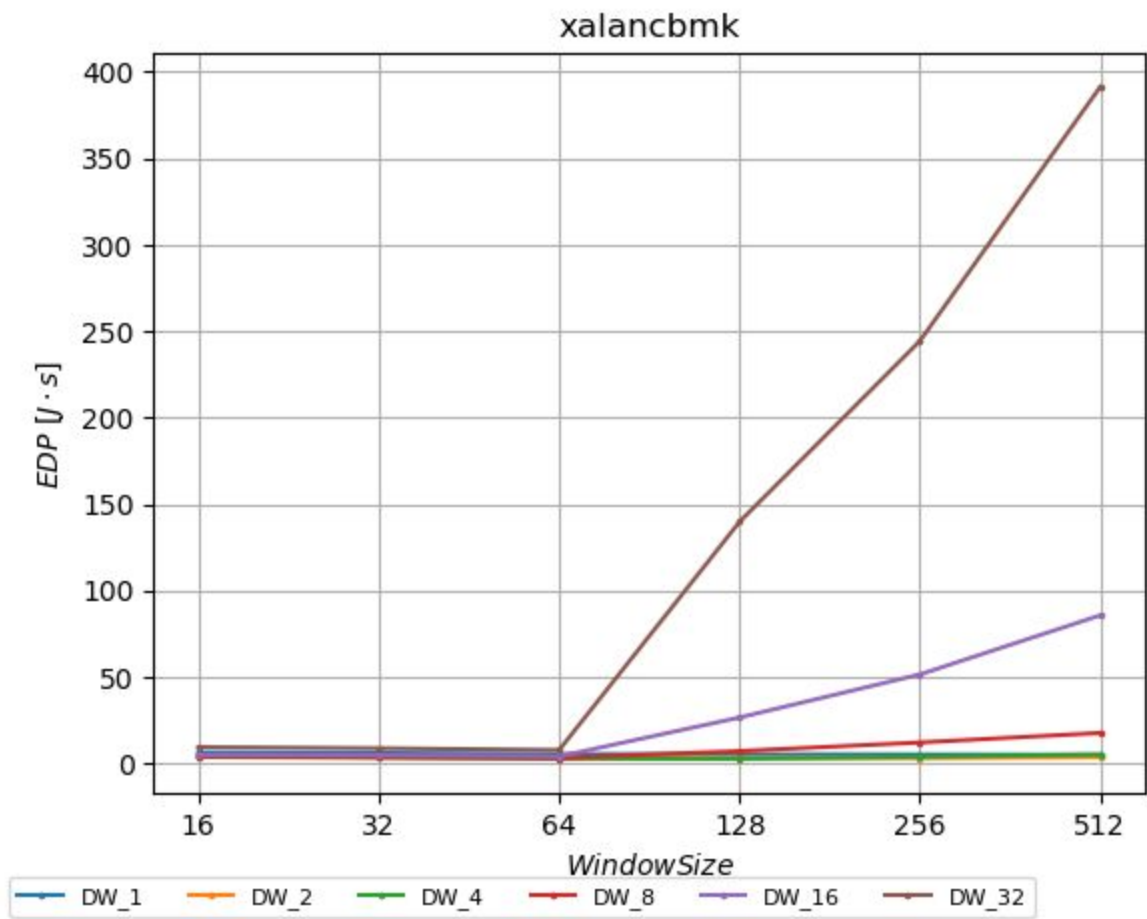
➤ 471.omnetpp



➤ 473.astar

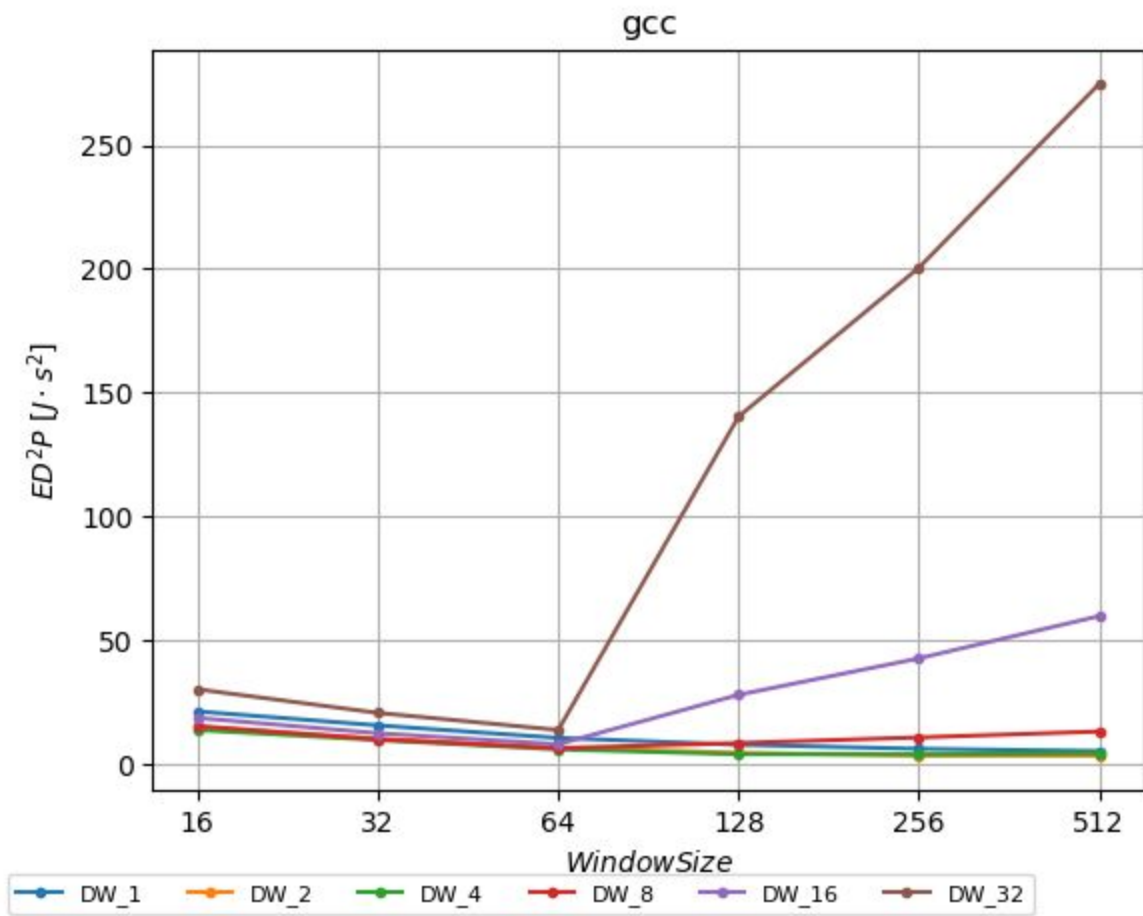


➤ 483.xalancbmk

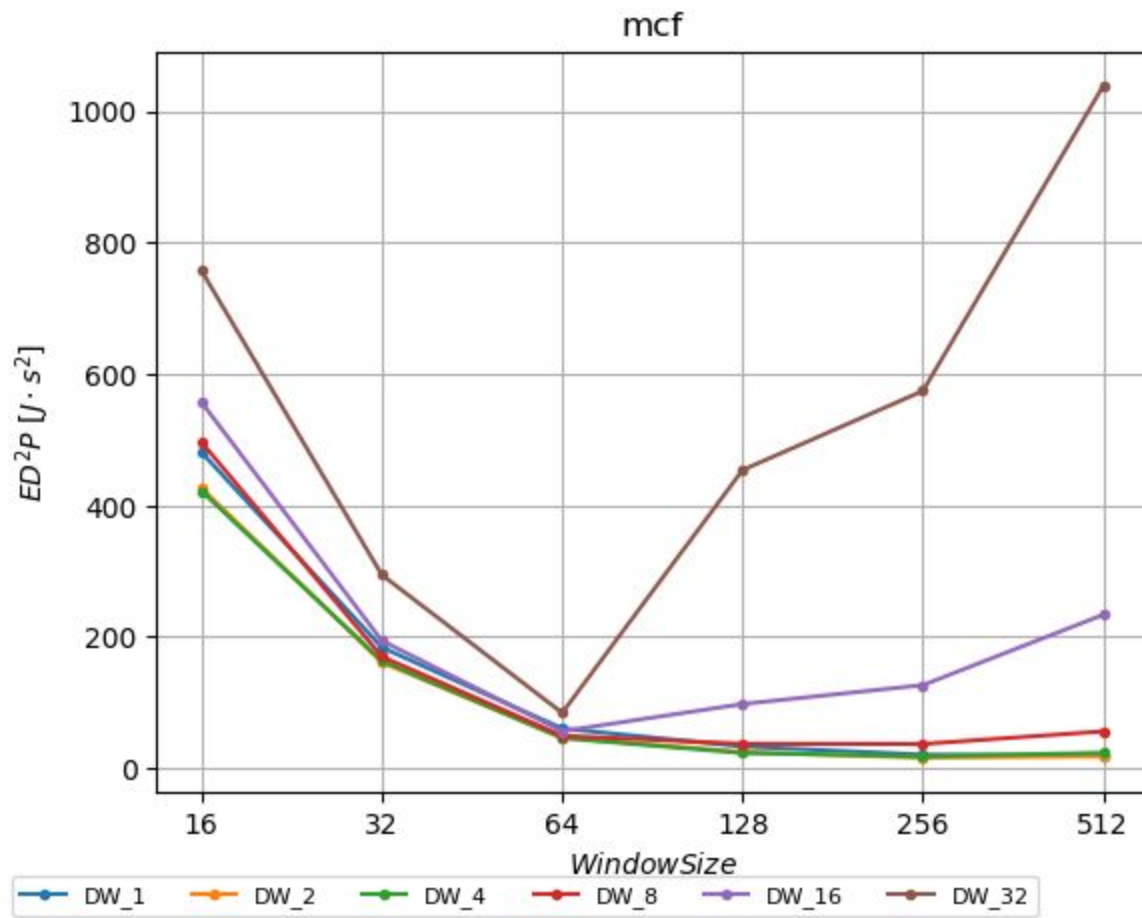


ED²P

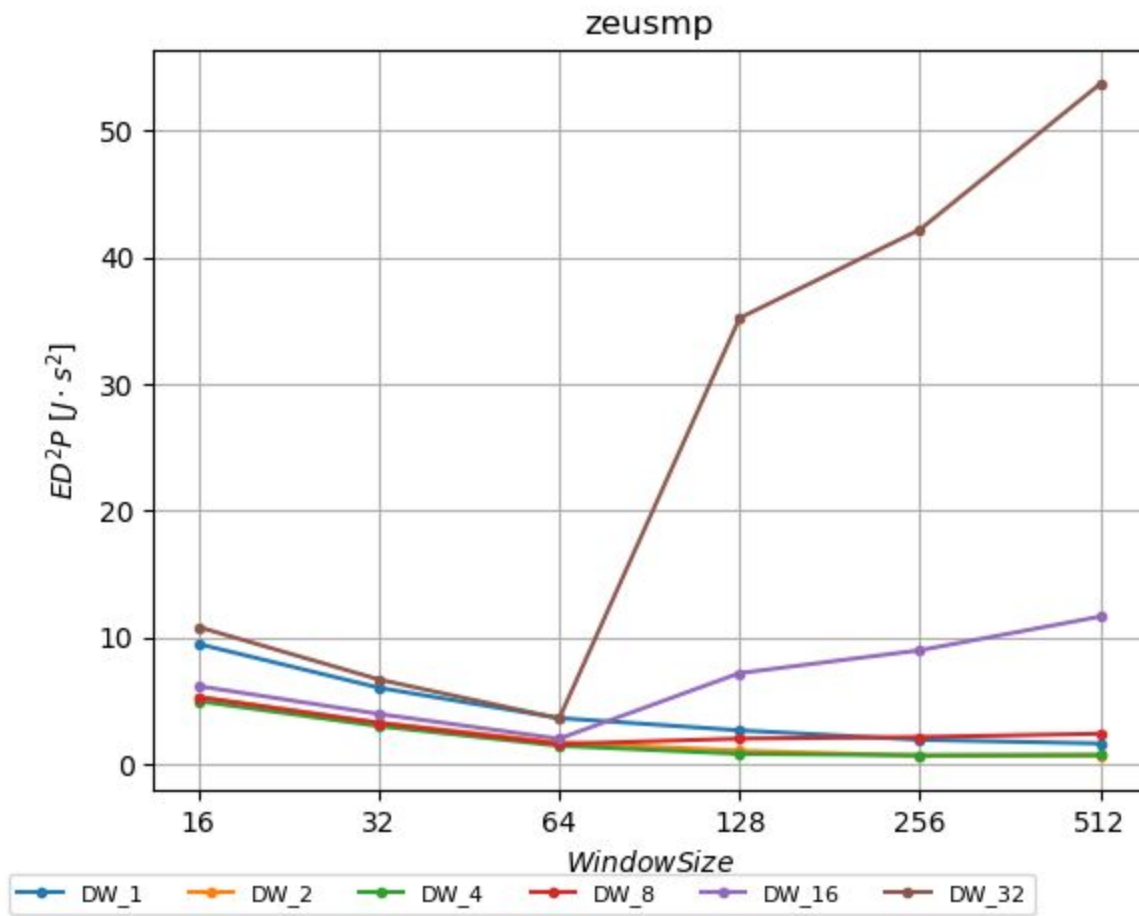
➤ 403.gcc



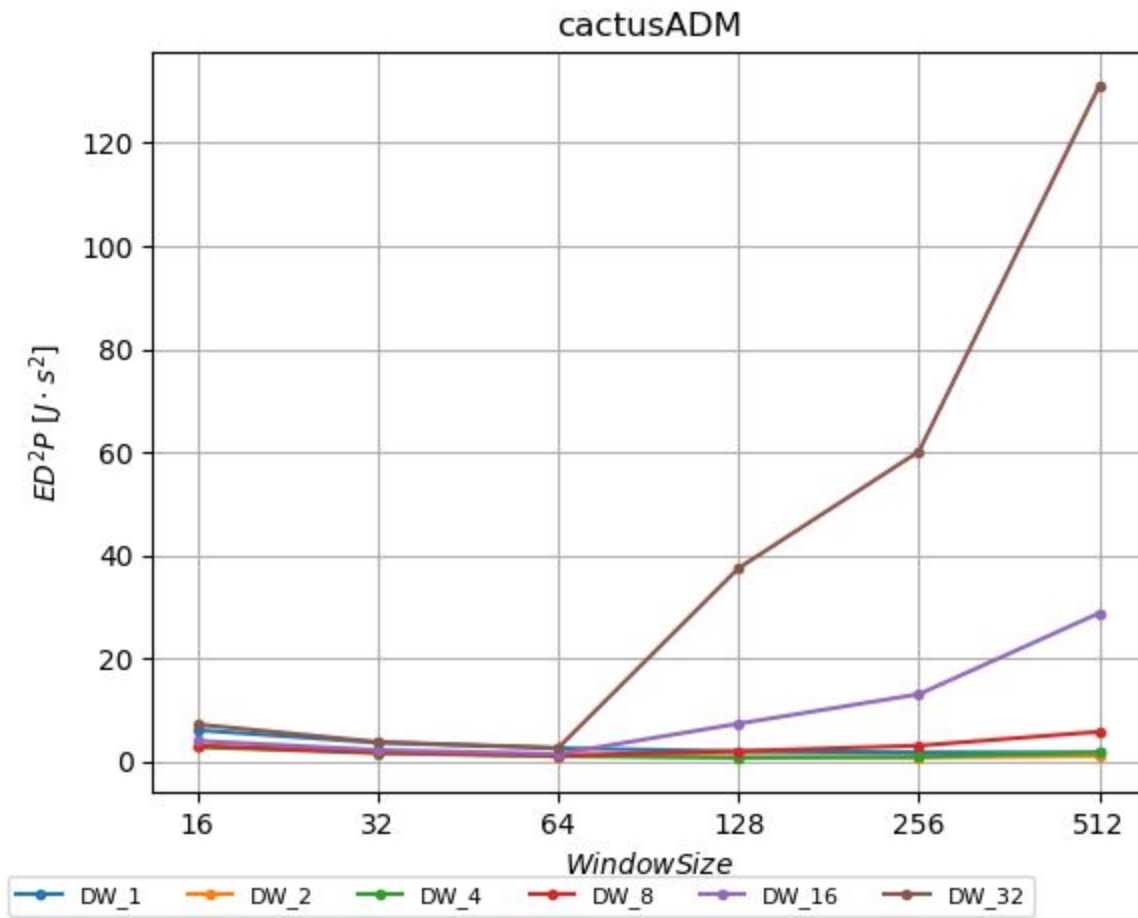
➤ 429.mcf



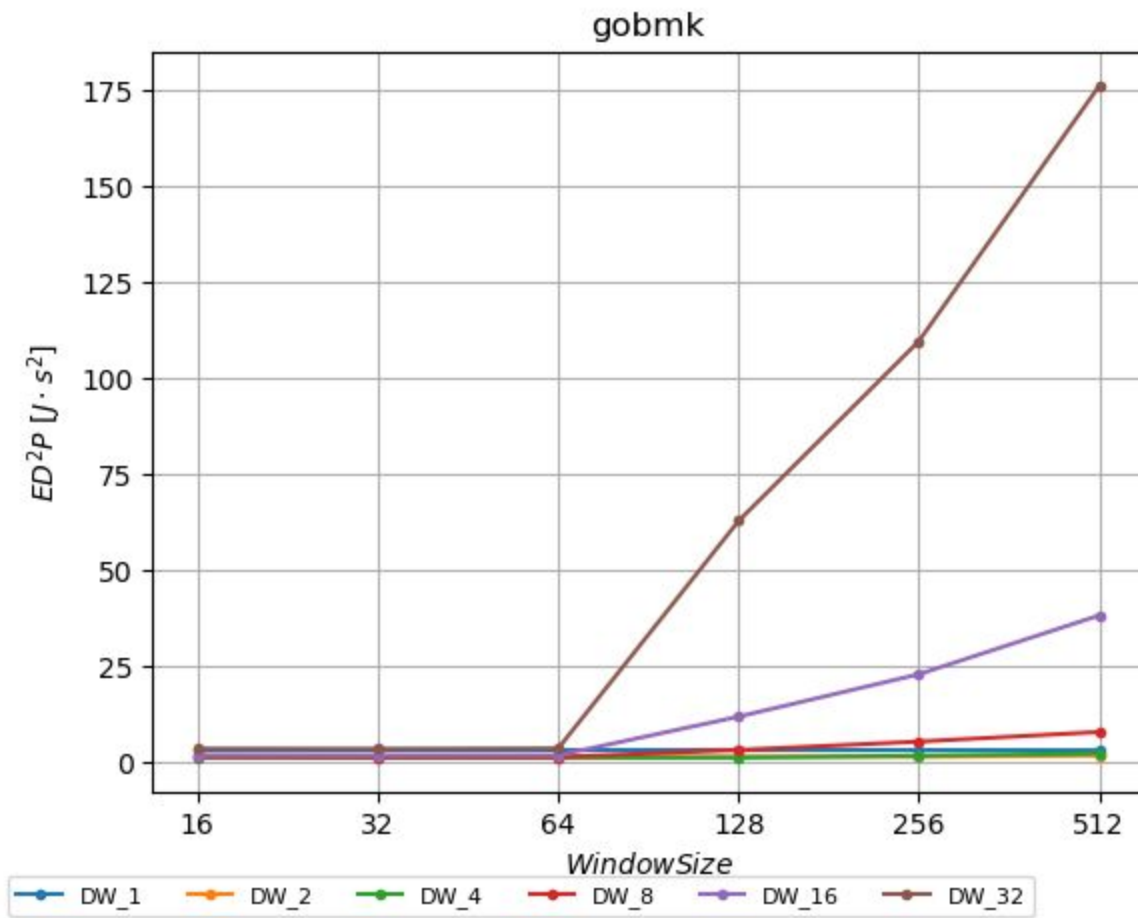
➤ 434.zeusmp



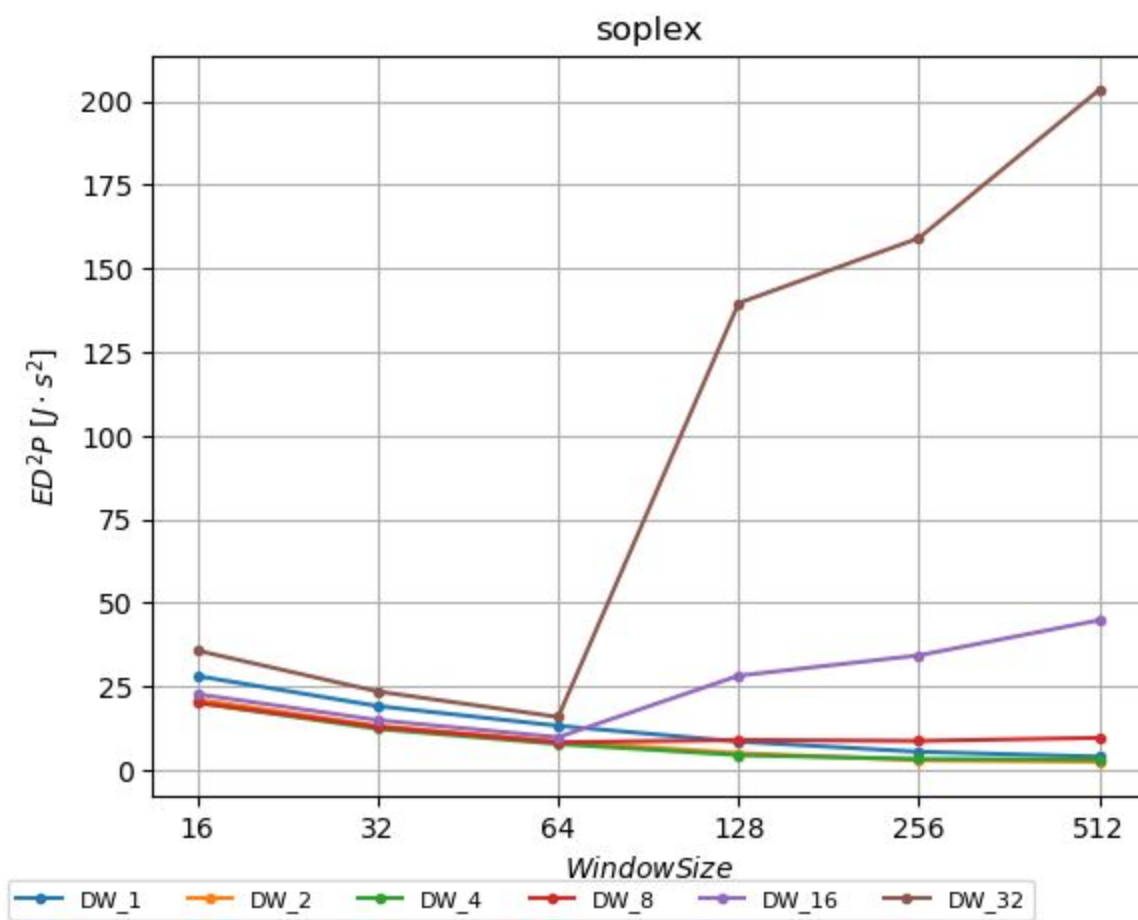
➤ 436.cactusADM



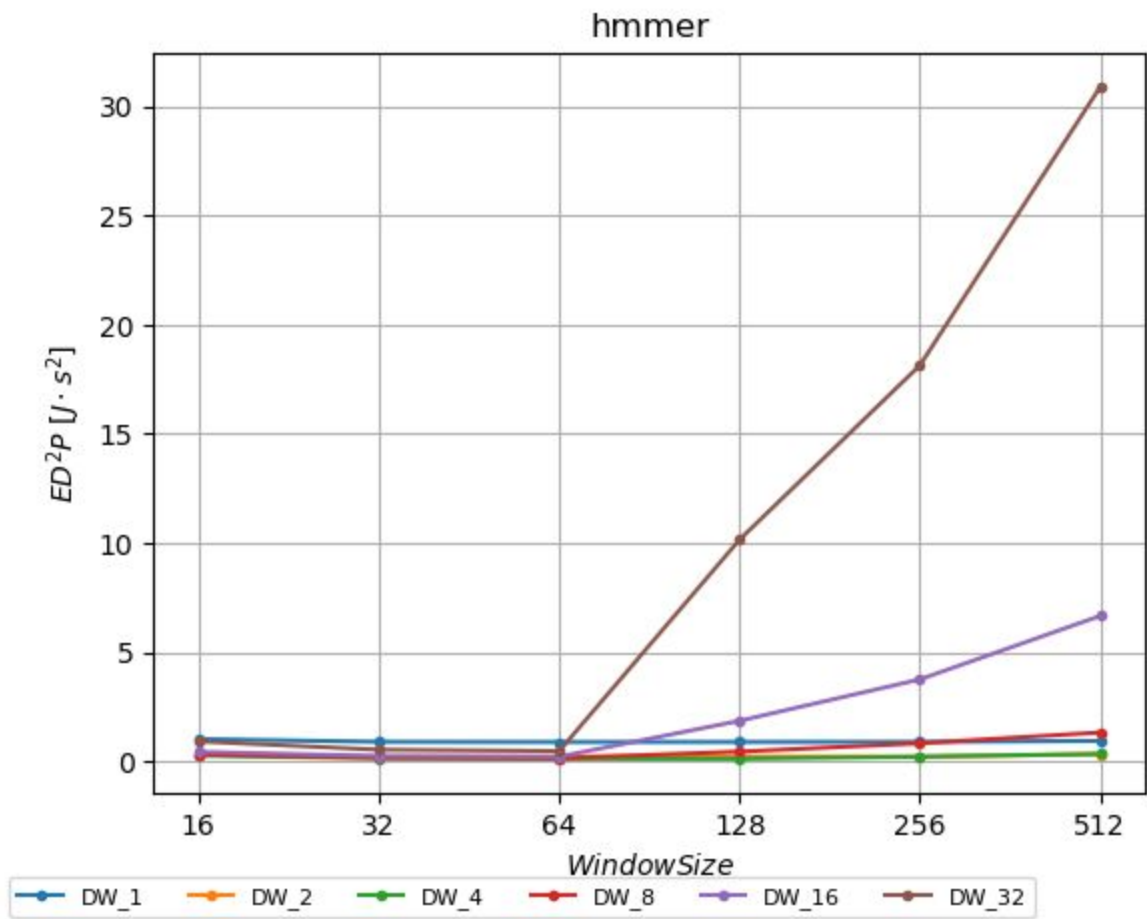
➤ 445.gobmk



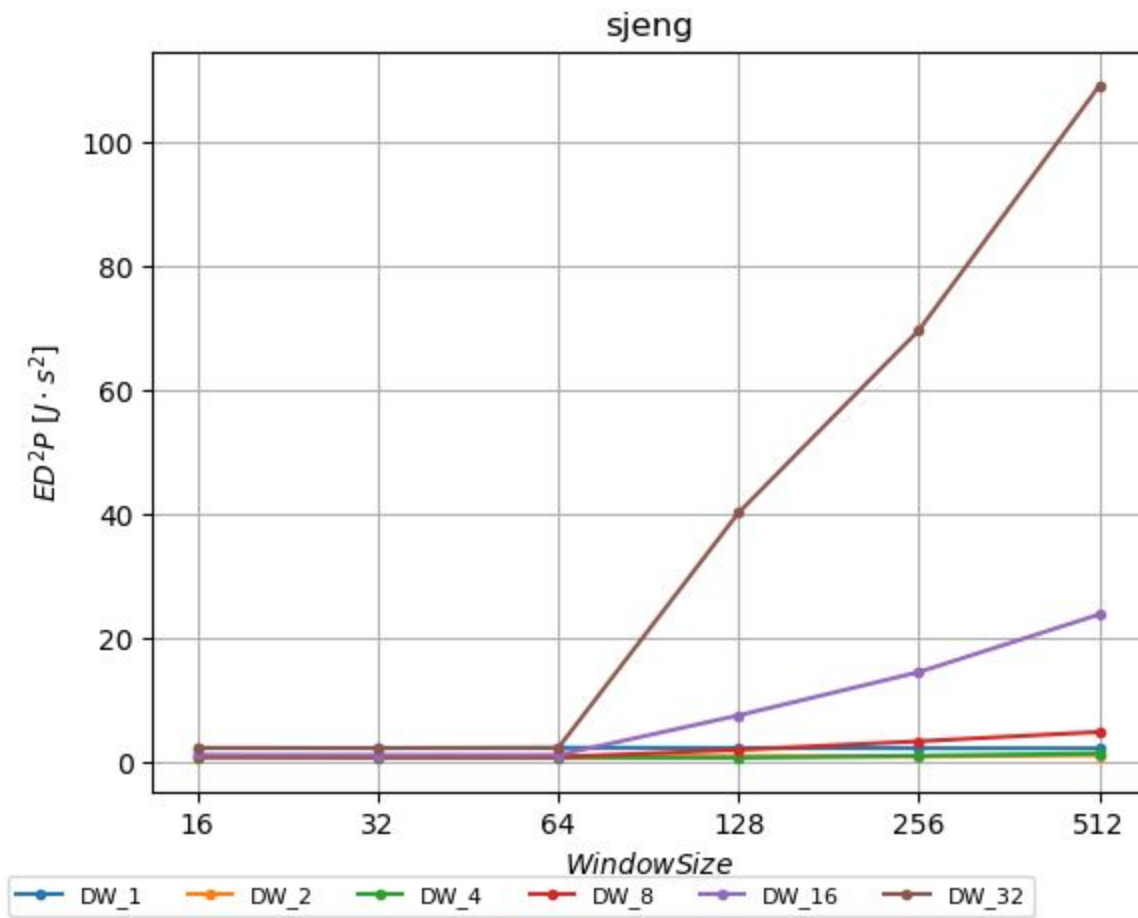
➤ 450.soplex



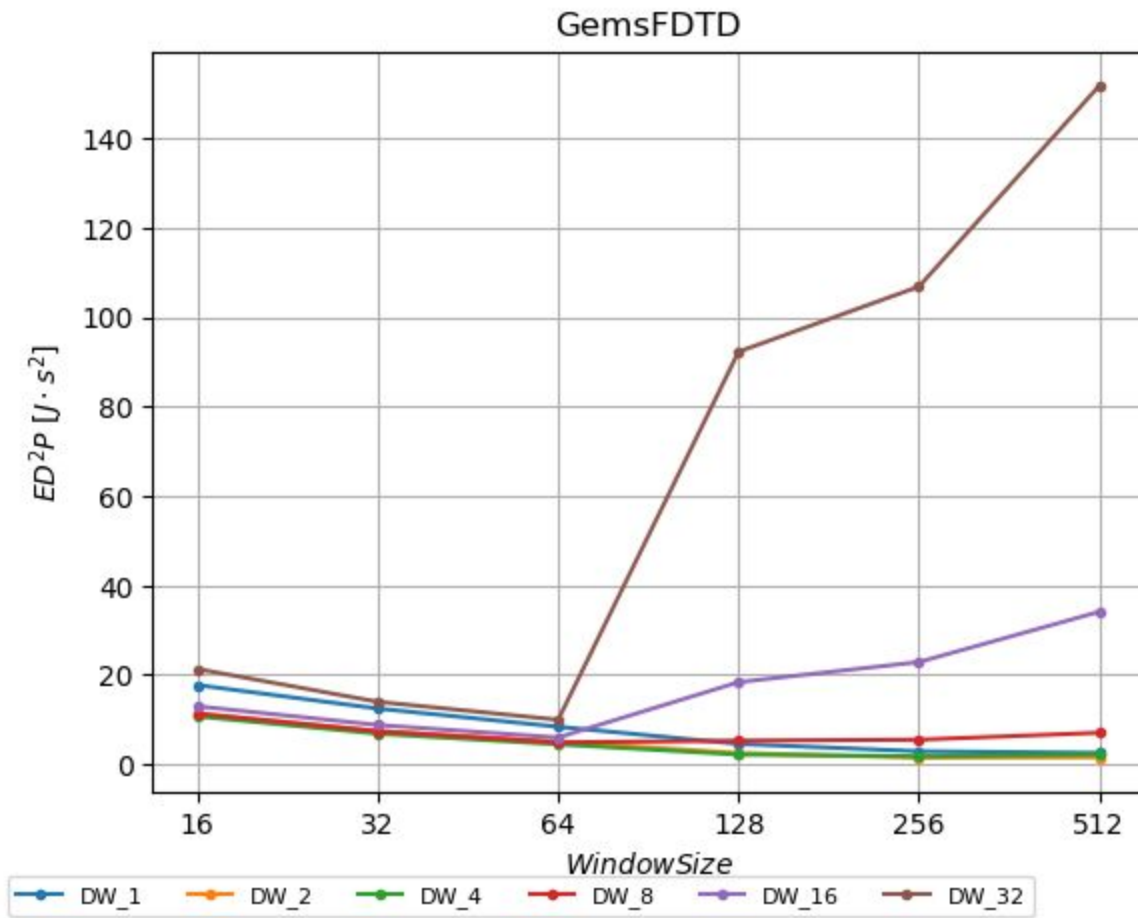
➤ 456.hmmmer



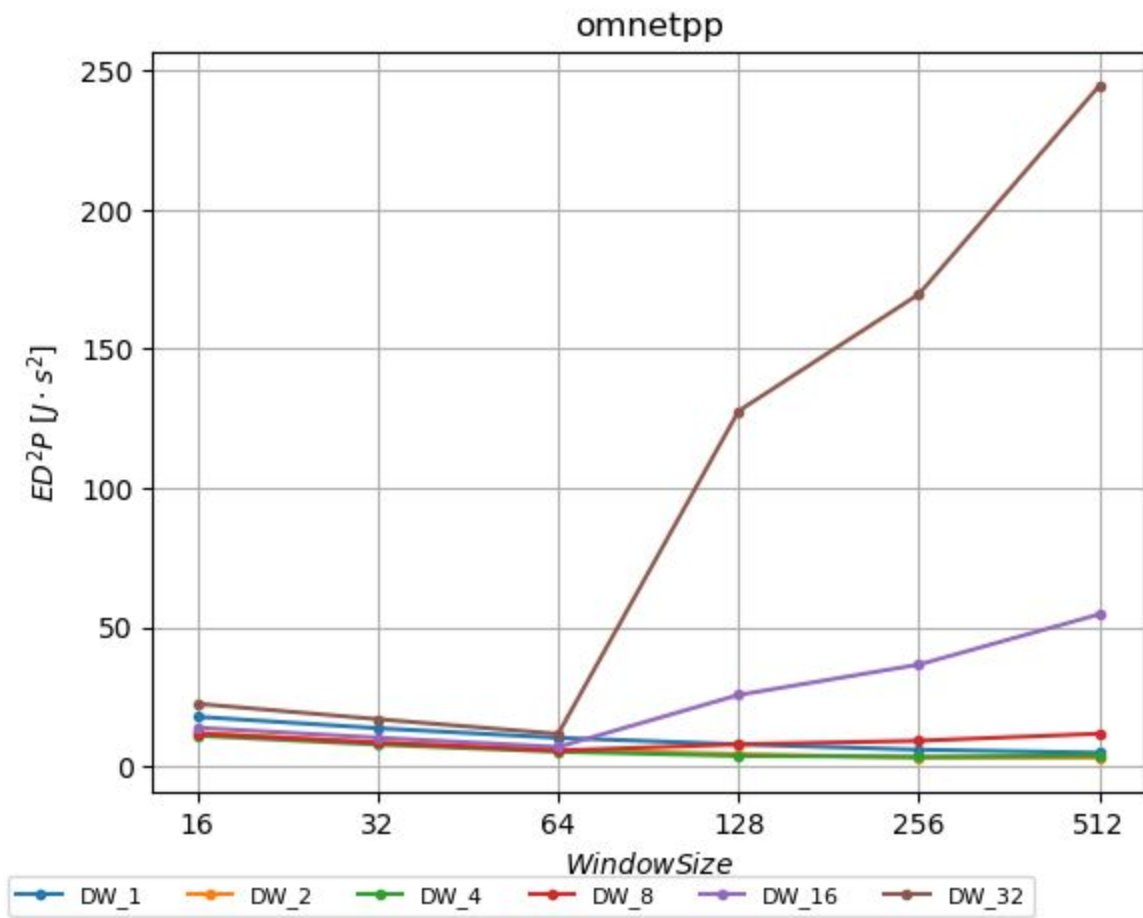
➤ 458.sjeng



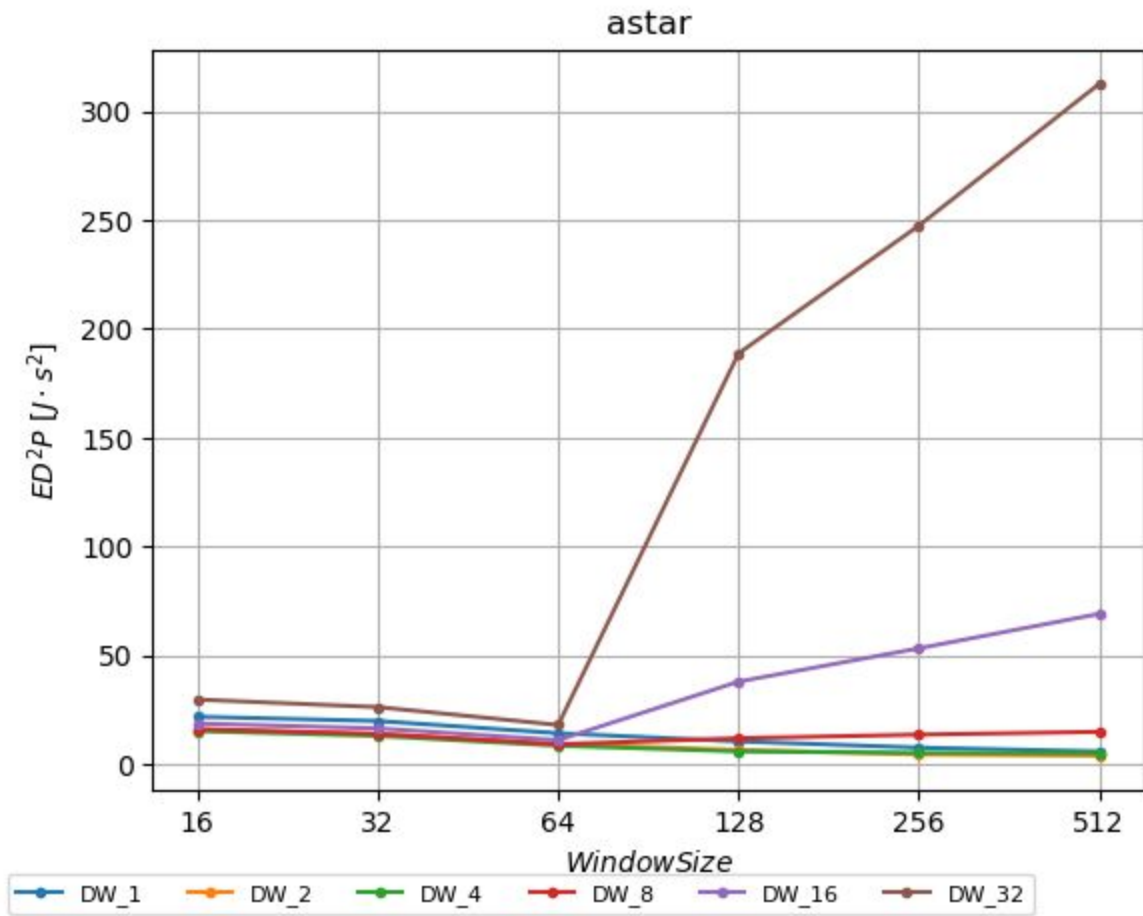
➤ 459.GemsFDTD



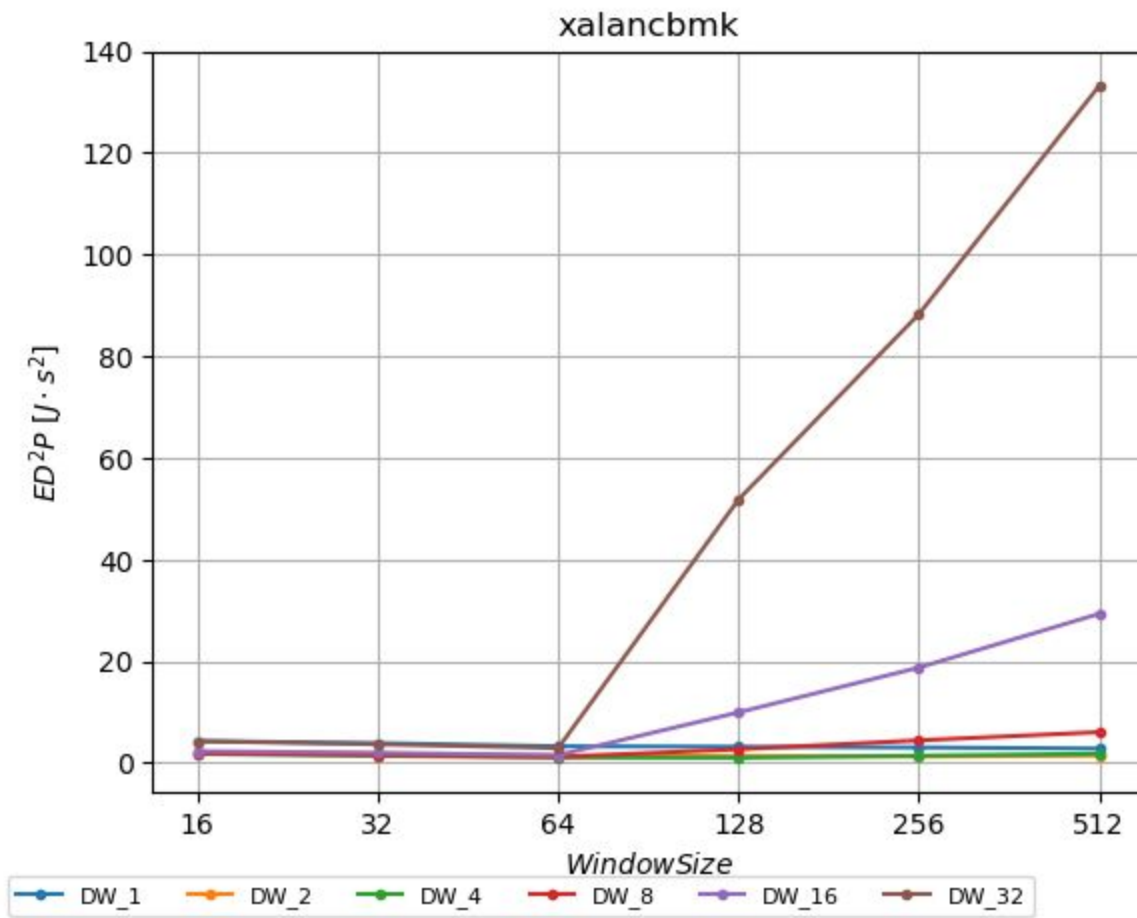
➤ 471.omnetpp



➤ 473.astar

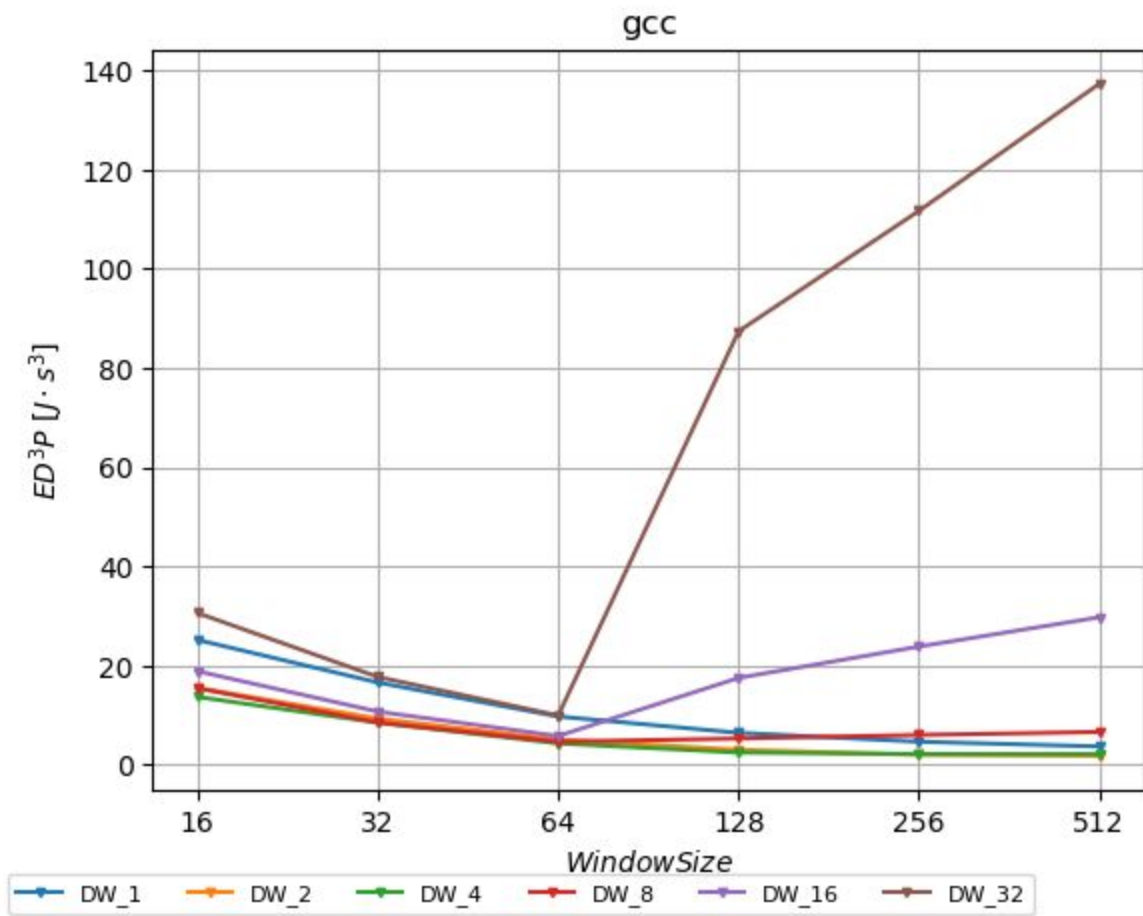


➤ 483.xalancbmk

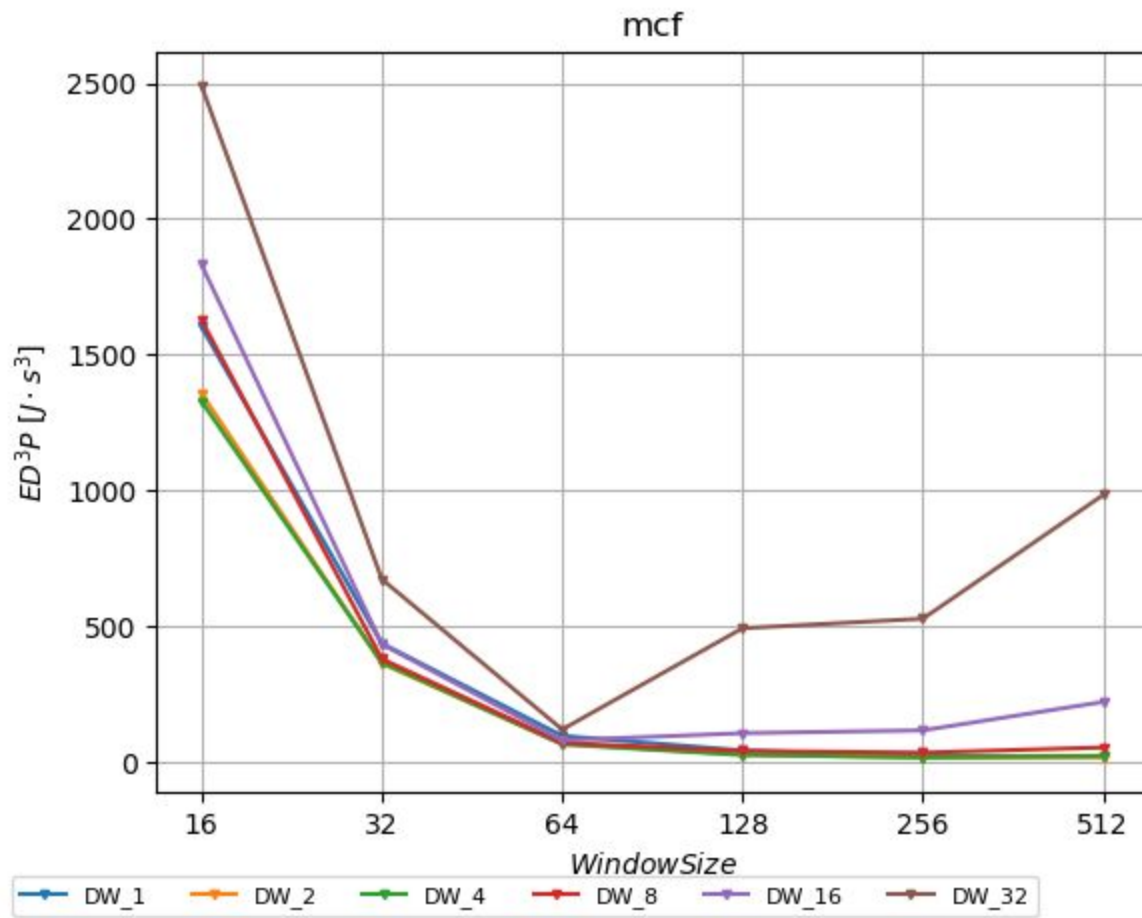


ED³P

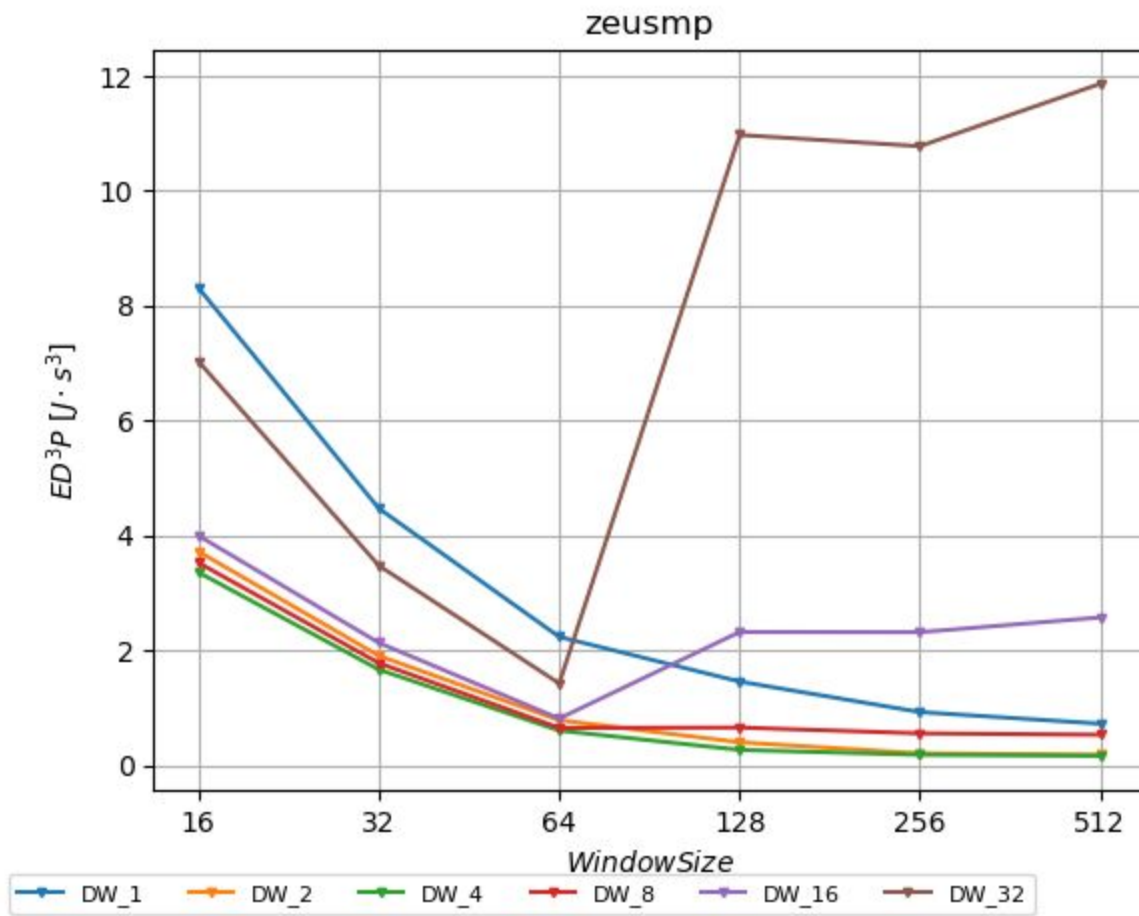
➤ 403.gcc



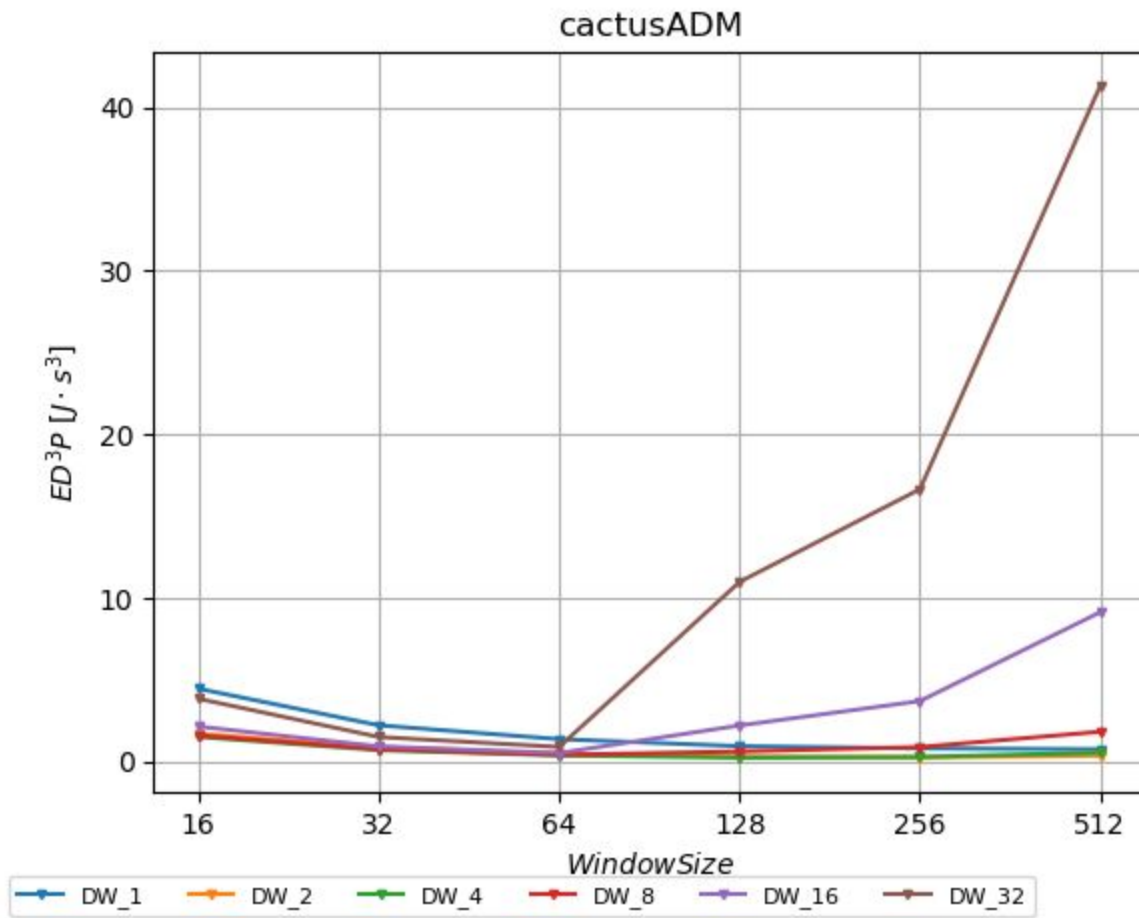
➤ 429.mcf



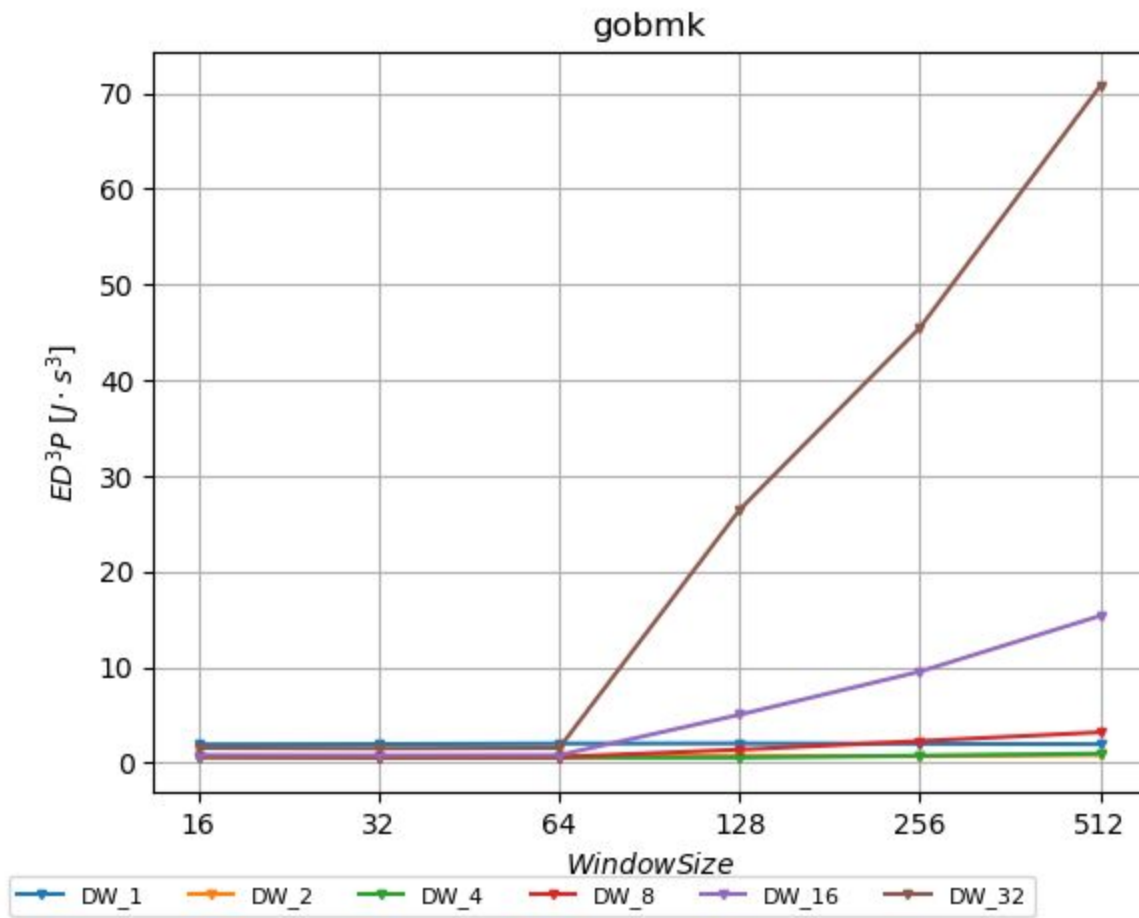
➤ 434.zeusmp



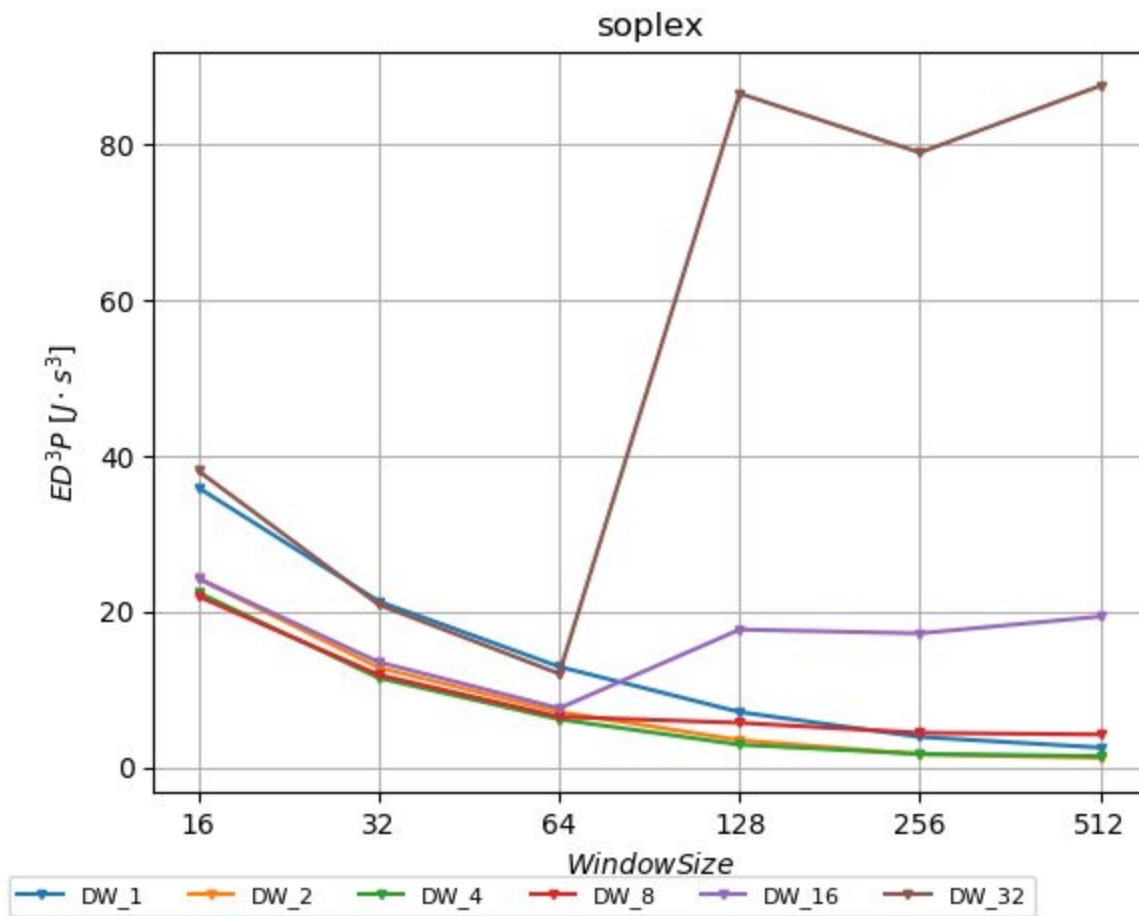
➤ 436.cactusADM



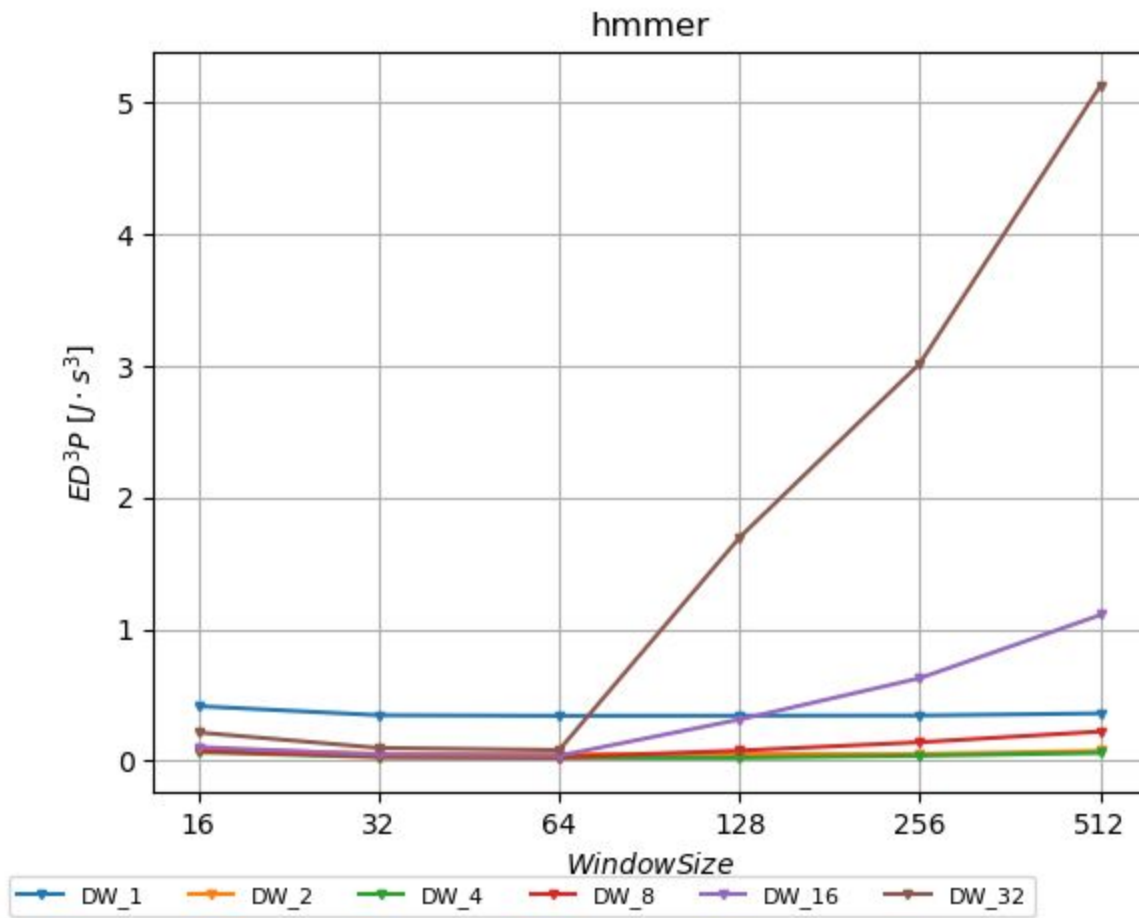
➤ 445.gobmk



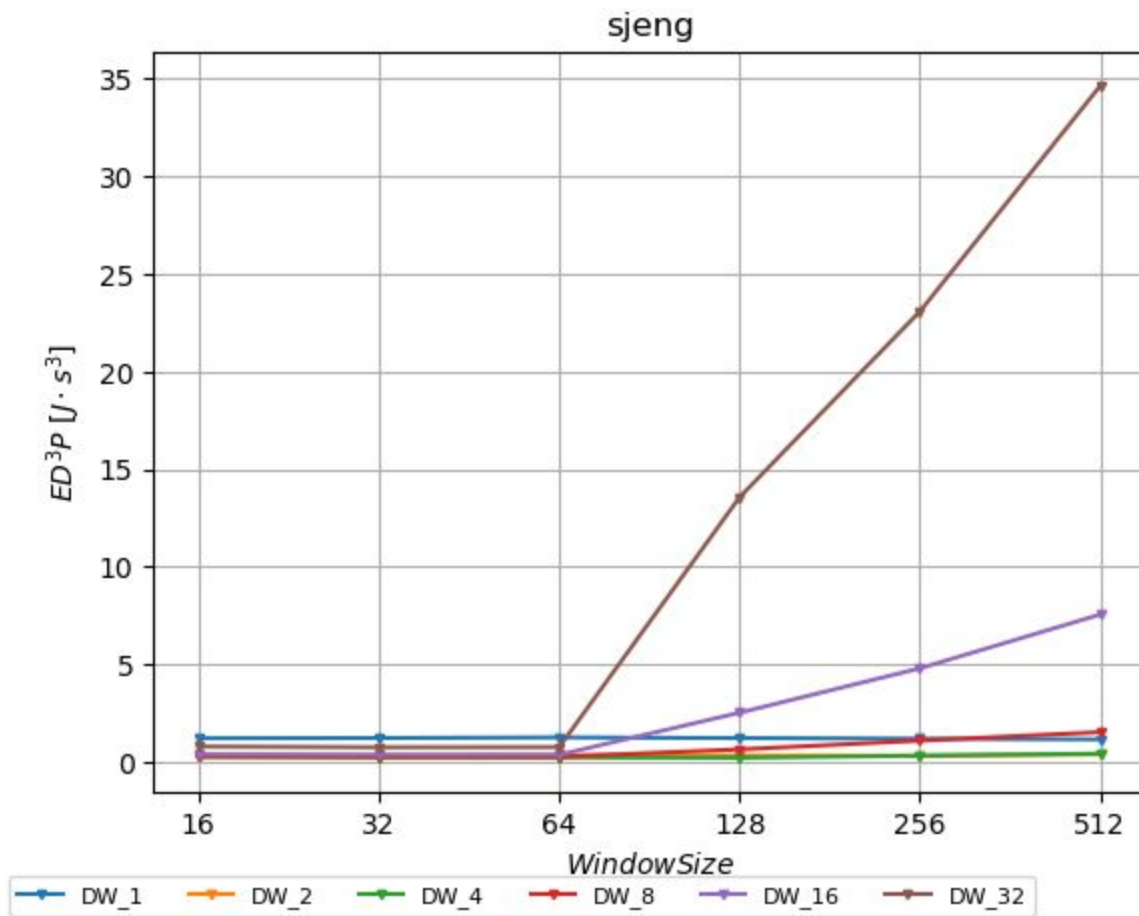
➤ 450.soplex



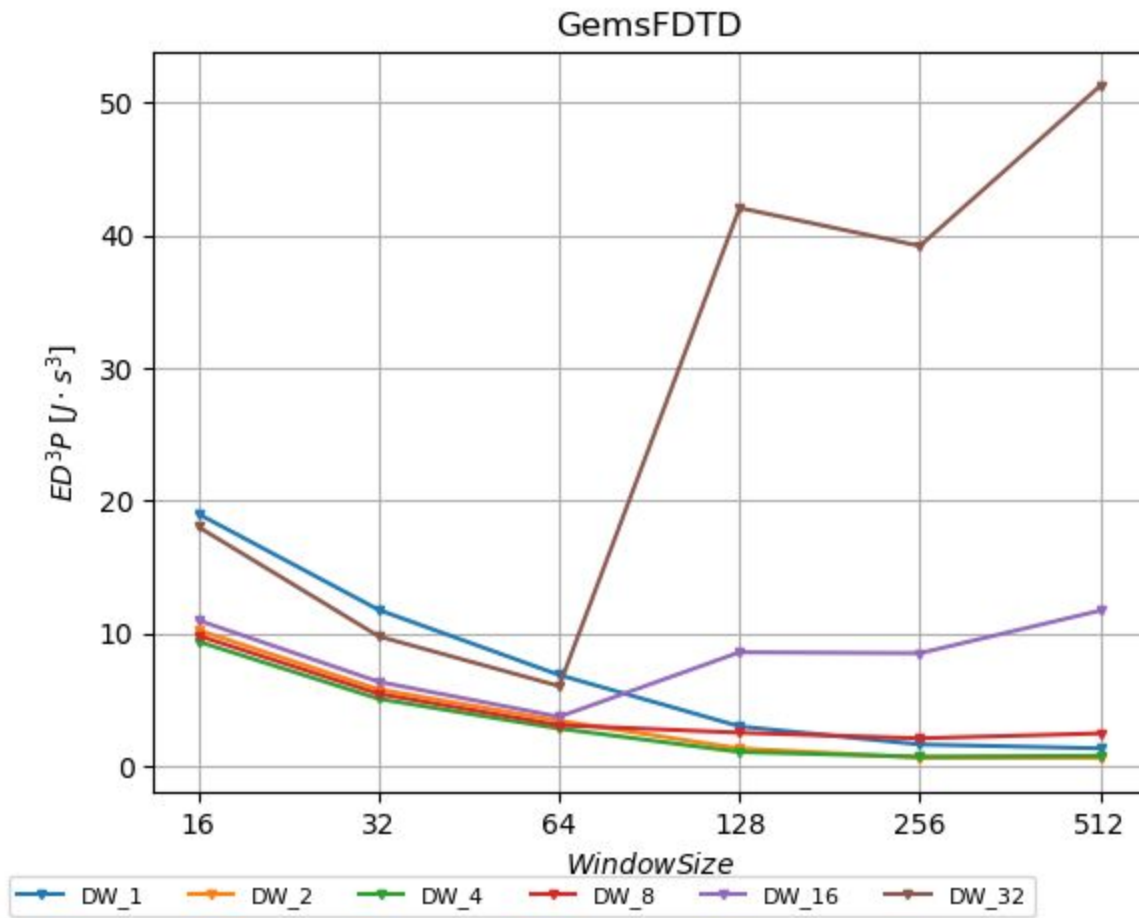
➤ 456.hmm



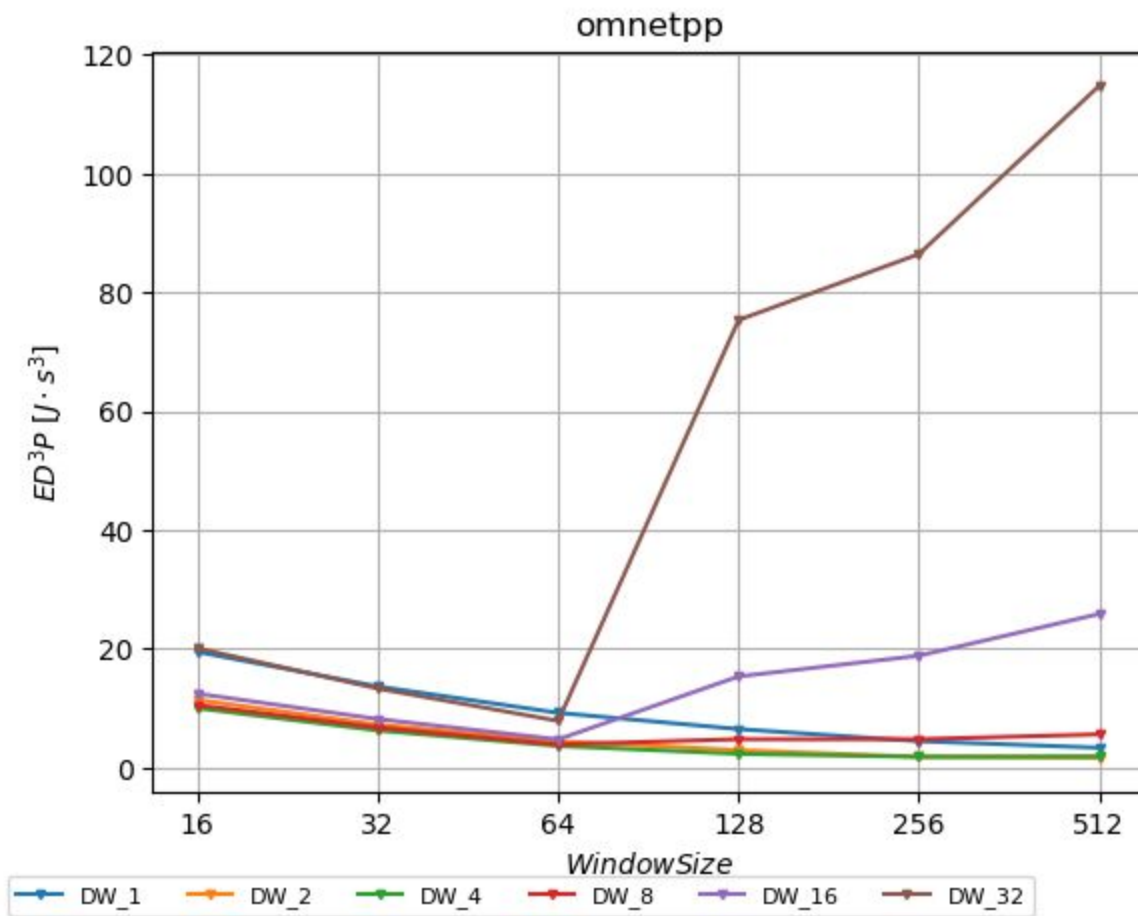
➤ 458.sjeng



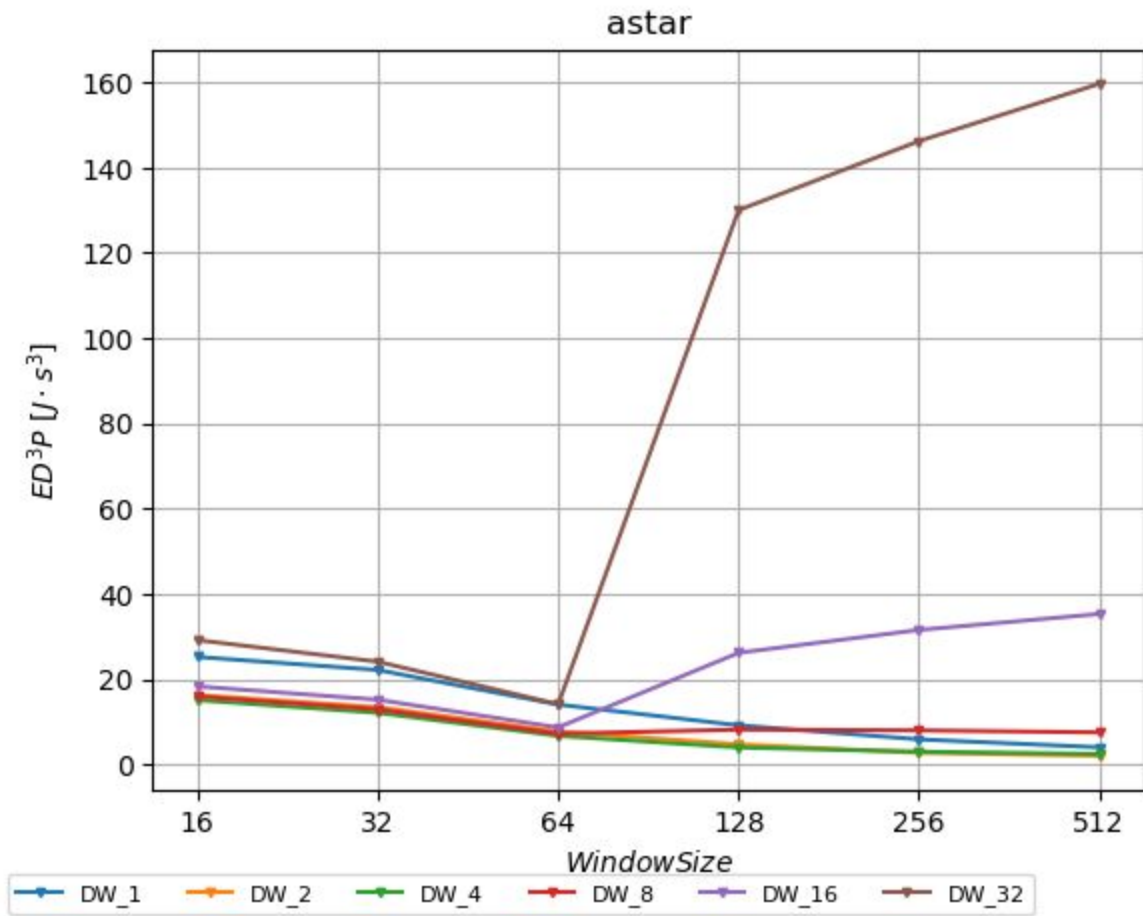
➤ 459.GemsFDTD



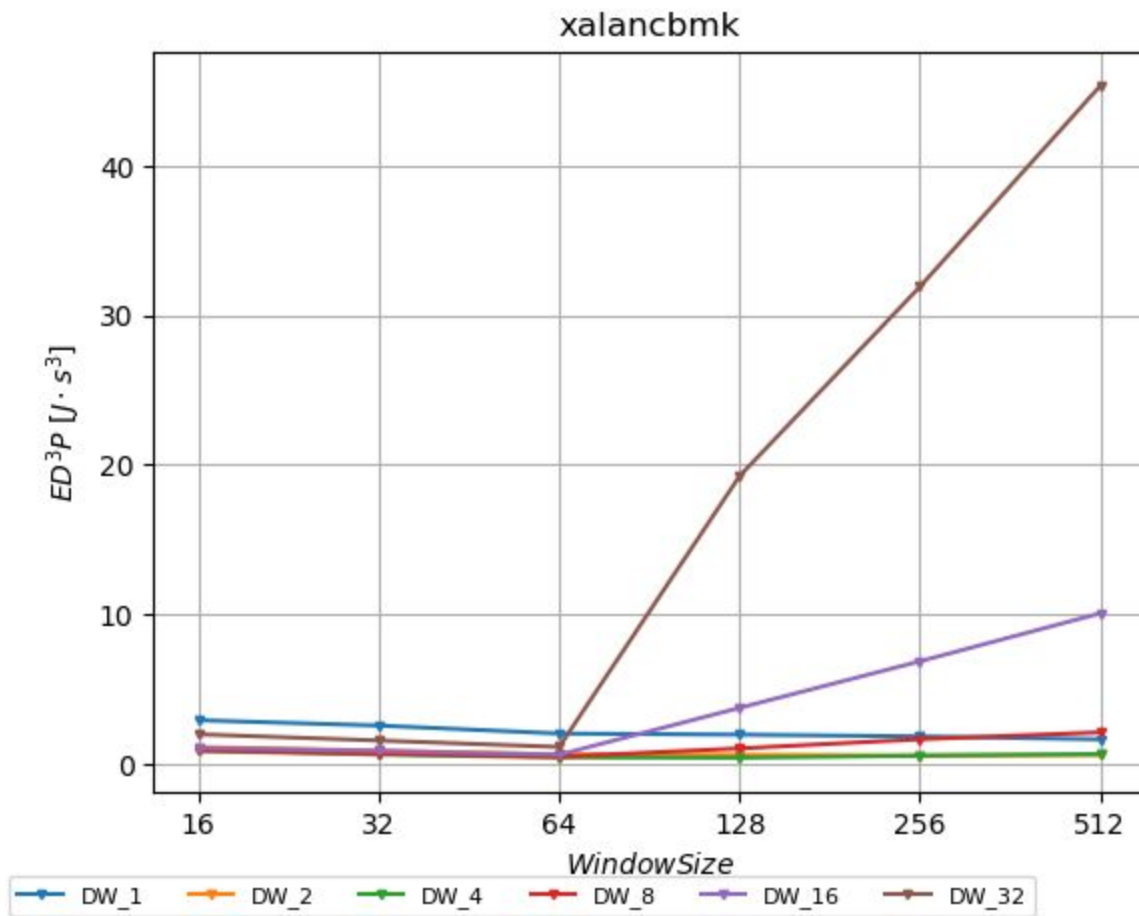
➤ 471.omnetpp



➤ 473.astar

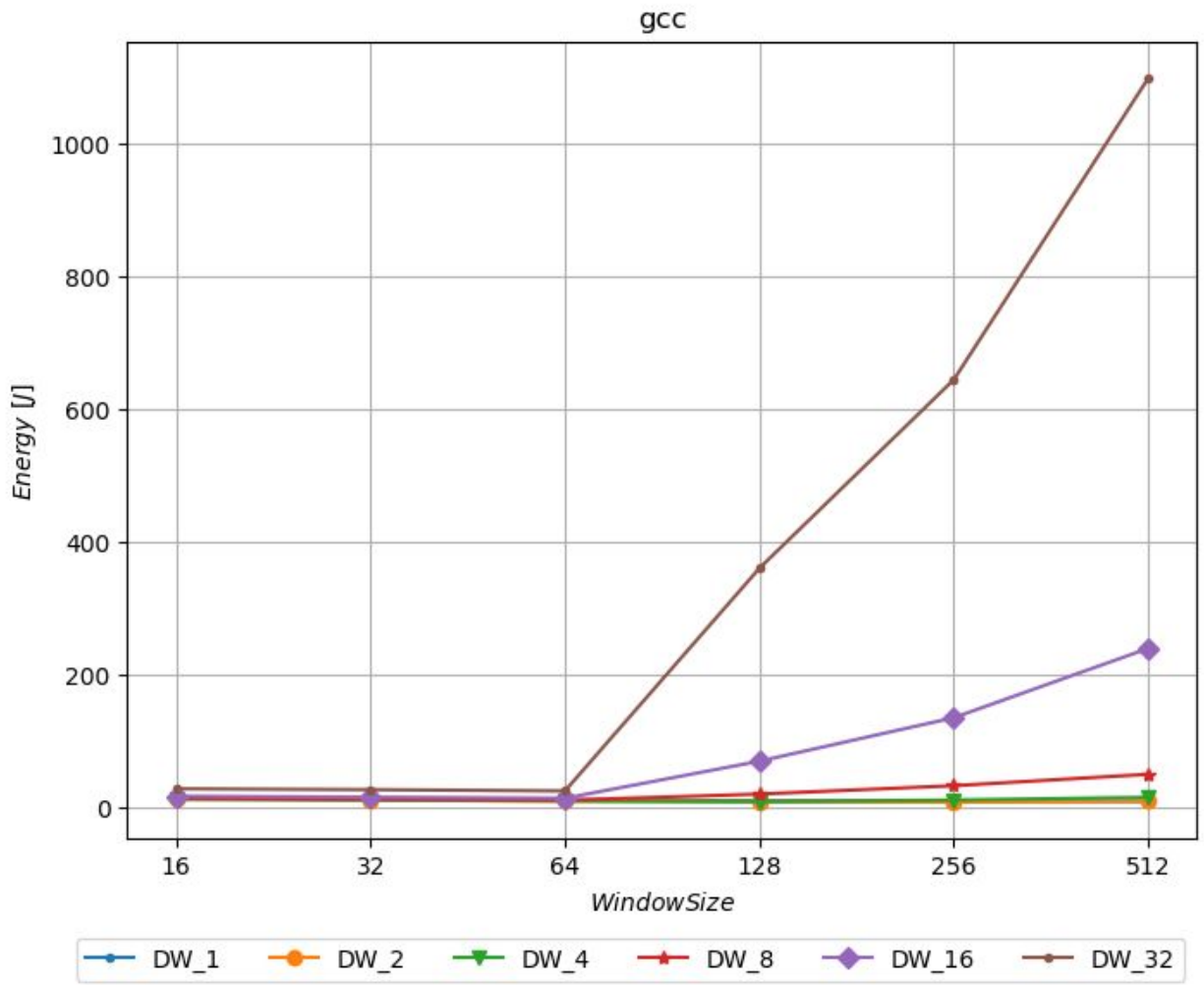


➤ 483.xalancbmk

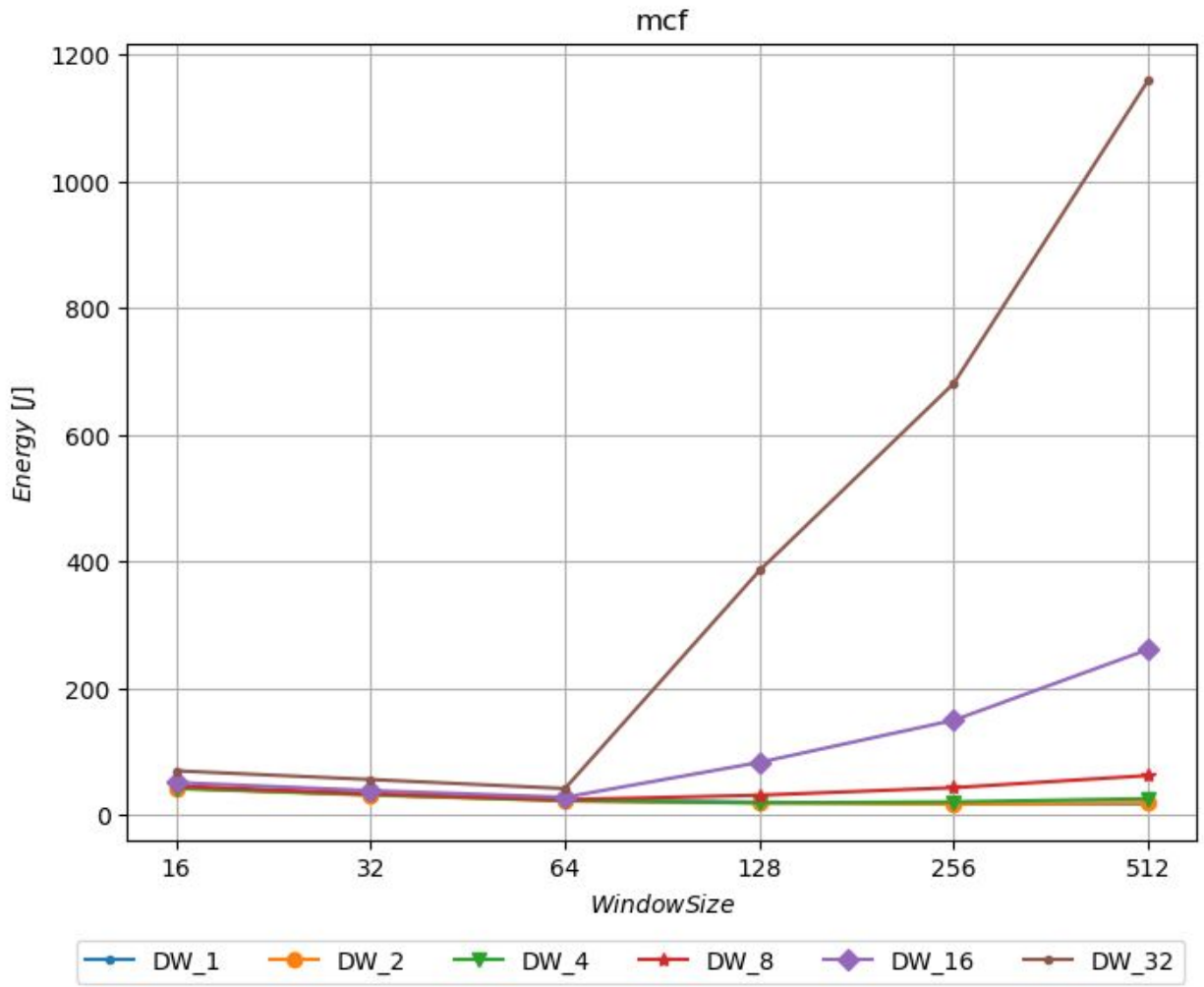


ENERGY

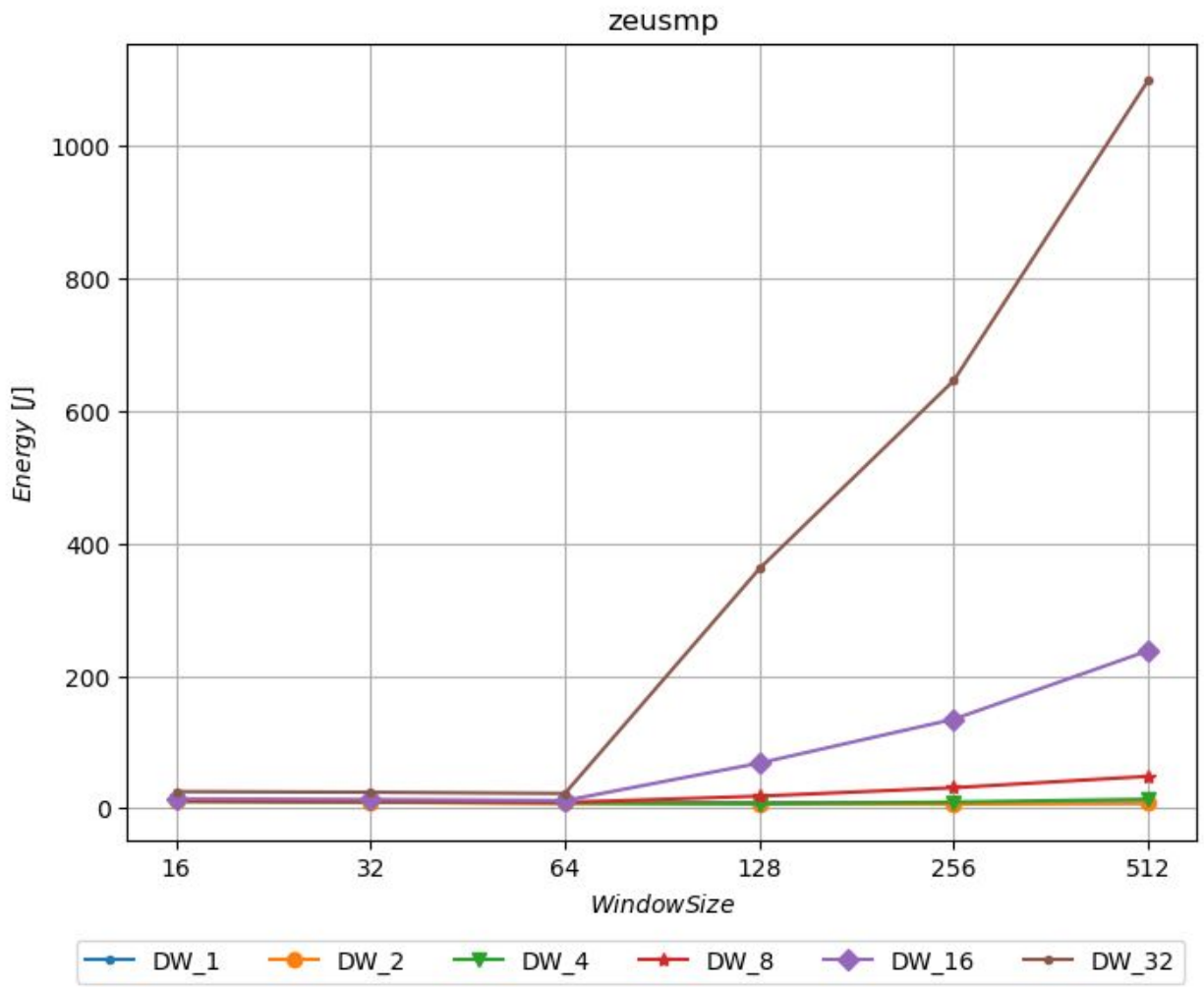
➤ 403.gcc



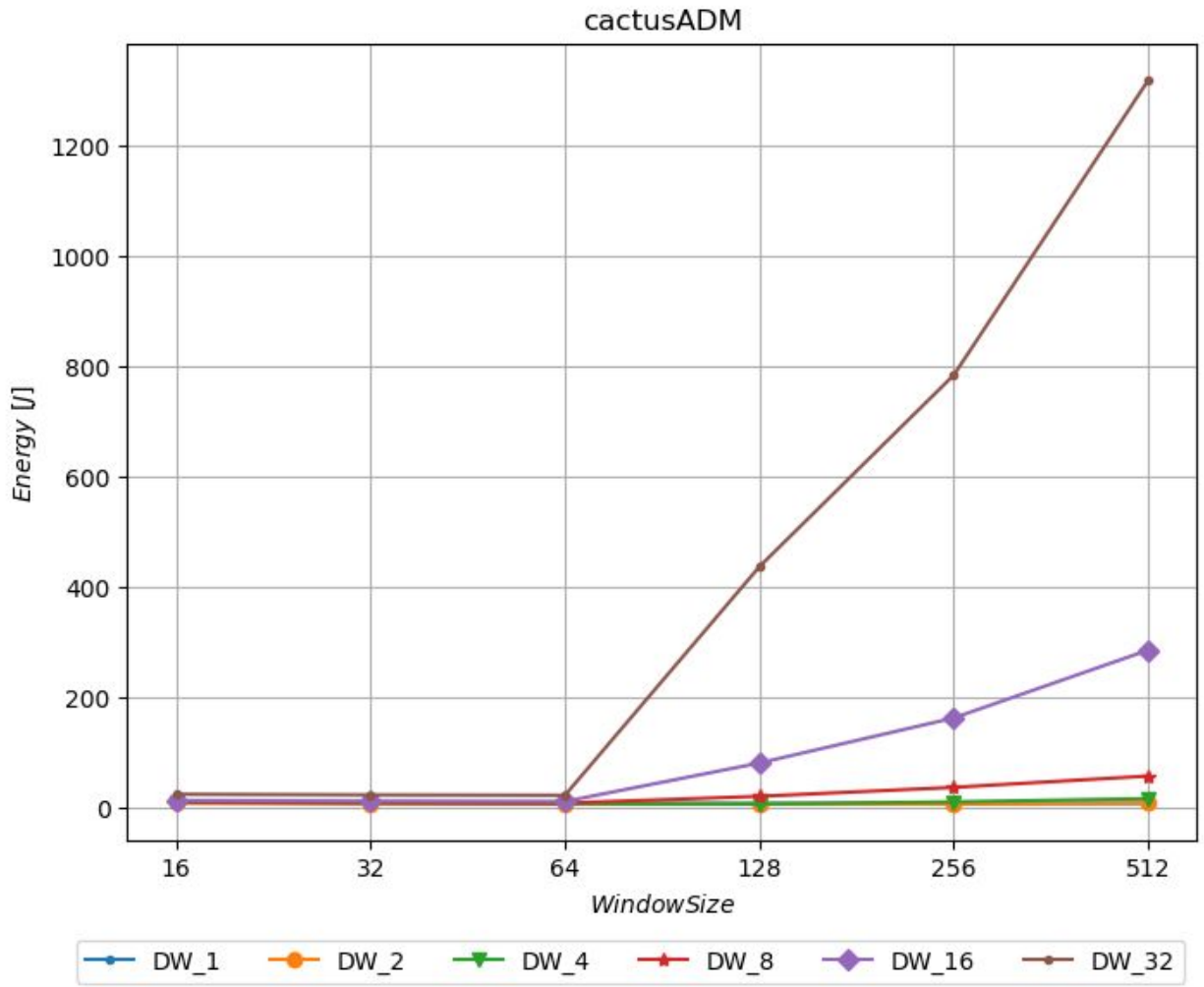
➤ 429.mcf



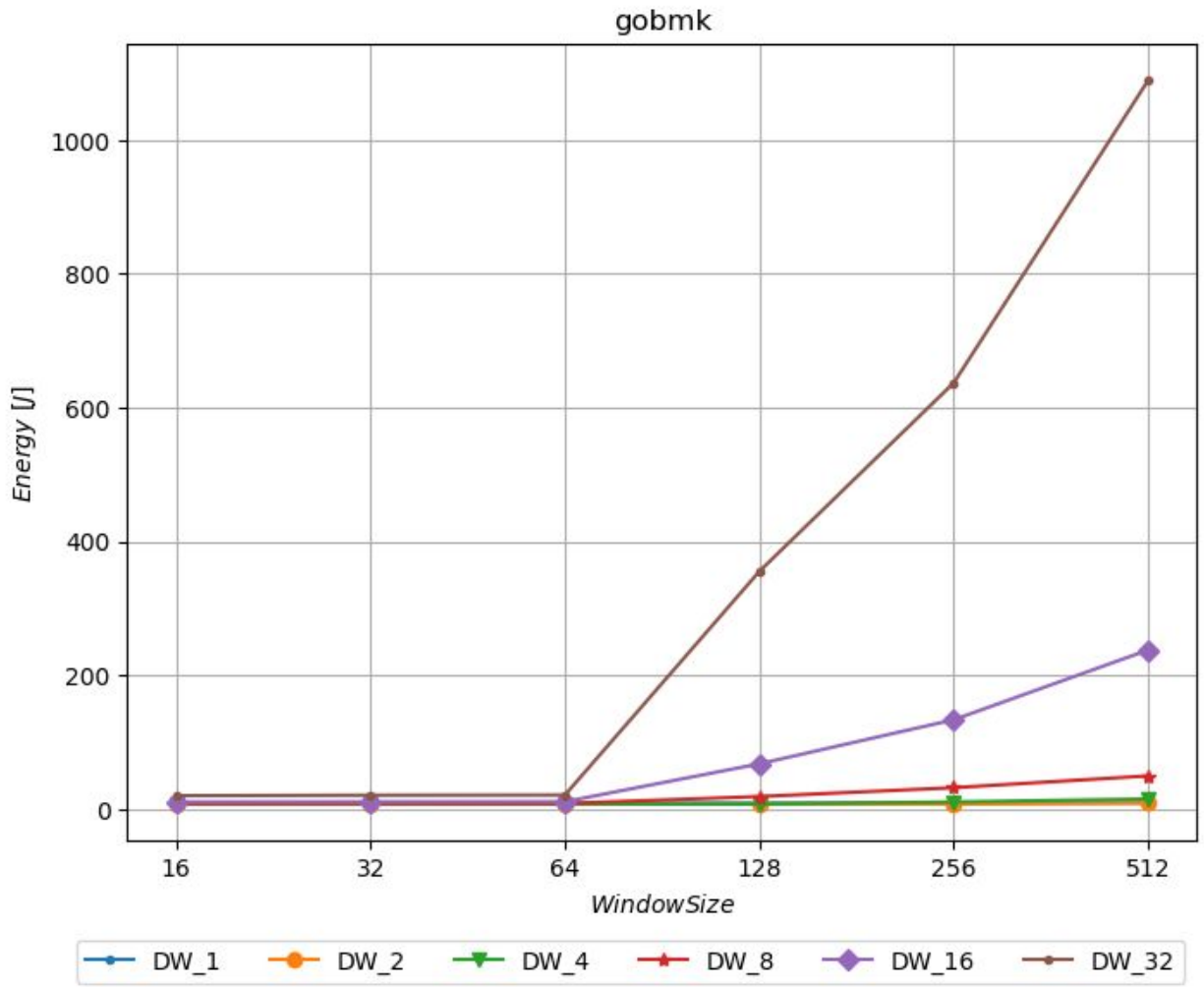
➤ 434.zeusmp



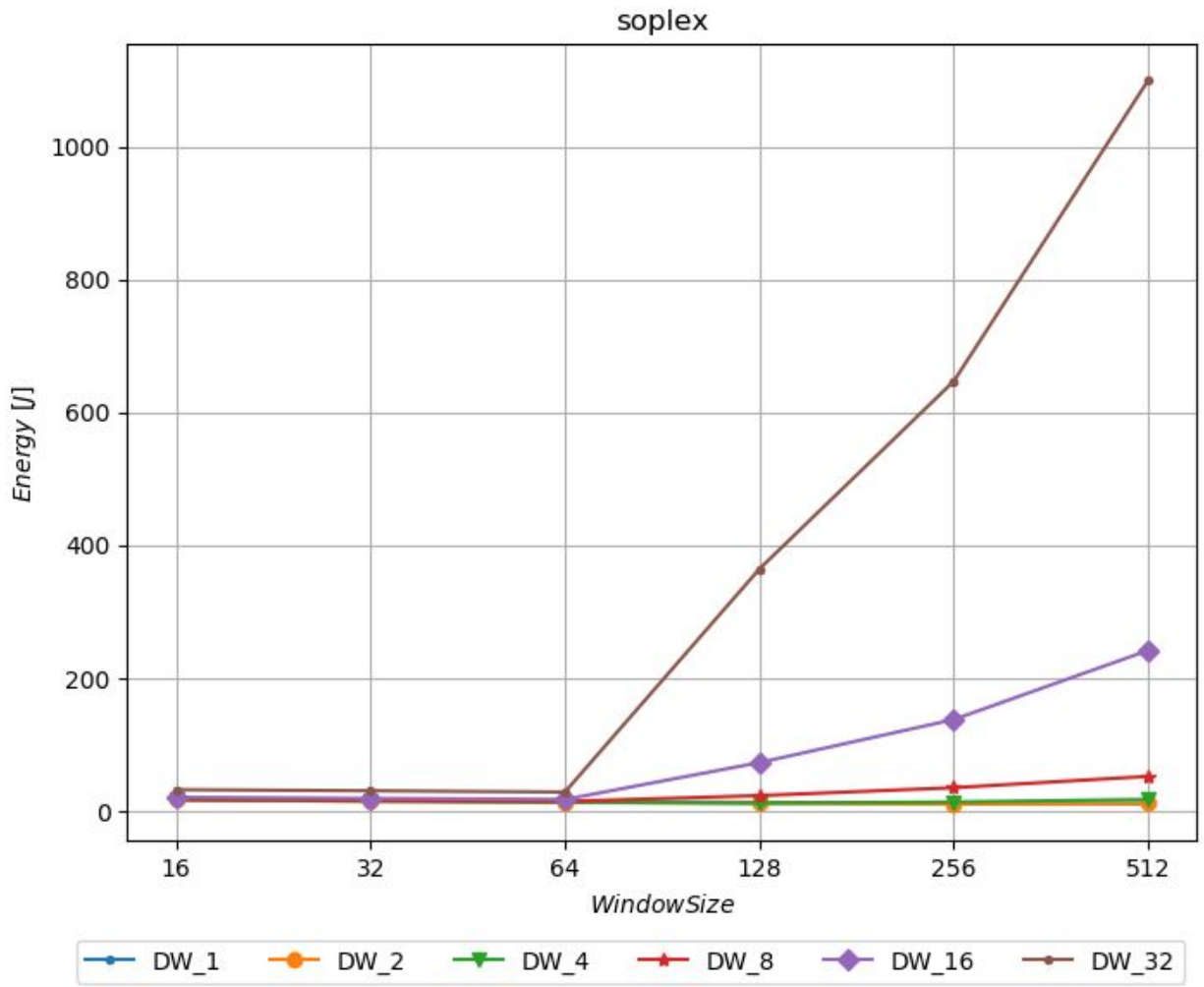
➤ 436.cactusADM



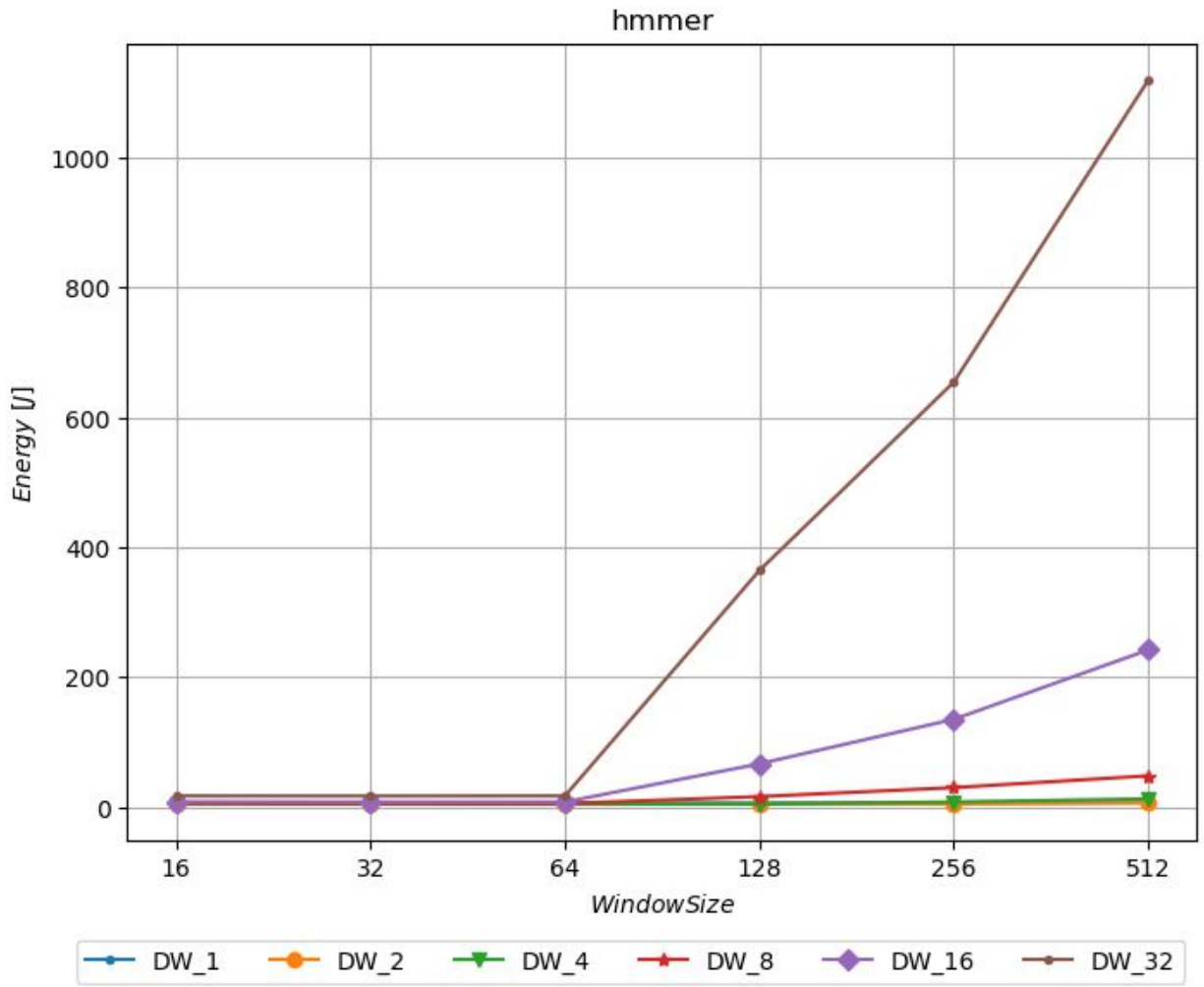
➤ 445.gobmk



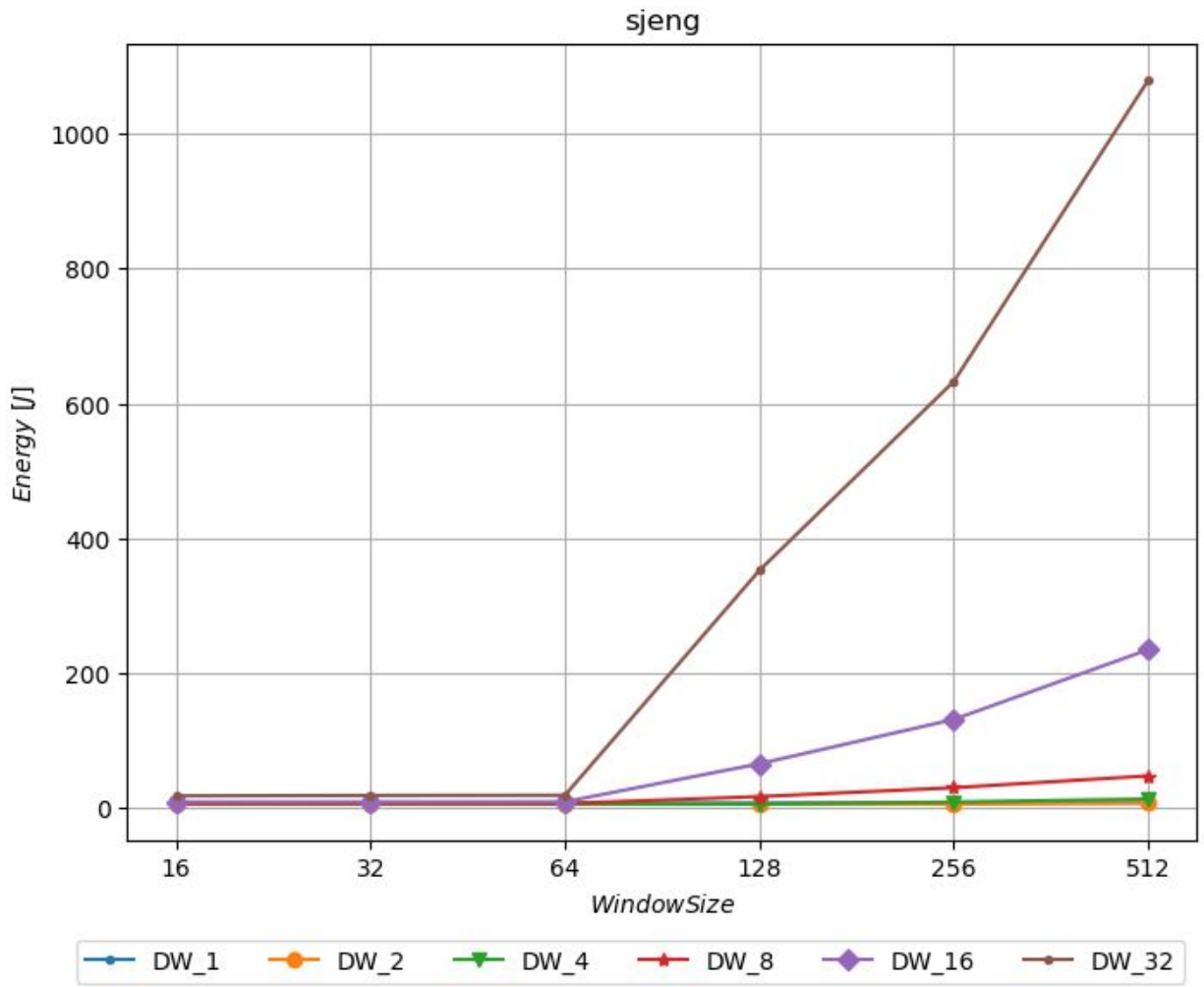
➤ 450.soplex



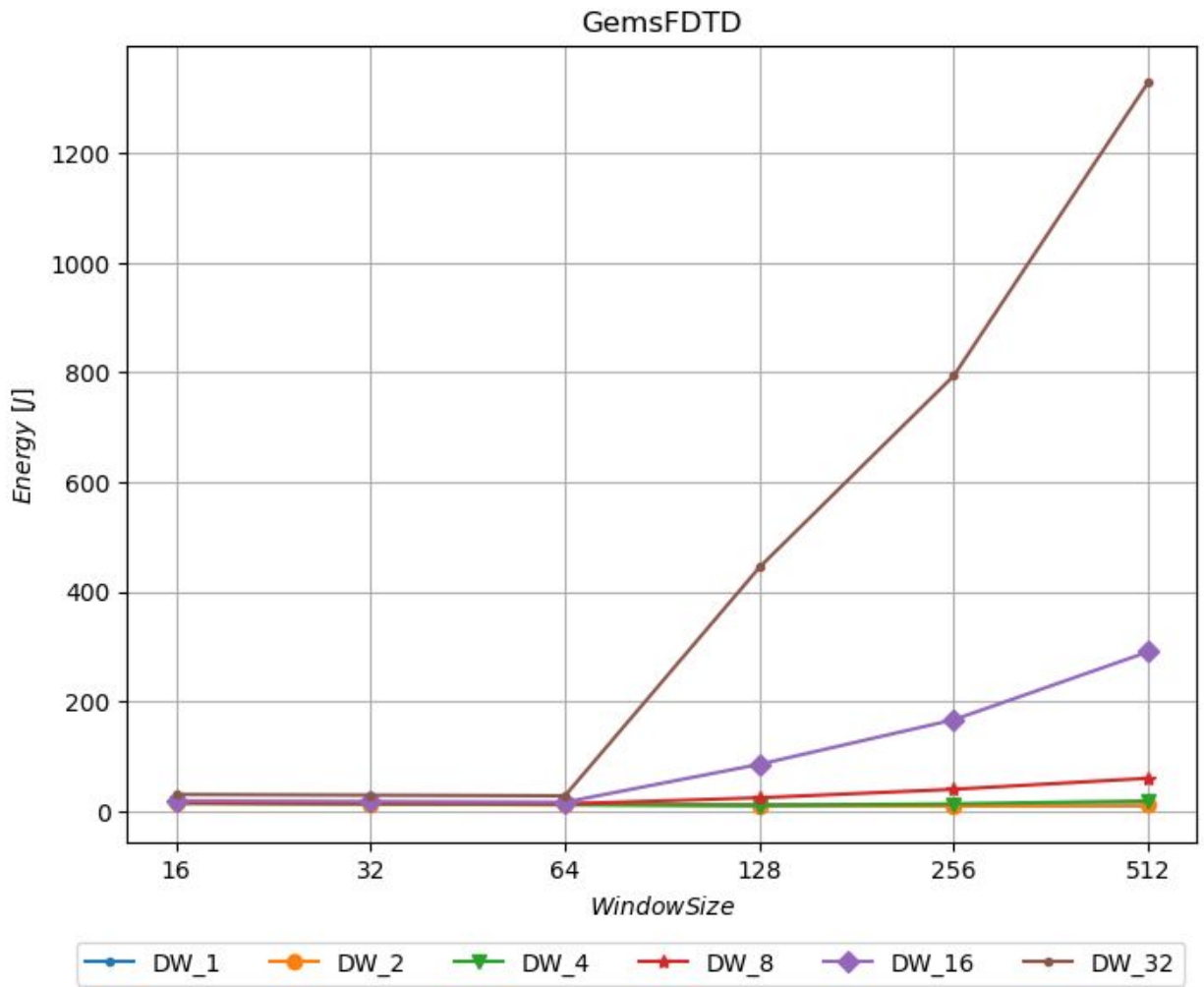
➤ 456.hmmmer



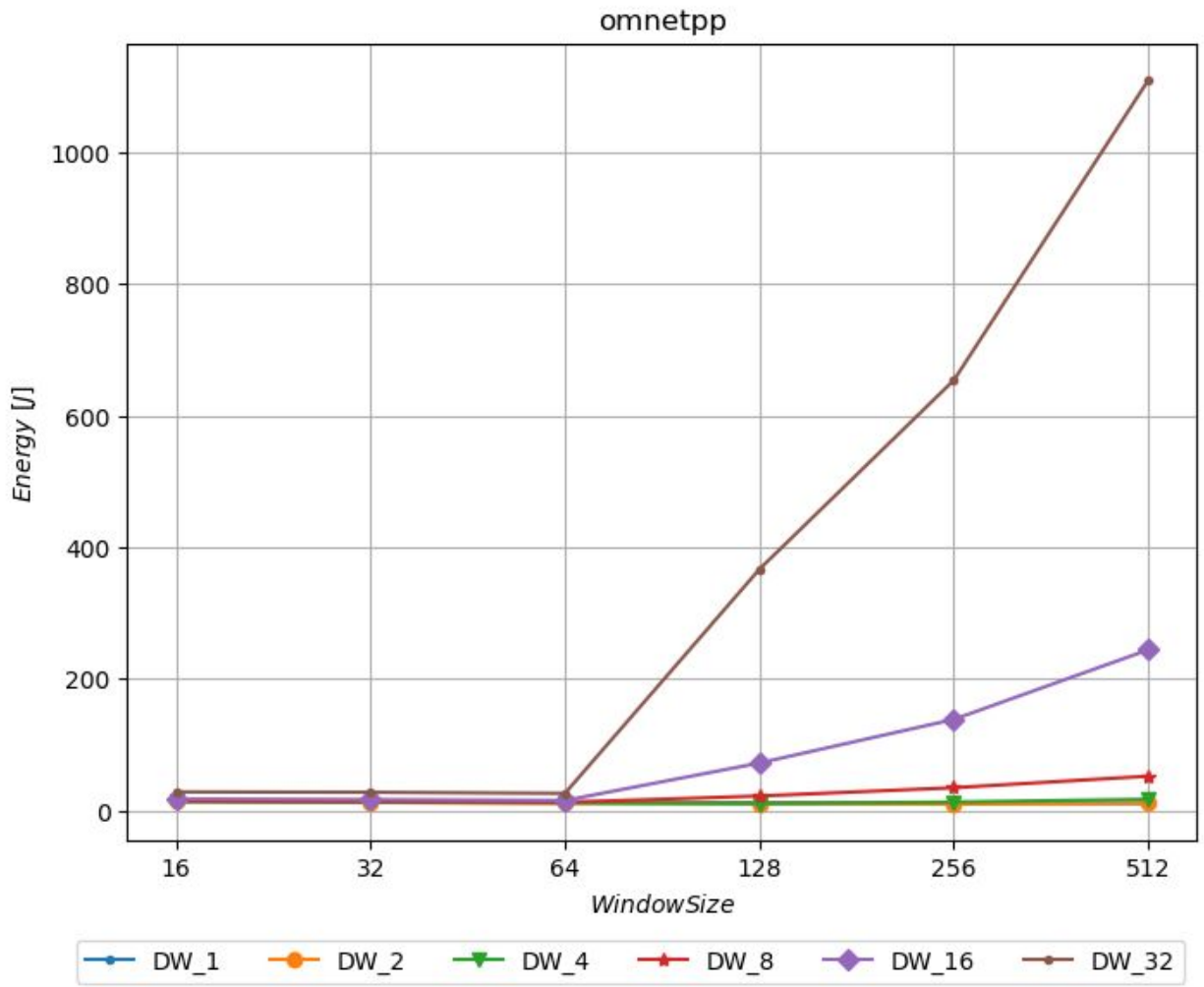
➤ 458.sjeng



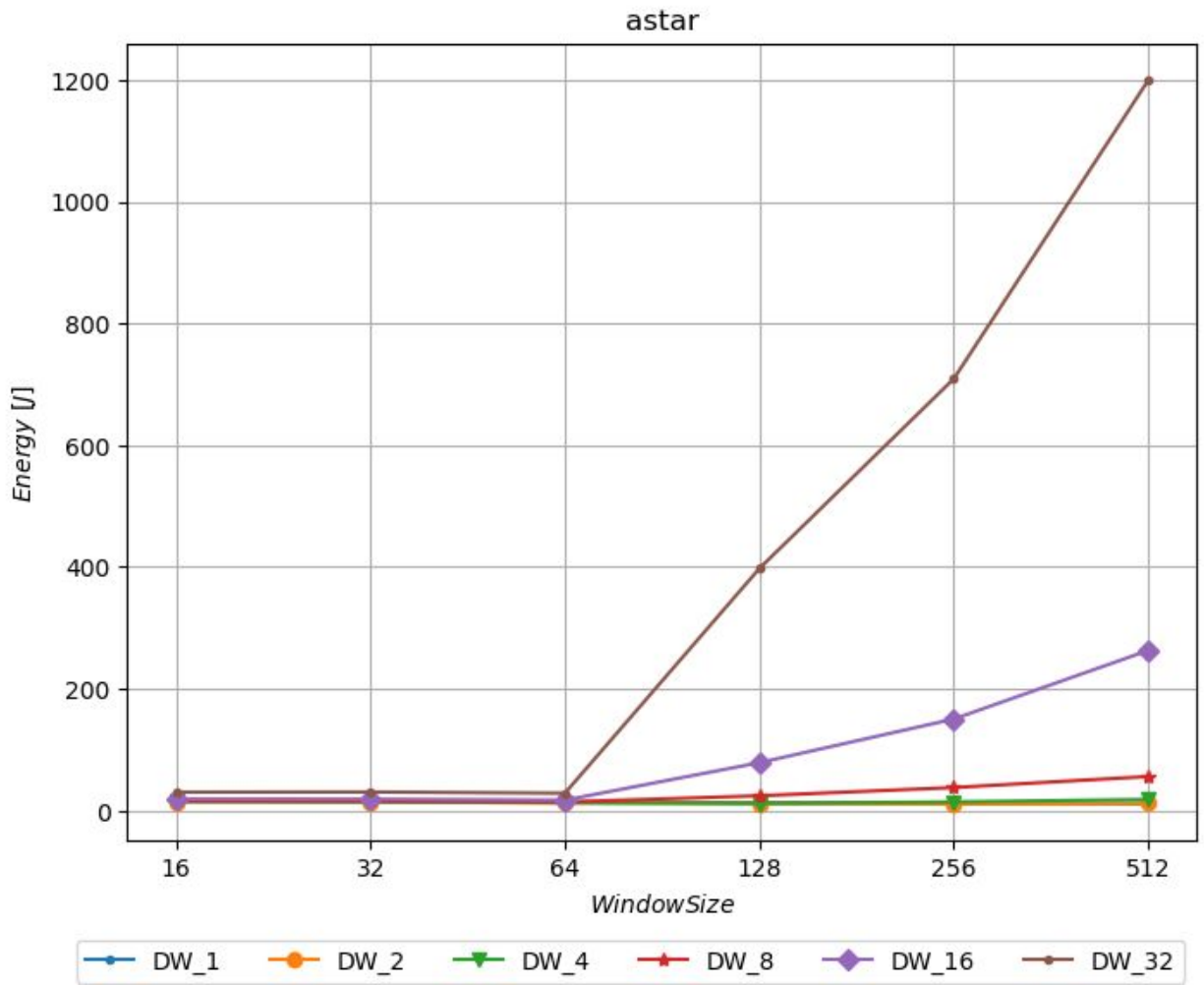
➤ 459.GemsFDTD



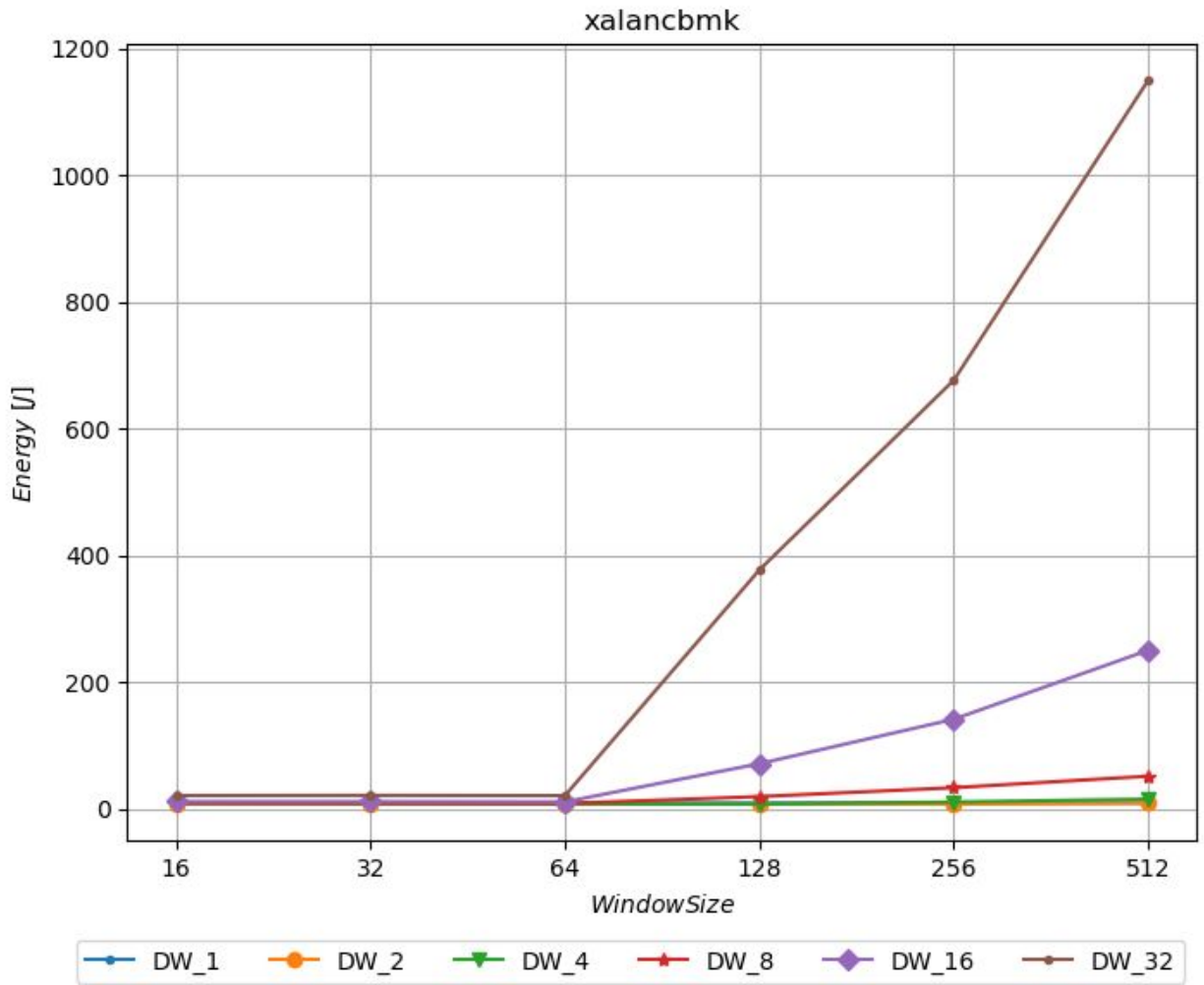
➤ 471.omnetpp



➤ 473.astar

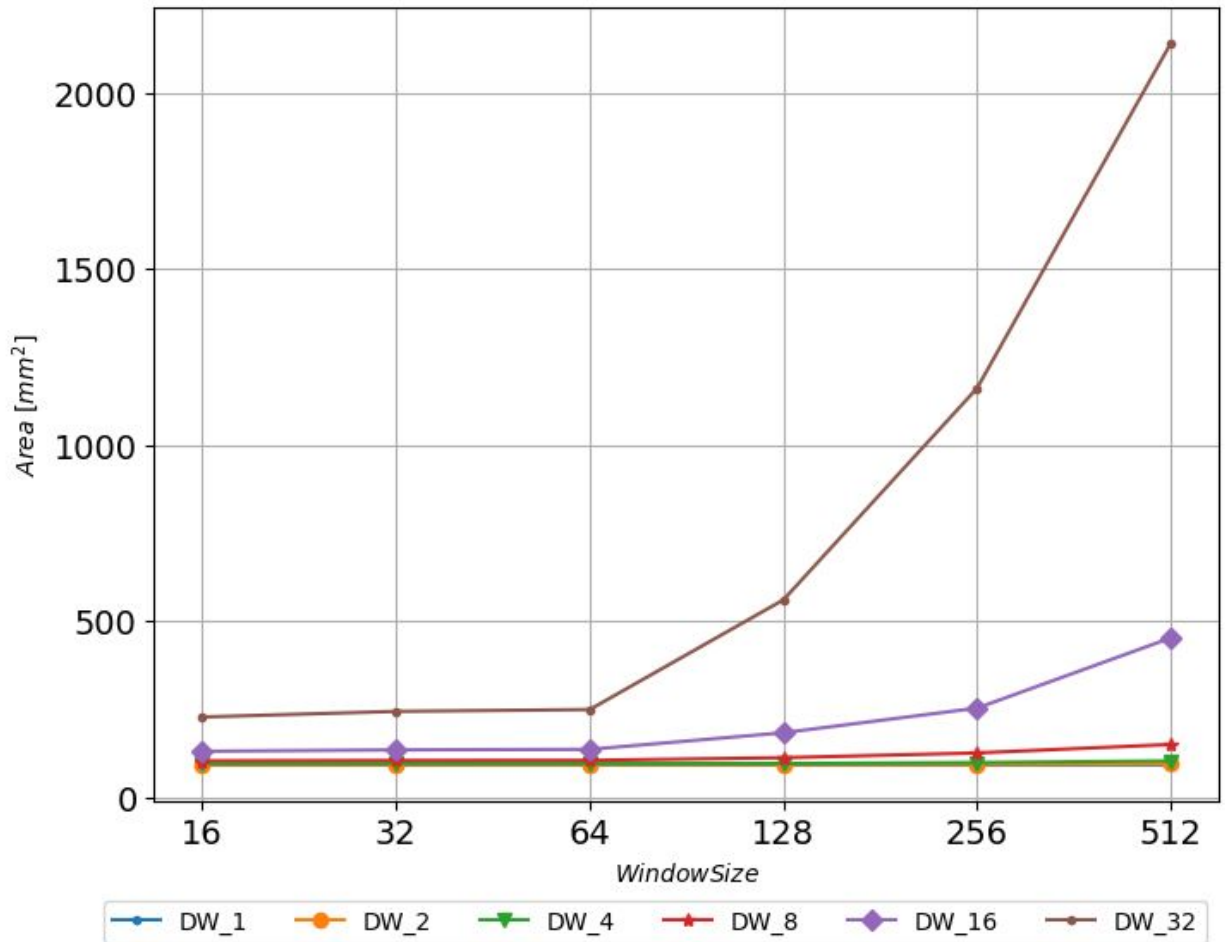


➤ 483.xalancbmk



Area

Εδώ για όλα τα μετροπρογράμματα ήταν πανομοιότυπο το διάγραμμα και είχε ως εξής:



Συμπεράσματα

Καταρχάς είναι εμφανές πως όσο αυξάνεται το dispatch width και το window size, τόσο αυξάνονται και το μέγεθος του τσιπ, καθώς και η κατανάλωση ενέργειας.

Για μικρότερες τιμές του dispatch width, το window size παίζει μικρότερο ρόλο στην μεταβολή των μελετούμενων μεγεθών. Για dispatch size από 16 και πάνω παρατηρούμε πολύ μεγάλη αύξηση.

Ταυτόχρονα, για μεγάλα dispatch width και window size, αν και έχουμε πολύ μεγάλη αύξηση της καταναλώμενης ενέργειας και του μεγέθους του τσιπ, βλέπουμε πως δεν υπάρχει ανάλογη αύξηση της απόδοσης. Αυτό

συμβαίνει εφόσον η αύξηση συμβαίνει στο core αφού το μέγεθος και η κατανάλωση της cache μένουν σταθερά.

iv) *Βρείτε τα αντίστοιχα χαρακτηριστικά (dispatch_width, window_size) για τον επεξεργαστή του προσωπικού σας υπολογιστή ή για κάποιον από τους σύγχρονους επεξεργαστές (π.χ. Broadwell, Skylake, Kabylake).*

Δικαιολογούνται οι τιμές που επιλέξαντε σε αυτά τα συστήματα οι αρχιτέκτονες με βάση τις προσομοιώσεις που εκτελέσατε και τα συμπεράσματα στα οποία καταλήξατε; Θα είχε νόημα να ήταν διαφορετικές (π.χ. μεγαλύτερο window_size); Για ποιό λόγο πιστεύετε δεν κάνανε κάποια άλλη επιλογή;

Ο προσωπικός μου υπολογιστής έχει window size 128 και dispatch width 4. Με βάση την παραπάνω ανάλυση αυτές είναι καλές επιλογές καθώς είναι από τους συνδυασμούς με το ελάχιστο κόστος παραγωγής που προσεγγίζουν την μέγιστη απόδοση των υπόλοιπων συνδυασμών. Δηλαδή αν μειώναμε κάτι από αυτά υπάρχουν περιπτώσεις που η απόδοση θα έπεφτε αρκετά, ενώ εάν αυξανόταν, παρόλο που θα είχαμε καλύτερη απόδοση, δεν θα ήταν τόσο μεγάλη η βελτίωση ενώ θα αυξανόταν το κόστος παραγωγής και άρα και πώλησης.