



ΕΘΝΙΚΟΝ & ΚΑΠΟΔΙΣΤΡΙΑΚΟΝ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**Τίτλος Έργου: «StochSocS: Συστήματα σε Ψηφίδα για Παράλληλη
Στοχαστική Προσομοίωση Βιολογικών Δικτύων στη Βιολογία
Συστημάτων»**

(Αριστεία ΙΙ κωδικός 3828 και Κ.Α. 70/3/12367)

ΠΑΡΑΔΟΤΕΟ 3.1

Υπηρεσία ιστού για στοχαστική προσομοίωση μέσω διαδικτύου

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Δηλώνω υπεύθυνα και εν γνώσει των συνεπειών του νόμου ότι το Παραδοτέο «Υπηρεσία ιστού για στοχαστική προσομοίωση μέσω διαδικτύου» αποτελεί πρωτότυπο έργο και δεν έχουν χρησιμοποιηθεί άλλες πηγές πέρα από αυτές στις οποίες γίνεται αναφορά. Όπου υπάρχει αναπαραγωγή δεδομένων, πινάκων και παραθεμάτων, αυτό δηλώνεται ρητά.

Ηλίας Μανωλάκος



Περιεχόμενα

1	ΕΙΣΑΓΩΓΗ.....	6
1.1	Υπηρεσίες ιστού	6
1.2	Υπάρχουσες υλοποιήσεις για στοχαστικές προσομοιώσεις βιολογικών συστημάτων	6
1.3	Στόχος – Σκοπός.....	7
1.4	Διάρθρωση Παραδοτέου	8
2	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΗΡΕΣΙΑΣ ΙΣΤΟΥ	9
2.1	Προδιαγραφές στοχαστικής προσομοίωση βιολογικών δικτύων	9
2.1.1	Πιστοποίηση (authentication) και εξουσιοδότηση (authorization) του χρήστη	9
2.1.2	Διαχείριση βιομοντέλων σε μορφή SBML.....	9
2.1.3	Τοποθέτηση εργασιών στοχαστικής προσομοίωσης.....	10
2.2	Επισκόπηση Αρχιτεκτονικής.....	11
2.2.1	Κόμβος Εφαρμογής Ιστού (Web Application Node).....	11
2.2.2	Κόμβος Προετοιμασίας και Τοποθέτησης Προσομοιώσεων (Simulation Provisioner)	12
2.2.3	Κόμβος Εκτέλεσης Προσομοιώσεων στο Υλικό FPGA (FPGA Worker).....	13
2.3	Επικοινωνία Κόμβων	13
3	ΔΙΚΤΥΑΚΟΣ ΚΟΜΒΟΣ (FRONTEND).....	16
3.1	Περιβάλλον Εκτέλεσης, Αυτοματοποίηση Εγκατάστασης και Ενεργοποίησης.....	16
3.2	Πλατφόρμα Ανάπτυξης και Υποστηρικτικές Βιβλιοθήκες.....	18
3.3	Μοντέλα, Ελεγκτές Δεδομένων και Όψεις (MVC).....	19
3.3.1	Μοντέλα Δεδομένων.....	20
3.3.2	Ελεγκτές.....	22
4	ΥΠΟΛΟΓΙΣΤΙΚΟΙ ΚΟΜΒΟΙ (BACKEND).....	23
4.1	Εργαλεία ανάπτυξης και Υποστηρικτικές Βιβλιοθήκες.....	23
4.2	Κόμβος Συντονισμού Προσομοιώσεων (Simulation Provisioner).....	24
4.2.1	Simulation Provisioner.....	24
4.2.2	Beanstalk Client	25
4.2.3	Input Parser	25
4.2.4	FPGA Programmer	25
4.2.5	Results Parser	26
4.3	Κόμβος Εκτέλεσης Προσομοιώσεων (FPGA Worker).....	26
4.3.1	FPGAWorker	27
4.3.2	Serial Communicator	27

5	ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	28
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	30

Κατάλογος Εικόνων

Εικόνα 1: Τυπική αρχιτεκτονική μιας υπηρεσίας ιστού	6
Εικόνα 2: Η αρχιτεκτονική της υπηρεσίας ιστού με τα κυριότερα συστατικά μέρη των εφαρμογών	11
Εικόνα 3: Η μεταφορά μηνυμάτων μεταξύ των κόμβων με κατάλληλες ουρές εργασίας.....	14
Εικόνα 4: Το αρχιτεκτονικό μοτίβο MVC της εφαρμογής ιστού	19



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη

ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Κατάλογος Πινάκων

Πίνακας 1: Παράμετροι εργασιών στοχαστικής προσομοίωσης.....	10
Πίνακας 2: Ουρές μηνυμάτων που χρησιμοποιεί η υπηρεσία ιστού	14
Πίνακας 3: Αρχεία δεδομένων κατά την εκτέλεση της κάθε προσομοίωσης	15
Πίνακας 4: Χαρακτηριστικά δικτυακού κόμβου	16
Πίνακας 5: Τα κυριότερα πακέτα λογισμικού του διαδικτυακού κόμβου	17
Πίνακας 6: Οι βοηθητικές βιβλιοθήκες λογισμικού του διαδικτυακού κόμβου.....	18
Πίνακας 7: Οι ιδιότητες του μοντέλου των SBML Bio-Μοντέλων (BioModels).....	20
Πίνακας 8: Οι ιδιότητες του μοντέλου των εργασιών προσομοίωσης (SimulationTasks).....	21
Πίνακας 9: Οι ιδιότητες του μοντέλου των λογαριασμών χρήστη (UserAccounts)	22
Πίνακας 10: Οι ελεγκτές για τα μοντέλα δεδομένων.....	22
Πίνακας 11: Οι υποστηρικτικές βιβλιοθήκες των υπολογιστικών κόμβων	23
Πίνακας 12: Τα αρχεία δεδομένων που παράγει ο Input Parser προς απόστολή στη συσκευή FPGA.....	25
Πίνακας 13: Οι συσκευές I/O για την επικοινωνία με τη συσκευή FPGA	27



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



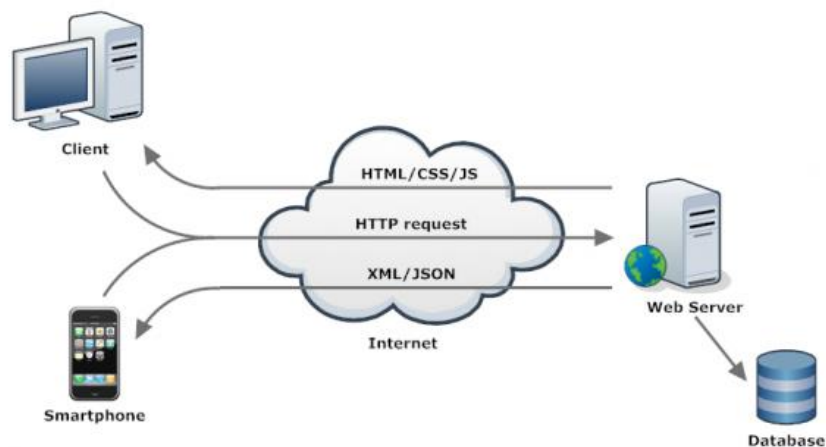
ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

1 ΕΙΣΑΓΩΓΗ

1.1 Υπηρεσίες ιστού

Μια υπηρεσία ιστού είναι μια διεπαφή με κάποια εφαρμογή που επιτυγχάνεται με τη βοήθεια τεχνολογιών διαδικτύου, συνήθως τεχνολογιών ιστού. Τυπικά παραδείγματα υπηρεσιών ιστού είναι:

- ένα αίτημα για την τιμή μιας μετοχής, ή ενός μερίσματος,
- ένα αίτημα για διαθέσιμες θέσεις σε μια συγκεκριμένη πτήση,
- ένα αίτημα για την παραγγελία και αποστολή ενός βιβλίου,
- ένα αίτημα για τη βέλτιστη διαδρομή από μια πόλη σε μια άλλη.



Εικόνα 1: Τυπική αρχιτεκτονική μιας υπηρεσίας ιστού

Μια τυπική αρχιτεκτονική μιας υπηρεσίας ιστού φαίνεται στην Εικόνα 1. Εδώ ένας φυλλομετρητής (browser) που χρησιμοποιεί ο χρήστης στον προσωπικό υπολογιστή του (PC) ή στο έξυπνο κινητό του (smartphone) παράγει HTTP αιτήματα (requests) μέσω του διαδικτύου, προς ένα διακομιστή ιστού (web server) ή προς ένα πρόγραμμα που φέρεται ως διακομιστής ιστού, εφόσον αυτό καταλαβαίνει το πρωτόκολλο HTTP. Τα αιτήματα αποκωδικοποιούνται από τον διακομιστή, ο οποίος στη συνέχεια αλληλεπιδρά με μια συγκεκριμένη εφαρμογή όπως μια βάση δεδομένων και επιστρέφει πίσω στο φυλλομετρητή του χρήστη τα αποτελέσματα σε κατάλληλη μορφή, χρησιμοποιώντας καθορισμένα στάνταρ όπως η HTML5 (HTML / CSS / JS) για την απεικόνιση των δεδομένων που μεταφέρονται σε επίσης στάνταρ μορφή αναπαράστασης τύπου XML ή JSON.

1.2 Υπάρχουσες υλοποιήσεις για στοχαστικές προσομοιώσεις βιολογικών συστημάτων

Αν και υπάρχουν αμέτρητες υλοποιήσεις υπηρεσιών ιστού για τις εφαρμογές που αναφέρθηκαν παραπάνω, είναι λίγες αυτές που αφορούν προσομοιώσεις επιστημονικού ενδιαφέροντος και απόσο γνωρίζουμε ελάχιστες που να αφορούν προσομοιώσεις βιολογικών συστημάτων. Δύο τέτοιες υπηρεσίες που γνωρίζουμε είναι το Condor-COPASI από το Πανεπιστήμιο του Μάντσεστερ [1] και το StochSS από το Πανεπιστήμιο της Καλιφόρνια, Σάντα Μπάρμπαρα [2].

Το Condor-COPASI είναι μια δικτυακή διεπαφή που ενσωματώνει το εργαλείο μοντελοποίησης και προσομοίωσης βιολογικών συστημάτων COPASI [3] μαζί με το περιβάλλον Condor [4] που επιτρέπει την υψηλή υπολογιστική ρυθμαπόδοση (High Throughput Computing – HTC) εφαρμογών μέσω της κατανεμημένης παραλληλίας και της εκτέλεσης πολλαπλών εργασιών (batch) ταυτόχρονα πάνω σε ένα σύμπλεγμα (cluster) υπολογιστών. Με αυτόν τον τρόπο το μονο-νηματικό COPASI μπορεί να λειτουργήσει ουσιαστικά παράλληλα επιταχύνοντας σημαντικά τις προσομοιώσεις που εκτελεί ο χρήστης σε σχέση με αν το χρησιμοποιούσε τοπικά στον υπολογιστή του. Η ομάδα που το ανέπτυξε διαθέτει μάλιστα και μια πιλοτική εγκατάσταση του για όσους χρήστες επιθυμούν να το χρησιμοποιήσουν κατόπιν συνενόησης και δημιουργίας σχετικού λογαριασμού στο σύστημα.

Το StochSS είναι μια παρόμοια δικτυακή διεπαφή που ενσωματώνει ένα εργαλείο μοντελοποίησης υλοποιημένο με τεχνολογίες ιστού HTML5 μαζί με τον προσομοιωτή στοχαστικών προσομοιώσεων StochKit [5] που έχει αναπτύξει η ίδια ομάδα εργασίας. Το πλεονέκτημα του StochKit είναι το γεγονός ότι επιτρέπει το ίδιο την πολύ-νηματική στοχαστική προσομοίωση χωρίς να χρειάζεται ένα έξτρα περιβάλλον όπως το Condor για να το επιτύχει. Επίσης διαθέτει πλήθος άλλων λειτουργιών για εύρεση και βελτιστοποίηση των στοχαστικών παραμέτρων, την επίλυση διαφορικών εξισώσεων για μοντέλα που δε μπορούν να προσομοιωθούν στοχαστικά, την αυτόματη εγκατάσταση σε κάποιο υπολογιστικό νέφος (cloud) που έχει πρόσβαση ο χρήστης κ.α. Το κύριο μειονέκτημα του είναι πως δε διατίθεται μια έτοιμη, έστω πιλοτική, εγκατάσταση που να μπορεί να αποκτήσει πρόσβαση ο χρήστης μέσω ενός λογαριασμού, είτε δωρεάν είτε ακόμα και με χρέωση, ώστε να μη χρειάζεται να το εγκαταστήσει ο ίδιος τοπικά ή σε κάποιο νέφος εφόσον δε διαθέτει τις απαιτούμενες τεχνικές γνώσεις για να επιτευχθεί αυτό.

1.3 Στόχος – Σκοπός

Ο στόχος και σκοπός της Ε.Ε.3 είναι να σχεδιαστεί και να αναπτυχθεί μια πρωτότυπη πιλοτική υπηρεσία ιστού που να επιτρέπει σε απομακρυσμένους χρήστες να υποβάλουν βιομοντέλα βιολογικών δικτύων τα οποία βρίσκονται σε SBML (Systems Biology Markup Language) [6] μορφή, μαζί με επιθυμητές παραμέτρους για στοχαστική προσομοίωση αυτών πάνω στα συστήματα σε ψηφίδα (System on Chip – SoC) υλοποιημένα για συσκευές FPGA που αναπτύξαμε για την Ε.Ε.1. Τα υποβληθέντα αρχεία SBML θα πρέπει να αναλύθούν και στη συνέχεια να ξεκινήσουν στοχαστικές προσομοιώσεις στο FPGA υλικό, είτε σε SSIP (Single Simulation in Parallel) είτε σε MSIP (Multiple Simulations In Parallel) τύπο λειτουργίας. Μετά την ολοκλήρωση της προσομοίωσης, ο διακομιστής (server) θα πρέπει να επιστρέφει πίσω τα αποτελέσματα στους χρήστες για περαιτέρω ανάλυση.

Ένας τέτοιος διακομιστής δεν υπήρχε μέχρι σήμερα και αποτελεί παγκόσμια πρωτοτυπία και πολύ χρήσιμη υπηρεσία για τη επιστημονική κοινότητα της συστημικής βιολογίας καθώς θα επιτρέπει την αποδοτική και επιταχυνόμενη στοχαστική προσομοίωση αληθινών βιομοντέλων σε μορφή SBML χωρίς ο χρήστης να χρειαστεί να ρυθμίσει, πόσο μάλλον να διαθέτει γρήγορους πολύ-νηματικούς επεξεργαστές ή σύμπλεγμα αυτών ή κατάλληλο SoC στοχαστικής προσομοίωσης πάνω στο FPGA υλικό. Όπως δείξαμε και στη Δράση 1.3, τα SoC στοχαστικής προσομοίωσης που σχεδιάσαμε και υλοποιήσαμε για συσκευές FPGA που εκτελούν προσομοιώσεις βιολογικών συστημάτων πολύ πιο αποδοτικά από οποιοδήποτε άλλο λογισμικό προσομοίωσης όπως π.χ το COPASI.

1.4 Διάρθρωση Παραδοτέου

Στις ενότητες που ακολουθούν δίνεται έμφαση στην τεχνική περιγραφή της υπηρεσίας ιστού που αναπτύχθηκε με σκοπό τη στοχαστική προσομοίωση, μέσω διαδικτύου, βιολογικών δικτύων πάνω στα SoCs (System on Chip) για συσκευές FPGA που αναπτύχθηκαν στα πλαίσια της Ε.Ε.1. Στο συγκεκριμένο παραδοτέο θα προσπαθήσουμε να μην αναφερθούμε σε λεπτομέρειες που αναφέρονται αναλυτικά στα υπόλοιπα οπότε και ο αναγνώστης μπορεί να ανατρέχει σε αυτά σε περίπτωση που έχει κάποια απορία σχετικά με το υλικό στο οποίο εκτελούνται τελικά οι προσομοιώσεις που τοποθετεί ο χρήστης στο σύστημα.

Αρχικά γίνεται αναφορά στην γενικότερη αρχιτεκτονική της υπηρεσίας ιστού, των κόμβων που περιλαμβάνει και πως αυτοί επικοινωνούν μεταξύ τους, ενώ στη συνέχεια ακολουθεί η πλήρης περιγραφή αυτών, το περιβάλλον εκτέλεσης και ανάπτυξης τους καθώς και ο τρόπος λειτουργίας τους. Πιο συγκεκριμένα η διάρθρωση του παραδοτέου είναι η εξής:

Ενότητα 2: Παρουσιάζεται η αρχιτεκτονική της υπηρεσίας ιστού με προδιαγραφές για στοχαστική προσομοίωση βιολογικών δικτύων πάνω σε κόμβους που επικοινωνούν μεταξύ τους μέσω μηνυμάτων.

Ενότητα 3: Προδιαγράφεται ο δικτυακός κόμβος (frontend) με τον οποίο αλληλεπιδρά ο χρήστης της υπηρεσίας ιστού, το περιβάλλον εκτέλεσης στο οποίο λειτουργεί, η πλατφόρμα ανάπτυξης με τις υποστηρικτικές της βιβλιοθήκες καθώς και τα μοντέλα δεδομένων που διαχειρίζονται οι ελεγκτές τους.

Ενότητα 4: Αναλύονται οι υπολογιστικοί κόμβοι (backend) για το συντονισμό και την εκτέλεση των προσομοιώσεων πάνω στο κατάλληλο FPGA SoC και πως αυτοί αναπτύχθηκαν με κατάλληλα εργαλεία ανάπτυξης και βοηθητικές βιβλιοθήκες

Ενότητα 5: Στην τελευταία ενότητα εκθέτουμε τα συμπεράσματα που προέκυψαν από την παρούσα ερευνητική εργασία, παραθέτουμε τους προβληματισμούς μας και τις πιθανές μελλοντικές μας ενέργειες.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΗΡΕΣΙΑΣ ΙΣΤΟΥ

2.1 Προδιαγραφές στοχαστικής προσομοίωση βιολογικών δικτύων

Η υπηρεσία ιστού μέσω διαδικτύου που υλοποιήσαμε έχει στόχο να επιτρέπει σε απομακρυσμένους χρήστες να εκτελούν στοχαστικές προσομοιώσεις βιολογικών συστημάτων πάνω στο υλικό που αναπτύξαμε για την Ε.Ε1 χωρίς να χρειάζεται να γνωρίζουν πως αυτό λειτουργεί ή να έχουν απευθείας πρόσβαση σε αυτό. Αποτέλεσμα αυτών των δύο στόχων είναι η μεγάλη ευκολία χρήσης, ακόμα και από άτομα που δεν διαθέτουν προχωρημένες γνώσεις χρήσης υπολογιστικών συστημάτων καθώς και η αυξημένη ασφάλεια του συστήματος από λανθασμένους χειρισμούς ή και υπερβολικό χρόνο χρήσης.

Για να επιτευχθούν αυτοί οι στόχοι σχεδιάσαμε και υλοποιήσαμε 3 κύριες λειτουργίες που σχετίζονται με α) την πιστοποίηση (authentication) και εξουσιοδότηση (authorization) του χρήστη στο σύστημα, β) τη διαχείριση βιομοντέλων που υποβάλλονται (upload) σε μορφή SBML και γ) την τοποθέτηση εργασιών στοχαστικής προσομοίωσης στο σύστημα με τις επιθυμητές παραμέτρους. Στη συνέχεια αυτής της ενότητας προδιαγράφουμε τις 3 αυτές κύριες λειτουργίες που παρέχονται στο χρήστη.

2.1.1 Πιστοποίηση (authentication) και εξουσιοδότηση (authorization) του χρήστη

Οι δύο αυτές λειτουργίες εξασφαλίζουν ότι στην υπηρεσία έχουν πρόσβαση μόνο διαπιστευμένοι χρήστες που διαθέτουν έναν λογαριασμό (account) στο σύστημα. Ο χρήστης πρέπει αρχικά να κάνει εγγραφή (registration) σε αυτό ώστε να δημιουργηθεί ο λογαριασμός του. Κατά την εγγραφή του χρήστη στο σύστημα ζητούνται τα εξής στοιχεία:

- Όνομα χρήστη (Username)
- Συνθηματικό χρήστη (Password)
- Λογαριασμός ηλεκτρονικού ταχυδρομίου (E-Mail)

Στη συνέχεια ο διαχειριστής τους συστήματος αφού ειδοποιηθεί για την εγγραφή μπορεί να προχωρήσει στην ενεργοποίηση ή μη του λογαριασμού αυτού. Το βήμα ενεργοποίησης υπάρχει γιατί η υπηρεσία στην παρούσα φάση λειτουργεί πιλοτικά και διαθέτει μόνο μια συσκευή υλικού FPGA, οπότε για να μην υπάρχει συμφόρηση οι δοκιμές της πρέπει να γίνονται με ελεγχόμενο τρόπο.

Μόλις ενεργοποιηθεί ο λογαριασμός, ο χρήστης μπορεί να προχωρήσει σε είσοδο στο σύστημα (login) χρησιμοποιώντας το όνομα χρήστη και το συνθηματικό που επέλεξε κατά την εγγραφή του.

2.1.2 Διαχείριση βιομοντέλων σε μορφή SBML

Η δεύτερη βασική λειτουργία της υπηρεσίας ιστού είναι να παρέχει στους πιστοποιημένους / εξουσιοδοτημένους χρήστες τη δυνατότητα να διαχειρίζονται τα βιομοντέλα τους τα οποία βρίσκονται σε μορφή SBML. Για να εισάγει ο χρήστης ένα SBML βιομοντέλο στο σύστημα πρέπει να το μεταφέρει (upload) μέσω μιας κατάλληλα διμορφωμένης φόρμας που επιτρέπει την επιλογή του SBML αρχείου τοπικά από τον προσωπικό του υπολογιστή καθώς και ένα αναγνωριστικό ονόματος για εύκολη επιλογή του κατά την τοποθέτηση εργασιών στοχαστικής προσομοίωσης (βλ. 2.1.3).

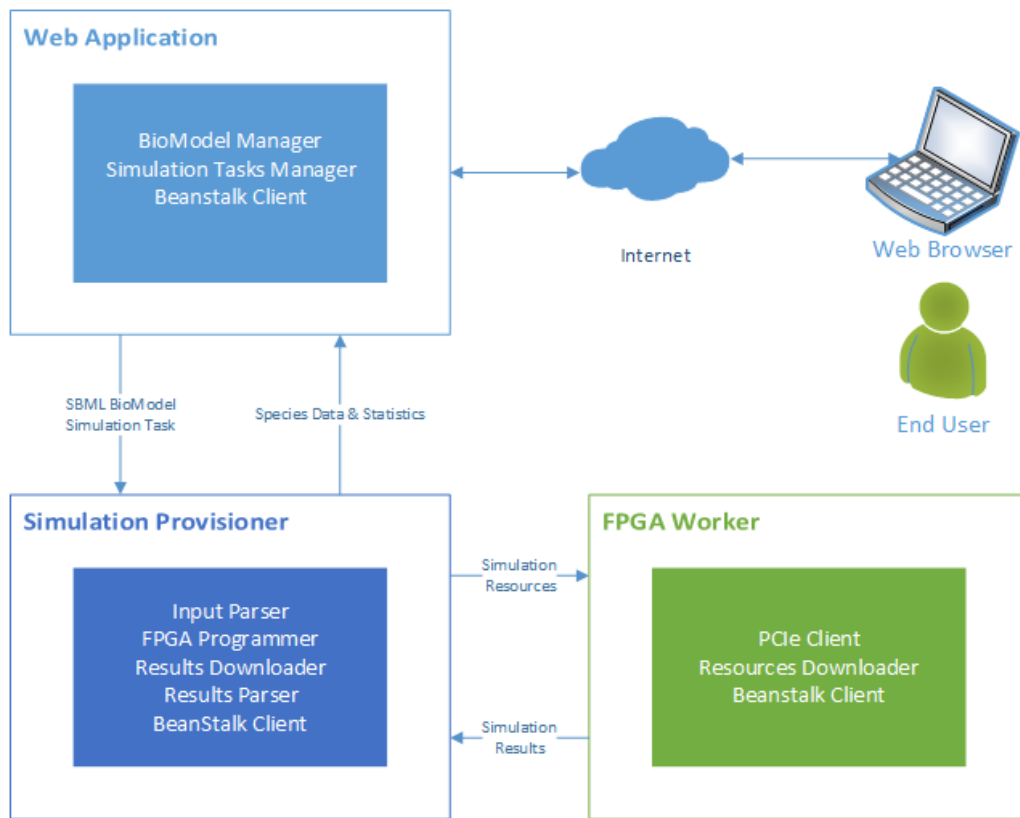
Πίνακας 1: Παράμετροι εργασιών στοχαστικής προσομοίωσης

Παράμετρος	Περιγραφή	Αρχική Τιμή
<i>Όνομα Προσομοίωσης</i>	Αναγνωριστικό νούμερο για την προσομοίωση	Τυχαίο όνομα
<i>Βιο-Μοντέλο SBML</i>	Επιλογή ενός SBML μοντέλου από αυτά που έχουν εισαχθεί στο σύστημα	-
<i>Αλγόριθμος</i>	Ο αλγόριθμος στοχαστικής προσομοίωσης (FRM ή NRM)	FRM
<i>Τρόπος Λειτουργίας</i>	Ο τύπος παράλληλης επεξεργασίας (SSIP ή MSIP)	SSIP
<i>Αριθμός Επαναλήψεων</i>	Ο αριθμός των επαναλήψεων (Repetitions, R)	128
<i>Χρονος Προσομοίωσης</i>	Ο χρόνος προσομοίωσης (lab time) του βιομοντέλου σε δευτερόλεπτα (Simulation Time, Tsim)	100.0 sec
<i>Περίοδος Δειγματοληψίας</i>	Κάθε πόσα δευτερόλεπτα χρόνου προσομοίωσης καταγράφονται οι πληθυσμοί των μοριακών ειδών (Sampling Period, T _{sam})	1.0 sec
<i>Μέγιστα Εσωτερικά Βήματα</i>	Ο μέγιστος επιτρεπόμενος αριθμός βημάτων που αντιστοιχούν σε κύκλους αντιδράσεων κατά τη διάρκεια μια περιόδου δειγματοληψίας (Max Internal Steps, MIS)	High (1B)

Εφόσον η μεταφορά ολοκληρωθεί με επιτυχία τότε το αρχείο τοποθετείται σε κατάλληλο φάκελο στον προσωπικό χώρο εργασίας του χρήστη (workspace). Αυτή τη στιγμή ο χώρος αυτός δεν έχει περιορισμό μεγέθους για αποτροπή υπερβολικής κατανάλωσης αποθηκευτικού χώρου συνεπώς κάθε πιστοποιημένος χρήστης είναι ελεύθερος να εισάγει στο σύστημα όσα SBML βιομοντέλα επιθυμεί να χρησιμοποιήσει για στοχαστική προσομοίωση στην υπηρεσία ιστού.

2.1.3 Τοποθέτηση εργασιών στοχαστικής προσομοίωσης

Η κύρια λειτουργία της υπηρεσίας ιστού είναι η τοποθέτηση εργασιών στοχαστικής προσομοίωσης στο σύστημα και η παράλληλη εκτέλεση τους από το κατάλληλο FRM / NRM SoC μεταξύ αυτών που αναπτύχθηκαν για την E.E.I και με το οποίο προγραμματίζεται η συσκευή υλικού FPGA (back-end), πριν ξεκινήσει η προσομοίωση. Για να τοπθετήσει ο χρήστης μια νέα εργασία στοχαστικής προσομοίωσης θα πρέπει να καθορίσει τιμές για τις παραμέτρους του Πίνακα 1. Όλες οι παραπάνω παράμετροι έχουν ενδεικτικές αρχικές τιμές (default values) ώστε ο χρήστης να αλλάζει μόνο αυτές που επιθυμεί. Μετά την επιτυχή ολοκλήρωση μιας εργασίας, η υπηρεσία ιστού παρουσιάζει στο χρήστη την επιλογή για κατέβασμα (download) των αποτελεσμάτων μέσω ενός συμπιεσμένου (zip) αρχείου το οποίο μπορεί να αποθηκεύσει στον προσωπικό υπολογιστή του με σκοπό την ανάλυση ή περαιτέρω επεξεργασία των αποτελεσμάτων.



Εικόνα 2: Η αρχιτεκτονική της υπηρεσίας ιστού με τα κυριότερα συστατικά μέρη των εφαρμογών

2.2 Επισκόπηση Αρχιτεκτονικής

Η αρχιτεκτονική μας περιλαμβάνει τρεις ξεχωριστούς κόμβους / μηχανήματα για τη λειτουργία της και η επικοινωνία μεταξύ τους γίνεται σε τοπικό δίκτυο (LAN) μέσω δικτυακών ουρών μηνυμάτων / εργασιών (message / task queues) όταν εκκινεί ή ολοκληρώνεται μια διαδικασία και μέσω του πρωτοκόλλου SSH (Secure Socket Host) [7] όταν χρειάζεται μεταφορά δεδομένων / αρχείων. Κάθε κόμβος περιλαμβάνει κατάλληλες εφαρμογές που με τη σειρά τους περιλαμβάνουν συστατικά μέρη (components) κατάλληλα για την κάθε διαδικασία που πρέπει να εκτελεστεί. Η συνολική αρχιτεκτονική της υπηρεσίας ιστού με τα κυριότερα συστατικά μέρη της εφαρμογής που εκτελείται σε κάθε κόμβο ξεχωριστά φαίνεται στην Εικόνα 1.

2.2.1 Κόμβος Εφαρμογής Ιστού (Web Application Node)

Ο πρώτος κόμβος είναι αυτός που στεγάζει την κύρια εφαρμογή ιστού (web application) με την οποία αλληλοεπιδρά ο χρήστης της υπηρεσίας όταν την επισκεφτεί με κάποιον φυλλομετρητή (browser) της προτίμησής του. Η πρόσβαση γίνεται προσωρινά κατά τη φάση των δοκιμών μέσω ενός προσωρινά δημιουργημένου ονόματος χώρου (domain name): **webstoch.ddns.net** ενώ στο μέλλον έχουμε σκοπό να προβούμε σε αγορά και χρήση ενός ονόματος χώρου όπως το **stochsocs.org**.

Η εφαρμογή ιστού στο πρώτο μηχανήμα επιτρέπει στο χρήστη να εισάγει (upload), να διαχειρίζεται ή να διαγράφει από το σύστημα τα επιθυμητά βιο-μοντέλα του σε μορφή SBML (BioModel Manager) καθώς και να

τοποθετεί, να επεξεργάζεται ή να απενεργοποιεί μια στοχαστική προσομοίωση που τελικά θα εκτελεστεί στο FPGA υλικό (Simulation Tasks Manager). Οι παράμετροι που μπορεί να επιλέξει ο χρήστης σχετικά με την προσομοίωση που επιθυμεί να εκτελεστεί παρουσιάστηκαν στην ενότητα 2.1 μαζί με τις υπόλοιπες προδιαγραφές της υπηρεσίας ιστού.

Για την υλοποίηση χρησιμοποιήθηκε η πλατφόρμα ανάπτυξης – πλαίσιο λογισμικού (development framework) Symfony2 [8] με τη βοήθεια της οποίας δημιουργήθηκαν οι MVC (Model View Controller) δομές λογισμικού σε γλώσσα PHP έκδοσης 5.6 [9] ενώ η τελική αποθήκευση των δεδομένων (persistent storage) γίνεται πάνω σε MySQL [10] σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) μέσω της πολύ αποδοτικής βιβλιοθήκης διαχείρισης σχέσεων - αντικειμένων (ORM – Object Relations Manager) Doctrine2 [11]. Ο εξυπηρετητής ιστού που επιλέχθηκε είναι ο δημοφιλής Apache2 [12].

Έγινε επίσης χρήση ενός πλήθους υποστηρικτικών βιβλιοθηκών λογισμικού με σκοπό την επιτάχυνση των εργασιών ανάπτυξης ενώ όλο το λογισμικό και οι εξαρτήσεις του εκτελούνται κατά τη φάση δοκιμών σε εικονική μηχανή (VM) με λειτουργικό Linux, η οποία και δημιουργήθηκε με τα κατάλληλα εργαλεία διαχείρισης και εγκατάστασης: VirtualBox [13], Vagrant [14] και Puppet [15].

2.2.2 Κόμβος Προετοιμασίας και Τοποθέτησης Προσομοιώσεων (Simulation Provisioner)

Ο δεύτερος κόμβος εκτελεί κατάλληλο λογισμικό που ονομάζεται *Simulation Provisioner* και το οποίο είναι υπεύθυνο για τις εξής τέσσερις κύριες λειτουργίες:

- Προετοιμασία της προσομοίωσης μέσω του εργαλείου *Input Parser* και μετέπειτα αποστολή των απαραίτητων αρχείων και παραμέτρων, ανάλογα τον αλγόριθμο στοχαστικής προσομοίωσης (FRM / NRM) που επιθυμεί ο χρήστης και επέλεξε κατά την τοποθέτηση μια νέας προσομοίωσης μέσω της εφαρμογής ιστού.
- Προγραμματισμό της συσκευής FPGA με το κατάλληλο αρχείο bitstream ανάλογα με τον αλγόριθμο στοχαστικής προσομοίωσης (FRM / NRM) και τον τρόπο παράλληλης επεξεργασίας (SSIP / MSIP) που επιθυμεί ο χρήστης και επέλεξε κατά την τοποθέτηση μια νέας προσομοίωσης μέσω της εφαρμογής ιστού.
- Επεξεργασία των αποτελεσμάτων με το εργαλείο *Results Parser* ώστε να δημιουργηθούν δεδομένα και στατιστικά κατανοητά από τον τελικό χρήστη.
- Επικοινωνία με τους άλλους δύο κόμβους για αποστολή και παραλαβή δεδομένων / αρχείων και μηνυμάτων εργασιών από τις κατάλληλες ουρές όταν αυτό είναι απαραίτητο.

Τα κύρια συστατικά μέρη λογισμικού που χρησιμοποιούνται για την επίτευξη των παραπάνω λειτουργιών είναι τα εξής:

- **Input Parser** – Προετοιμασία των δεδομένων / αρχείων της προσομοίωσης
- **FPGA Programmer** – Προγραμματισμός της συσκευής FPGA
- **Results Downloader** – Παραλαβή των αποτελεσμάτων από το FPGA SoC
- **Results Parser** – Επεξεργασία των αποτελεσμάτων
- **Beanstalk Client** – Επικοινωνία με τους άλλους κόμβους μέσω ουρών μηνυμάτων / εργασιών

2.2.3 Κόμβος Εκτέλεσης Προσομοιώσεων στο Υλικό FPGA (FPGA Worker)

Ο τρίτος κόμβος είναι αυτός που εκτελεί τελικά τις προσομοιώσεις που επιθυμεί ο χρήστης, χρησιμοποιώντας το λογισμικό *FPGA Worker* που λειτουργεί πάνω στο υπάρχον εργαλείο λογισμικού Serial Communicator και το FPGA υλικό (FRM, NRM SoCs) που είχαμε ήδη σχεδιάσει και υλοποιήσει για τις Ε.Ε1.1 & 1.2. Τα κύρια συστατικά μέρη λογισμικού που χρησιμοποιεί για την επίτευξη των παραπάνω λειτουργιών είναι τα εξής:

- **Serial Communicator** – Αποστολή και λήψη δεδομένων προς / από το FPGA SoC
- **Resources Downloader** – Παραλαβή των δεδομένων / αρχείων του SBML βιομοντέλου καθώς και των παραμέτρων προσομοίωσης που χρειάζεται το FPGA SoC
- **Beanstalk Client** – Επικοινωνία με τον κόμβο Simulation Provisioner για την έναρξη ή ολοκλήρωση των λειτουργιών

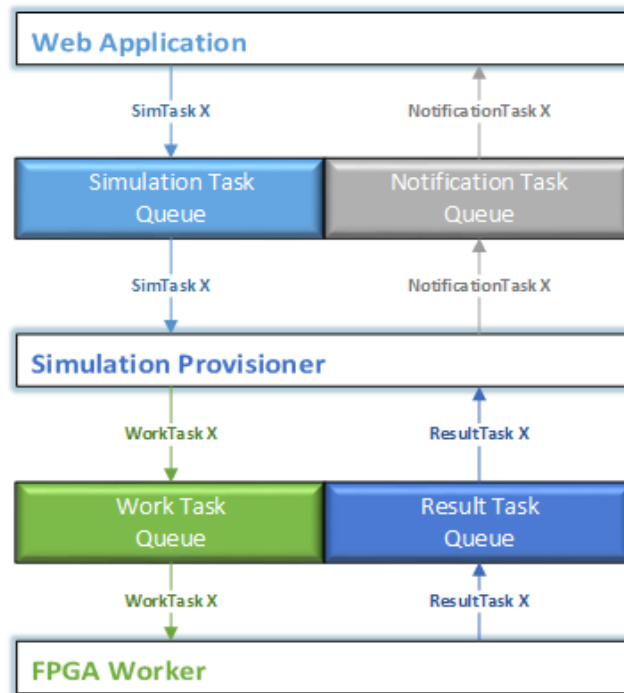
Οι κεντρικές εφαρμογές της υπηρεσίας ιστού στον 2^ο κόμβο (Simulation Provisioner) και στον 3^ο κόμβο (*FPGA Worker*) είναι γραμμένες σε σύγχρονη C++11 για συμβατότητα με τα υπάρχοντα εργαλεία όπου είχαν ήδη αναπτυχθεί (*Input Parser*, *Serial Communicator*, *Results Parser*) και χρησιμοποιούνταν για την έναρξη και ολοκλήρωση «χειροκίνητων» προσομοιώσεων. Πάνω στα τελευταία έγιναν επίσης αρκετές αλλαγές με σκοπό την ορθότερη σχεδίαση του συστήματος και την επίλυση κάποιων ασυμβατοτήτων ή / και προβλημάτων που παρατηρήθηκαν κατά τη φάση της ενοποίησης τους (integration).

Τέλος πρέπει να αναφέρουμε ότι για καλύτερη διαχείριση του αρκετά πολύπλοκου συστήματος, από της σκοπιά της ανάπτυξης, έγινε χρήση σύγχρονων εργαλείων κατασκευής ολοκληρωμένων συστημάτων λογισμικού (build system) και διαχείρισης των εξαρτήσεών τους (dependency manager). Επίσης έγινε χρήση και άλλων βιβλιοθηκών ανοιχτού λογισμικού πέρα από αυτές που αναφέρθηκαν προηγουμένως (Boost System, Boost FileSystem, Boost IOStreams, Boost Process, CurlCPP, GoogleFlags, GoogleLog, GoogleTest). Όλα τα παραπάνω είχαν τελικά ως αποτέλεσμα την επιτάχυνση των εργασιών ανάπτυξης και την ολοκλήρωση τους στο προβλεπόμενο χρονικό διάστημα.

2.3 Επικοινωνία Κόμβων

Οι κόμβοι ανταλλάσσουν μηνύματα, χρησιμοποιώντας το πρωτόκολλο μηνυμάτων BeanStalk [16], το οποίο είναι βελτιστοποιημένο για ανάθεση εργασιών (jobs / tasks). Για αυτό το σκοπό έχουν δημιουργηθεί οι ακόλουθες ουρές μηνυμάτων / εργασιών που φαίνονται στην Εικόνα 3 και αντίστοιχα στο Πίνακα 2. Ο αποστολέας και ο παραλήπτης είναι οι εφαρμογές που τρέχουν στον κατάλληλο κόμβο και τοποθετούν ή λαμβάνουν μια εργασία αντίστοιχα.

Για την ανταλλαγή δεδομένων / αρχείων χρησιμοποιείται το πρωτόκολλο SSH μέσω των βιβλιοθηκών ανοιχτού λογισμικού libSSH2 [17] και libCurl [18]. Η πρώτη βιβλιοθήκη παρέχει στη δεύτερη τις υποστηριζόμενες λειτουργίες του πρωτόκολλου SSH ενώ η δεύτερη χρησιμοποιείται από τις εφαρμογές που δημιουργήσαμε για την εκτέλεση των μεταφορών σε ένα υψηλότερο επίπεδο αφάιρεσης (abstraction). Τα δεδομένα / αρχεία που μεταφέρονται κατά την εκτέλεση μιας FRM ή NRM προσομοίωσης στο αντίστοιχο FPGA SoC φαίνονται στον Πίνακα 3.



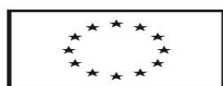
Εικόνα 3: Η μεταφορά μηνυμάτων μεταξύ των κόμβων με κατάλληλες ουρές εργασίας

Πίνακας 2: Ουρές μηνυμάτων που χρησιμοποιεί η υπηρεσία ιστού

Όνομα Ουράς	Αποστολέας	Παραλήπτης	Λειτουργία
<i>Simulation Tasks</i>	Web Application	Simulation Provisioner	Αποστολή μοντέλου SBML και παραμέτρων προσομοίωσης
<i>Setup Tasks</i>	Simulation Provisioner	FPGA Worker	Προγραμματισμός συσκευής FPGA με κατάλληλο bitstream ανάλογα τον αλγόριθμο και τον τρόπο λειτουργίας
<i>Work Tasks</i>	Simulation Provisioner	FPGA Worker	Αποστολή δεδομένων / αρχείων του μοντέλου και εκκίνηση προσομοίωσης στη συσκευή FPGA
<i>Results Tasks</i>	FPGA Worker	Simulation Provisioner	Ολοκλήρωση προσομοίωσης και επιστροφή αποτελεσμάτων
<i>Completed Tasks</i>	Simulation Provisioner	Web Application	Ενημέρωση εφαρμογής ιστού με τα τελικά αποτελέσματα

Πίνακας 3: Αρχεία δεδομένων κατά την εκτέλεση της κάθε προσομοίωσης

Όνομα Αρχείου	Αλγόριθμος	Αποστολέας	Παραλήπτης
<i>Reaction Table (RT)</i>	FRM, NRM	Simulation Provisioner	FPGA Worker
<i>Species Table (ST)</i>	FRM, NRM	Simulation Provisioner	FPGA Worker
<i>Mersenne Twister (MT)</i>	FRM, NRM	Simulation Provisioner	FPGA Worker
<i>Dependency Graph (DG)</i>	NRM	Simulation Provisioner	FPGA Worker
<i>Binary Results (BR)</i>	FRM, NRM	FPGA Worker	Simulation Provisioner



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

3 ΔΙΚΤΥΑΚΟΣ ΚΟΜΒΟΣ (FRONTEND)

Στην ενότητα αυτή θα παρουσιαστεί ο δικτυακός κόμβος που στεγάζει την κύρια εφαρμογή ιστού (web application) με την οποία αλληλεπιδρά ο χρήστης της υπηρεσίας όταν την επισκεφτεί με τον φυλλομετρητή (browser) που διαθέτει.

3.1 Περιβάλλον Εκτέλεσης, Αυτοματοποίηση Εγκατάστασης και Ενεργοποίησης

Ο δικτυακός κόμβος στεγάζεται σε εικονική μηχανή (Virtual Machine – VM) η οποία φιλοξενείται (Guest) πάνω σε ένα δυνατό μηχάνημα ξενιστή (Host) με τα χαρακτηριστικά που φαίνονται στον Πίνακα 4:

Πίνακας 4: Χαρακτηριστικά δικτυακού κόμβου

Επεξεργαστής (CPU)	8-Core Intel Core-i7 5960X @ 3GHz (3.5GHz Turbo) with HT (16 Threads)
Κεντρική Μνήμη (Memory)	32GB (4 X 8GB) Quad Channel DDR4 @ 2133MHz CAS15
Αποθηκευτικός Χώρος	2 X 256GB SSD RAID1 (Mirroring)
Δίκτυο	Intel Ethernet Controller 1Gbps

Η χρήση εικονικής μηχανής για το συγκεκριμένο κόμβο μας παρέχει τα εξής πλεονεκτήματα σε σχέση με την απευθείας εγκατάσταση του στο λειτουργικό που εκτελεί το μηχάνημα ξενιστή (Host) της εικονικής μηχανής:

- Ευελιξία στην επιλογή πακέτων λογισμικού ανεξαρτήτως του τύπου ή της έκδοσης του λειτουργικού συστήματος που διαθέτει ο ξενιστής
- Ευκολία εγκατάστασης ή απεγκατάστασης
- Γρήγορη μεταφορά σε νέο μηχάνημα σε περίπτωση βλάβης του αρχικού
- Δυνατότητα για κανομή του φορτίου εφόσον γίνει αναπαραγωγή (replication) σε περισσότερα του ενός μηχανήματα
- Χρήση της εικονικής μηχανής σε υπηρεσίες νέφους (cloud)

Τόσο ο ξενιστής όσο και η φιλοξενούμενη εικονική μηχανή χρησιμοποιούν λειτουργικό σύστημα GNU / Linux της οικογένειας Ubuntu ρυθμισμένο κατάλληλα για την απρόσκοπτη λειτουργία της υπηρεσίας ιστού. Ειδικά για την περίπτωση της εικονικής μηχανής η δημιουργία και η διαχείριση της εικονικής μηχανής καθώς και η διαρύθμιση των απαραίτητων πακέτων λογισμικού έγινε με τη βοήθεια των εργαλείων VirtualBox, Vagrant και Puppet.

Πίνακας 5: Τα κυριότερα πακέτα λογισμικού του διαδικτυακού κόμβου

Όνομα	Τύπος	Έκδοση
<i>Apache</i>	Εξυπηρετητής Ιστού (Web Server)	2.4.7
<i>MySQL</i>	Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (Relational Database Management System – RDBMS)	5.5.44
<i>PHP</i>	Διερμηνέας (interpreter) για τη γλώσσα διαδικτυακού προγραμματισμού PHP	5.5.9
<i>OpenSSH</i>	Εξυπηρετητής για το ασφαλές δικτυακό πρωτόκολλο SSH (Secure SHell)	6.6
<i>BeanstalkD</i>	Εξυπηρετητής για ουρές εργασίας (work queues)	1.10

Το VirtualBox της εταιρείας Oracle είναι μια από της πιο διαδεδομένες υλοποιήσεις δημιουργίας και διαχείρισης εικονικής μηχανής (hypervisor) για επεξεργαστές αρχιτεκτονικής x86. Παρέχει υποστήριξη για τα περισσότερα σύγχρονα λειτουργικά συστήματα υπολογιστών, πλήθος λειτουργιών και εικονικών συσκευών, με μόνο μειονέκτημα την έλλειψη αυτοματοποίησης κατά τη δημιουργία ή χρήση της εικονικής μηχανής που διαχειρίζεται.

Το κενό αυτό το συμπληρώνει ιδανικά το εργαλείο Vagrant το οποίο είναι ένα κέλυφος (wrapper) γύρο από τις πιο διαδεδομένες υλοποιήσεις διαχείρισης εικονικής μηχανής που επιτρέπει την αυτόματη δημιουργία, εκκίνηση, προσωρινή απενεργοποίηση, επαναφορά, τερματισμό της εικονικής μηχανής με μόλις μια εντολή στο τερματικό του λειτουργικού συστήματος ξενιστή. Έτσι επιτυγχάνεται ευκολία χρήσης της εικονικής μηχανής χωρίς να απωλεσθεί οποιαδήποτε λειτουργικότητα αυτής.

Η αρχική διαρύθμιση όλων των πακέτων λογισμικού που είναι απαραίτητα για τη λειτουργία της υπηρεσίας μας είναι μια επίπονη διαδικασία που δεν είναι εύκολο να εκτελεστεί απρόσκοπτα και χωρίς κανένα λάθος από το διαχειριστή του συστήματος. Για αυτό το λόγο επιλέξαμε να χρησιμοποιήσουμε για την αυτοματοποίηση της το εργαλείο Puppet, που είναι ένα από το πιο διαδεδομένα εργαλεία διαχείρισης της διαρύθμισης ενός συστήματος. Χωρίς να μπορούμε σε πολλές λεπτομέρειες το συγκεκριμένο εργαλείο επιτρέπει την αυτόματη εγκατάσταση πακέτων λογισμικού χρησιμοποιώντας μια δηλωτική (declarative) γλώσσα για να περιγραφούν οι ανάγκες διαρύθμισης τους.

Με τα παραπάνω εργαλεία καταφέραμε να προσθέσουμε εύκολα στη ροή εργασιών της υπηρεσίας ιστού τη λειτουργικότητα της εικονικής μηχανής που περιέχει τον δικτυακό κόμβο μας με όλα τα πακέτα λογισμικού που είναι απαραίτητα για τη λειτουργία του. Τα σημαντικότερα από αυτά παρουσιάζονται στον Πίνακα 5.

Ο εξυπηρετητής ιστού Apache προκειμένου να επικοινωνήσει με τον φυλλομετρητή (browser) των χρηστών χρησιμοποιεί τη θύρα (port) 80. Παρόμοια ο εξυπηρετητής για ουρές εργασίας Beanstalk επικοινωνεί με τα προγράμματα πελάτες (clients) των υπολογιστικών κόμβων (βλ. Κεφάλαιο 4) χρησιμοποιεί τη θύρα 11300. Τέλος ο εξυπηρετητής OpenSSH για το ασφαλές δικτυακό πρωτόκολλο SSH χρησιμοποιεί τη θύρα 22 στην οποία πρέπει να συνδεθεί ο διαχειριστής του συστήματος και να έχει πρόσβαση κελύφους (shell access). Επειδή υπήρχε πρόβλημα στο άνοιγμα των θυρών 80 και 11300 πάνω στην εικονική μηχανή μέσω προώθησης (port forwarding) επιλέξαμε τη λύση των τοπικών (local) SSH τούνελ (tunnels).

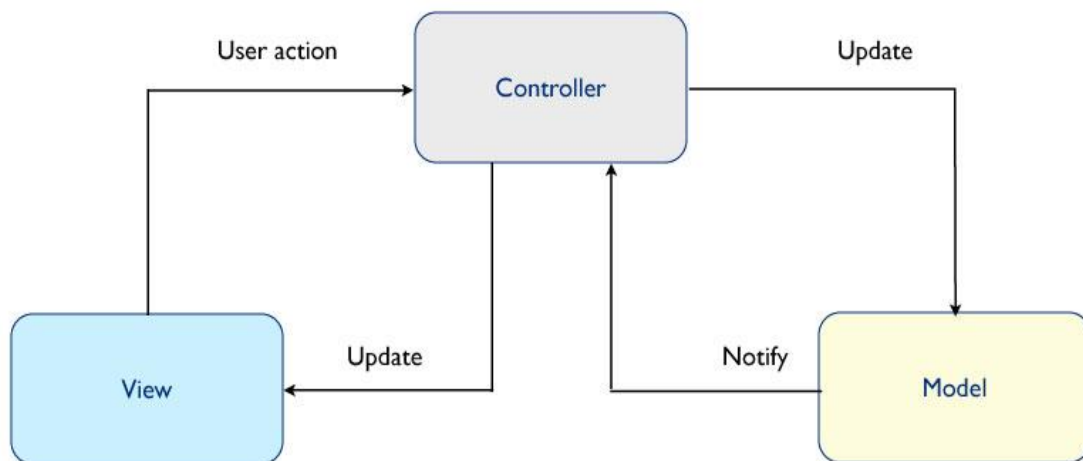
3.2 Πλατφόρμα Ανάπτυξης και Υποστηρικτικές Βιβλιοθήκες

Πίνακας 6: Οι βοηθητικές βιβλιοθήκες λογισμικού του διαδικτυακού κόμβου

Όνομα Πακέτου Βιβλιοθήκης	Σύντομη Περιγραφή	Έκδοση
<i>doctrine/orm</i>	Διαχείριση σχέσεων αντικειμένων (ORM)	2.2
<i>doctrine/dbal</i>	Διαχείριση σύνδεσης με το MySQL RDBMS	2.4
<i>doctrine/doctrine-bundle</i>	Ενσωμάτωση Doctrine μέσα στο Symfony2	1.4
<i>doctrine/doctrine-fixtures-bundle</i>	Εισαγωγή αρχικών (default) δεδομένων	2.2
<i>symfony/assetic-bundle</i>	Διαχείριση βοηθητικών αρχείων εικόνων JS, CSS	2.3
<i>symfony/swiftmailer-bundle</i>	Αποστολή E-Mail	2.3
<i>symfony/monolog-bundle</i>	Καταγραφή συμβάντων (system logger)	2.4
<i>sensio/distribution-bundle</i>	Βοηθητικές λειτουργίες για το Symfony2	3.4
<i>sensio/framework-extra-bundle</i>	Βοηθητικές λειτουργίες για το Symfony2	3.0
<i>incenteev/composer-parameter-handler</i>	Βοηθητικές λειτουργίες για το Symfony2	2.0
<i>gedmo/doctrine-extensions</i>	Προηγμένες λειτουργίες για το Doctrine	2.4
<i>friendsofsymfony/user-bundle</i>	Διαχείριση λογαριασμών χρήστη	2.0
<i>knplabs/knp-gaufrette-bundle</i>	Πρόσβαση στο σύστημα αρχείων με αφηρημένο (abstract) τρόπο	dev
<i>leazy/pheanstalk-bundle</i>	Πελάτης (client) για το πρωτόκολλο του Beanstalk	3.0
<i>mopa/bootstrap-bundle</i>	Σχεδίαση προτύπων (templates) HTML5	dev
<i>knplabs/knp-menu-bundle</i>	Δημιουργία μενού	2.0
<i>liip/theme-bundle</i>	Βοηθητικές λειτουργίες για τα πρότυπα (templates)	1.3
<i>twbs/bootstrap</i>	Πλαίσιο σχεδίασης προτύπων της εταιρείας Twitter	3.3
<i>vich/uploader-bundle</i>	Βοηθητικές λειτουργίες για τη μεταφορά (upload) αρχείων	1.0
<i>herzult/php-ssh</i>	Πελάτης (client) για το πρωτόκολλο SSH	1.1

Η πλατφόρμα ανάπτυξης – πλαίσιο λογισμικού (development framework) που επιλέξαμε να χρησιμοποιήσουμε είναι η Symfony2 στην έκδοση 2.7 με τη βοήθεια της οποίας δημιουργήθηκαν στη γλώσσα PHP οι MVC (Model View Controller) δομές λογισμικού. Η τελική αποθήκευση των δεδομένων (persistent storage) γίνεται πανω σε MySQL σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) μέσω της πολύ αποδοτικής βιβλιοθήκης διαχείρισης σχέσεων - αντικειμένων (ORM – Object Relations Manager) Doctrine2.

Οι βοηθητικές βιβλιοθήκες λογισμικού που χρησιμοποιήσαμε φαίνονται στον Πίνακα 6.



Εικόνα 4: Το αρχιτεκτονικό μοτίβο MVC της εφαρμογής ιστού

3.3 Μοντέλα, Ελεγκτές Δεδομένων και Όψεις (MVC)

Η εφαρμογή ιστού (web application) που υλοποιήσαμε χρησιμοποιεί MVC (Model – View – Controller) δομές λογισμικού. Οι δομές αυτές περιλαμβάνονται στο αρχιτεκτονικό μοτίβο λογισμικού MVC που χρησιμοποιείται κατά κόρον για την υλοποίηση διεπαφών χρήστη όπως αυτές που χρειάζονται σε μια σύγχρονη εφαρμογή ιστού ή και σε κλασικές εφαρμογές που χρησιμοποιούν γραφικές διεπαφές χρήστη (Graphical User Interfaces – GUIs). Με αυτόν τον τρόπο το λογισμικό χρησιμοποιεί τρία ξεχωριστά κομμάτια τα οποία είναι διασυνδεδεμένα όπως φαίνεται στην Εικόνα 4. Το πλαίσιο λογισμικού Symfony2 που χρησιμοποιήσαμε διαθέτει πλήρη υποστήριξη για το συγκεκριμένο αρχιτεκτονικό μοτίβο με πολλά βοηθητικά εργαλεία και συστατικά (components) λογισμικού.

Οι ελεγκτές (controllers) είναι οι κεντρικές οντότητες ελέγχου για κάθε συγκεκριμένη πράξη (action) του χρήστη ενώ σε συγκεκριμένες περιπτώσεις χρησιμοποιείται ο ίδιος ελεγκτής για παραπάνω από μια πράξεις και άρα για περισσότερα μοντέλα (models) και όψεις (views). Η κύρια δουλειά του κάθε ελεγκτή είναι να στέλνει εντολές στο μοντέλο για τα σχετικά δεδομένα που χρειάζεται με σκοπό την ανανέωση της κατάστασης του (state). Στη συνέχεια και αφού λάβει τα απαραίτητα δεδομένα που χρειάζεται, στέλνει εντολές και στη συσχετισμένη με αυτόν όψη ώστε να αλλάξει η παρουσίαση της με βάση τα πιθανώς ανανεωμένα δεδομένα του μοντέλου που χρησιμοποιείται για την πράξη.

Το κάθε μοντέλο είναι η διεπαφή με την οποία αναπαρίστανται τα δεδομένα που ζητάει ο ελεγκτής και εμφανίζονται στην όψη. Στη δική μας περίπτωση το κάθε μοντέλο είναι μια κλάση (class) στην οποία περιγράφονται ως ιδιότητες (properties) τα χαρακτηριστικά του μοντέλου και στη συνέχεια οι ελεγκτές δημιουργούν στιγμιότυπα (instances) – αντικείμενα (objects) αυτών για τις αντίστοιχες εγγραφές (records) που βρίσκονται στους πίνακες (tables) της σχεσιακής βάσης δεδομένων (MySQL) της εφαρμογής μας.

Για να το επιτύχουν αυτό οι ελεγκτές μας χρησιμοποιούν μια πολύ σημαντική βιβλιοθήκη λογισμικού που ονομάζεται Doctrine ORM (Object Relation Mapper). Η βιβλιοθήκη αυτή χρησιμοποιεί την ομώνυμη προγραμματιστική τεχνική ώστε να αντιστοιχεί τα μοντέλα - κλάσεις με τις ιδιότητες τους σε πίνακες και πεδία της βάσης δεδομένων. Με αυτόν τον τρόπο η υλοποίηση των ελεγκτών μας χρησιμοποιεί απευθείας τα πρώτα χωρίς να χρειάζεται να αναφέρεται ρητά στα δεύτερα.

3.3.1 Μοντέλα Δεδομένων

Πίνακας 7: Οι ιδιότητες του μοντέλου των SBML Bio-Μοντέλων (BioModels)

Ιδιότητα	Σύντομη Περιγραφή	Τύπος Δεδομένων
<i>id</i>	Το αναγνωριστικό νούμερο	Integer (Auto Increment)
<i>name</i>	Το αναγνωριστικό όνομα του	String (Max Length = 255 Characters)
<i>fileName</i>	Το όνομα αρχείου SBML με κατάληξη .xml	String (MaxLength = 255 Characters) MimeType = application/sbml+xml)
<i>createdAt</i>	Η χρονική στιγμή που δημιουργήθηκε η εγγραφή	DateTime Timestampable OnCreate
<i>updatedAt</i>	Η χρονική στιγμή που ανανεώθηκε η εγγραφή	DateTime Timestampable OnUpdate
<i>status</i>	Η κατάσταση της εγγραφής (Ενεργή / Ανένεργη)	Boolean
<i>creator</i>	Ο λογαριασμός χρήστη που δημιούργησε την εγγραφή	ManyToOne Relationship [UserAccount Entity]
<i>simTasks</i>	Οι εργασίες προσομοίωσης στις οποίες συμμετέχει το βιο-μοντέλο	OneToMany Relationship [SimTask Entity]
<i>file</i>	Το αρχείο SBML με τα δεδομένα του βιο-μοντέλου	File (Uploadable on BioModels workspace, Max Size = 8MB)

Για την εφαρμογή μας σχεδιάστηκαν και υλοποιήθηκαν τρία μοντέλα δεδομένων για να αναπαρασταθούν τα SBML Bio-Μοντέλα (BioModel) που διαχειρίζεται ο χρήστης στο χώρο εργασίας του (workspace), οι εργασίες προσομοίωσης που τοποθετεί ο χρήστης στο υπολογιστικό σύστημα (SimulationTask) και τέλος οι λογαριασμοί των χρηστών που έχουν πρόσβαση στην εφαρμογή. Στους Πίνακες 7, 8 και 9 δίνονται οι ιδιότητες των μοντέλων αυτών, οι τύποι δεδομένων μαζί με τους περιορισμούς που τους συνοδεύουν και μικρά σχόλια – περιγραφές για αυτά όπου χρειάζεται.

Κάποια από τα πεδία σε κάθε μοντέλο είναι σχεσιακού τύπου και συνδέουν την ιδιότητα – πεδίο με αντικείμενα κάποιου άλλου μοντέλου ώστε να αναπαρασταθούν από τον ORM οι σχέσεις αυτές στη σχεσιακή βάση δεδομένων στην οποία τελικά όλα καταλήγουν να αποθηκεύονται όταν δε χρειαζόμαστε άλλο τα αντικείμενα στη μνήμη (persist operation). Οι σχέσεις αυτές αναπαριστούν ξένα κλειδιά (foreign keys) στους πίνακες της βάσης δεδομένων και καθορίζουν τα SQL ερωτήματα που θα δημιουργηθούν για τη δημιουργία (insert), ανανέωση (update) ή επιλογή (select) καθώς και τις ενώσεις (joins) μεταξύ πινάκων των τελευταίων ώστε να διαβαστούν οι εγγραφές (records). Για όλα τα πεδία έχουν δημιουργηθεί κατάλληλοι getters και setters ώστε να είναι δυνατή η χρήση τους από όλο το υπόλοιπο πλαίσιο λογισμικού, ειδικά στις περιπτώσεις των σχεσιακών πεδίων όπου είναι απαραίτητη η ανανέωση και των δύο μοντέλων άρα και πινάκων στη βάση δεδομένων.

Πρέπει να αναφέρουμε ότι δώθηκε ιδιαίτερη προσοχή σε αυτές τις περιπτώσεις ένωσης επειδή με βάση την επιλογή μας για το ποιο είναι το πρωτεύον μοντέλο – οντότητα και ποιο το δευτερεύον έπρεπε να γραφτεί

Πίνακας 8: Οι ιδιότητες του μοντέλου των εργασιών προσομοίωσης (SimulationTasks)

Ιδιότητα	Περιγραφή	Τύπος Δεδομένων - Περιορισμοί
<i>id</i>	Το αναγνωριστικό νούμερο	Integer (Auto Increment)
<i>name</i>	Το αναγνωριστικό όνομα του	String (Max Length = 255 Characters)
<i>bioModel</i>	Το βιο-μοντέλο για το οποίο θα εκτελεστεί η προσομοίωση στο υπολογιστικό σύστημα	ManyToOne Relationship [BioModel Entity]
<i>algorithm</i>	Ο αλγόριθμος στοχαστικής προσομοίωσης	String (Max Length = 15 Characters) AssertChoice = FRM or NRM Default = FRM
<i>mode</i>	Ο τρόπος παράλληλης λειτουργίας	String (Max Length = 15 Characters) AssertChoice = SSIP or MSIP Default = SSIP
<i>repetitions</i>	Ο αριθμός των επαναλήψεων	Integer (Range: [1 – 65536]) Default = 128
<i>simulationTime</i>	Ο χρόνος προσομοίωσης του βιομοντέλου σε δευτερόλεπτα	Float (Range: [0.1 – 1,209,600]) Default = 100.0
<i>samplePleriodTime</i>	Κάθε πόσα δευτερόλεπτα χρόνου προσομοίωσης καταγράφονται οι πληθυσμοί των μοριακών ειδών	Float (Range: [0.001 – 1,000,000]) Default = 100.0
<i>maxInternalSteps</i>	Ο μέγιστος αριθμός βημάτων που αντιστοιχούν σε κύκλους αντιδράσεων κατά τη διάρκεια μια περιόδου δειγματοληψίας	Integer (Range: [1,000 – 2,000,000,000]) Default = 1,000,000)
<i>createdAt</i>	Η χρονική στιγμή που δημιουργήθηκε η εγγραφή	DateTime Timestampable OnCreate
<i>updatedAt</i>	Η χρονική στιγμή που ανανεώθηκε η εγγραφή	DateTime Timestampable OnUpdate
<i>completed</i>	Αν έχει ολοκληρωθεί η προσομοίωση	Boolean
<i>failed</i>	Αν έχει αποτύχει η προσομοίωση	Boolean
<i>status</i>	Η κατάσταση της εγγραφής (Ενεργή / Ανένεργη)	Boolean
<i>creator</i>	Ο λογαριασμός χρήστη που δημιούργησε την εγγραφή	ManyToOne Relationship [UserAccount Entity]

κατάλληλος κώδικας που επηρεάζει τον τρόπο και τη χρονική στιγμή που αποθηκεύονται τα αντικείμενα. Δε θα επεκταθούμε σχετικά με αυτό στο παρόν κείμενο καθώς είναι πέραν από τους σκοπούς τους παραδοτέου.

Πίνακας 9: Οι ιδιότητες του μοντέλου των λογαριασμών χρήστη (UserAccounts)

Ιδιότητα	Περιγραφή	Τύπος Δεδομένων
<i>id</i>	Το αναγνωριστικό νούμερο	Integer (Auto Increment)
<i>username</i>	Το όνομα του χρήστη	String
<i>email</i>	Το email του χρήστη	String
<i>password</i>	Το (κωδικοποιημένο) συνθηματικό του χρήστη	String
<i>salt</i>	Το 'salt' για τον κατακερματισμό του συνθηματικού	String
<i>enabled</i>	Αν ο λογαριασμός χρήστη είναι ενεργός	Boolean
<i>locked</i>	Αν ο λογαριασμός χρήστη είναι κλειδωμένος	Boolean
<i>lastLogin</i>	Η τελευταία χρονική στιγμή που μπήκε στο σύστημα ο χρήστης	DateTime
<i>expiresAt</i>	Η χρονική στιγμή που θα πάψει να είναι ενεργός ο λογαριασμός	DateTime
<i>roles</i>	Ο ρόλος του χρήστη στο σύστημα	Array (ROLE_ADMIN or ROLE_USER)
<i>bioModels</i>	Τα βιο-μοντέλα που έχει εισάγει ο χρήστης στο χώρο εργασίας του (workspace)	OneToMany Relationship [BioModel Entity]
<i>simTasks</i>	Οι εργασίες προσομοίωσης που έχει τοποθετήσει ο χρήστης στο σύστημα	OneToMany Relationship [SimTask Entity]

3.3.2 Ελεγκτές

Τέλος οι ελεγκτές που υλοποιήσαμε για κάθε λειτουργία μαζί με τις δρομολογήσεις (routes) με τις οποίες αυτοί σχετίζονται φαίνονται στον Πίνακα 10. Οι πράξεις που εκτελούν είναι CRUD (Create, Read, Update, Delete).

Πίνακας 10: Οι ελεγκτές για τα μοντέλα δεδομένων

Ελεγκτής	Περιγραφή	Δρομολόγηση
<i>BioModel.Upload</i>	Ανέβασμα (upload) ενός βιο-μοντέλου	/biomodel/upload
<i>BioModel.List</i>	Εμφάνιση της λίστας των βιο-μοντέλων	/biomodel/list
<i>BioModel.Show</i>	Εμφάνιση βιο-μοντέλου με αναγνωριστικό {bid}	/biomodel/show/{bid}
<i>BioModel.Edit</i>	Αλλαγή βιο-μοντέλου με αναγνωριστικό {bid}	/biomodel/edit/{bid}
<i>BioModel.Remove</i>	Διαγραφή βιο-μοντέλου με αναγνωριστικό {bid}	/biomodel/remove/{bid}
<i>SimTask.Create</i>	Δημιουργία μιας νέας εργασίας προσομοίωσης	/simtask/create
<i>SimTask.List</i>	Εμφάνιση της λίστας των εργασιών προσομοίωσης	/simtask/list
<i>SimTask.Show</i>	Εμφάνιση εργασίας προσομ. με αναγνωριστικό {tid}	/simtask/show/{tid}
<i>SimTask.Edit</i>	Αλλαγή εργασίας προσομ. με αναγνωριστικό {tid}	/simtask/edit/{tid}
<i>SimTask.Remove</i>	Διαγραφή εργασίας προσομ. με αναγνωριστικό {tid}	/simtask/remove/{tid}

4 ΥΠΟΛΟΓΙΣΤΙΚΟΙ ΚΟΜΒΟΙ (BACKEND)

Σε αυτή την ενότητα θα παρουσιάσουμε τους δύο υπολογιστικούς κόμβους που αναλαμβάνουν να εκτελέσουν τις εργασίες προσομοίωσης καθώς και τα εργαλεία ανάπτυξης και τις υποστηρικτικές βιβλιοθήκες που χρησιμοποιήσαμε. Ο πρώτος κόμβος είναι ο κόμβος συντονισμού των προσομοιώσεων (*Simulation Provisioner*) και ο οποίος έχει σα κύριο στόχο την συγκέντρωση και αποστολή προς εκτέλεση των εργασιών προσομοίωσης τις οποίες τοποθετεί ο χρήστης στο διαδικτυακό κόμβο. Ο δεύτερος είναι ο κόμβος εκτέλεσης των προσομοιώσεων (*FPGA Worker*) και είναι αυτός ο οποίος διαθέτει τη συσκευή FPGA η οποία προγραμματίζεται με το κατάλληλο bitsream, εκτελεί την προσομοίωση με βάση τις παραμέτρους που του δίνει ο κόμβος συντονισμού των προσομοιώσεων και επιστρέφει τελικά τα αποτελέσματα πίσω σε αυτόν. Η Εικόνα 2 στην Ενότητα 2 δείχνει την αλληλεπίδραση αυτών των δύο κόμβων μεταξύ τους και σε σχέση με τον διαδικτυακό κόμβο που παρουσιάστηκε στην Ενότητα 3.

4.1 Εργαλεία ανάπτυξης και Υποστηρικτικές Βιβλιοθήκες

Το λογισμικό των υπολογιστικών κόμβων αναπτύχθηκε κυρίως με τη γλώσσα C++ ακολουθώντας το τελευταίο πρότυπο της (C++14). Μοναδική εξαίρεση αποτελεί το συστατικό *Serial Communicator* το οποίο είναι γραμμένο σε γλώσσα C καθώς ήταν βολικότερη η χρήση της για το συγκεκριμένο σκοπό. Ο μεταγλωττιστής που χρησιμοποιήθηκε είναι ο GCC [19] στην έκδοση 4.9 ενώ για την κατασκευή (build) του λογισμικού χρησιμοποιήθηκαν τα εργαλεία CMake [20] και Bii [21]. Το πρώτο είναι εξέλιξη του γνωστού εργαλείου Make και το χρησιμοποιεί σε ενδιάμεσο στάδιο κατασκευής, αφού πρώτα επεξεργαστεί τις πιο υψηλού επιπέδου περιγραφές που έχουμε προδιαγράψει. Το Bii με τη σειρά του είναι μια καινοτόμος πρόταση στο χώρο των εργαλείων της C++ με στόχο να βοηθήσει στην ενοποίηση (integration) όλων των συστατικών του λογισμικού, επιλύοντας ταυτόχρονα και τα θέματα εξαρτήσεων (dependencies) μεταξύ τους αλλά και μεταξύ των υποστηρικτικών βιβλιοθηκών που αυτά χρησιμοποιούν κατά τη μεταγλώττιση και λειτουργία τους.

Πίνακας 11: Οι υποστηρικτικές βιβλιοθήκες των υπολογιστικών κόμβων

Βιβλιοθήκη	Έκδοση	Επιλογή Κατασκευής	Περιγραφή
<i>lib XML2</i>	2.9.2	Package Manager (OS)	Επεξεργασία XML από την lib SBML
<i>lib SBML</i>	5.11.4	Package Manager (OS)	Επεξεργασία SBML βιο-μοντέλων
<i>lib SSH2</i>	1.5	Package Manager (OS)	Συνδεσιμότητα SSH – SFTP για μεταφορές
<i>lib CURL</i>	7.43	Package Manager (OS)	Υψηλού επιπέδου αναπαράσταση μεταφορών
<i>lib CURL CPP</i>	Dev	Bii	C++ Περιτύλιγμα (wrapper) για τη lib CURL
<i>Google Flags</i>	2.1.2	Bii	Επεξεργασία επιλογών γραμμής εντολών
<i>Google Log</i>	0.3.3	Bii	Καταγραφή συμβάντων (logs)
<i>Google Test</i>	1.7	Bii	Δημιουργία δοκιμών (tests) λογισμικού

Οι βιβλιοθήκες που χρησιμοποιήσαμε για την ανάπτυξη φαίνονται στον Πίνακα 11 μαζί με μια σύντομη περιγραφή και την έκδοσή τους. Όλες είναι ανοιχτού κώδικα με συμβατές άδειες λογισμικού. Για να μεταγλωττιστεί κάθε βιβλιοθήκη είχαμε την επιλογή είτε αυτό να γίνει μέσω του εργαλείου Bii είτε να τις χρησιμοποιήσουμε απευθείας από τη GNU/Linux διανομή μας. Όπου ήταν εφικτό προτιμήσαμε την πρώτη λύση ώστε το λογισμικό μας να είναι όσο το δυνατόν πιο φορητό (portable) και να μην εξαρτάται από το λειτουργικό σύστημα στο οποίο χτίζεται ή εκτελείται. Στις περιπτώσεις που αυτό ήταν αδύνατο, για λόγους που είναι πέραν

από το σκοπό του παρόντος παραδοτέου, επιλέξαμε κατάλληλες εκδόσεις από το διαχειριστή πακέτων που μας παρείχε το λειτουργικό (package manager).

Τέλος αναπτύξαμε και εγκαταστήσαμε στους υπολογιστικούς κόμβους κατάλληλα σκρίπτ εκκίνησης (startup script) τύπου SysV Init με τη γλώσσα κελύφους Bash για εκκίνηση/επανεκκίνηση/τερματισμό των διεργασιών εξυπηρέτησης (daemons). Επίσης για τον προγραμματισμό των συσκευών FPGA με το κατάλληλο bitstream που περιέχει τα FRM / NRM SoCs εγκαταστάθηκε και χρησιμοποιήθηκε από το συστατικό *FPGA Programmer* το εργαλείο Impact που περιέχεται στη σουίτα εργαλείων ISE 14.7 της εταιρείας Xilinx [22], [23], [24].

4.2 Κόμβος Συντονισμού Προσομοιώσεων (Simulation Provisioner)

Ο κόμβος συντονισμού των προσομοιώσεων (*Simulation Provisioner*) έχει σα κύριο στόχο την συγκέντρωση και αποστολή προς εκτέλεση των εργασιών προσομοίωσης τις οποίες τοποθετεί ο χρήστης στο διαδικτυακό κόμβο. Για την επίτευξη των απαραίτητων λειτουργιών του σύμφωνα με τις προδιαγραφές της αρχιτεκτονικής της υπηρεσίας ιστού που αναφέρθηκαν στο κεφάλαιο 2 χρησιμοποιεί τα συστατικά μέρη που ακολουθούν στις επόμενες υποενότητες. Κάποια από αυτά έχουν αναπτυχθεί ανεξάρτητα μεταξύ τους κατά τη διάρκεια εξέλιξης του έργου και χρησιμοποιήθηκαν ατομικά το καθένα πριν ετοιμαστεί η ίδια η υπηρεσία ιστού και γίνει η ενοποίηση τους κάτω από τη στέγη του πρώτου που έχει και την ομώνυμη ονομασία.

Κατά την εκκίνηση (boot) του κόμβου εκτελείται αυτόματα ένα ειδικό σκρίπτ εκκίνησης (startup script) το οποίο δημιουργήσαμε με σκοπό να εκτελέσει με τη σειρά τις ακόλουθες λειτουργίες που είναι απαραίτητες για την ομαλή λειτουργία όλης της υπηρεσίας ιστού:

1. Αρχικοποίηση και εκκίνηση της εικονικής μηχανής που περιέχει το δικτυακό κόμβο που περιγράφηκε στην προηγούμενη ενότητα.
2. Δημιουργία τοπικού SSH τούνελ (tunnel) για τη θύρα 80 της εικονικής μηχανής ώστε ο εξυπηρετητής ιστού (Apache) που βρίσκεται εκεί να είναι διαθέσιμος από το εξωτερικό μηχάνημα και άρα από το διαδίκτυο και τους χρήστες της υπηρεσίας ιστού.
3. Δημιουργία τοπικού SSH τούνελ (tunnel) για τη θύρα 11300 της εικονικής μηχανής ώστε ο εξυπηρετητής για τις ουρές εργασίας (work queues) που βρίσκεται εκεί να είναι διαθέσιμος από τον ίδιο και τον κόμβο εκτέλεσης προσομοιώσεων που θα περιγραφεί στην υποενότητα 4.3.
4. Εκτέλεση του εκτελέσιμου *Simulation Provisioner* και που είναι το κυρίως συστατικό λογισμικού

Το σκρίπτ εκκίνησης είναι εγκατεστημένο στο στάνταρ φάκελο /etc/init.d/ του λειτουργικού συστήματος και ο τρόπος εκτέλεσης του είναι ο ακόλουθος με επιλογές start ή stop ή restart, ανάλογα αν θέλουμε να εκκινήσουμε, να σταματήσουμε ή να επανεκκινήσουμε τις παραπάνω λειτουργίες. Το λειτουργικό σύστημα του κόμβου το εκτελεί αυτόματα με επιλογή start κατά την εκκίνηση του και με stop κατά τον τερματισμό του.

```
sudo /etc/init.d/webstoch.provisioner start | stop | restart
```

4.2.1 Simulation Provisioner

Το κύριως συστατικό το οποίο και καλεί τα υπόλοιπα για όλες τις λειτουργίες του. Αποτελεί την ενοποίηση όλων των υπολοίπων και τα χρησιμοποιεί ως βιβλιοθήκες ενώ το ίδιο δρα ως το εκτελέσιμο το οποίο εκκινείται αυτόματα κάθε φορά που εκκινεί ο κόμβος, μέσω του σκρίπτ εκκίνησης που αναφέρθηκε παραπάνω. Περιλαμβάνει τις ακόλουθες δομές δεδομένων – κλάσεις:

- MersenneTwister – Δημιουργία αρχείου διανυσμάτων καταστάσεων (state vectors) για τον αλγόριθμο παραγωγής τυχαίων αριθμών MersenneTwister [25]
- ResultsDownloader – Παραλαβή αποτελεσμάτων με το πρωτόκολλο SSH - SFTP από τον κόμβο εκτέλεσης της προσομοίωσης χρησιμοποιώντας τη βιβλιοθήκη CURL

Πίνακας 12: Τα αρχεία δεδομένων που παράγει ο Input Parser προς απόστολή στη συσκευή FPGA

Όνομα	Αρχείο	Περιγραφή
<i>Reaction Table</i> [RT]	FRM_RT.bin ή NRM_RT.bin	Περιέχει τις αντιδράσεις του βιο-μοντέλου
<i>Species Table</i> [ST]	FRM_ST.bin ή NRM_ST.bin	Περιέχει την αρχική στοιχειομετρία των μοριακών ειδών του βιο-μοντέλου
<i>Dependency Graph</i> [DG]	NRM_DG.bin	Περιέχει το γράφο εξαρτήσεων των αντιδράσεων του βιο-μοντέλου για τον αλγόριθμο NRM
<i>Parameters</i> [PAR]	FRM_Parameters.bin ή NRM_Parameters.bin	Περιέχει τις παραμέτρους τις στοχαστικής προσομοίωσης σε κατάλληλη μορφή

- SimTask – Παράμετροι για την εκτέλεση της προσομοίωσης που έστειλε ο διαδικτυακός κόμβος
- SimulationProvisioner – Η κύρια κλάση που εκτελεί τις απαραίτητες λειτουργίες για το συντονισμό της προσομοίωσης

4.2.2 Beanstalk Client

Το συγκεκριμένο συστατικό είναι υπεύθυνο για την επικοινωνία με τους άλλους κόμβους μέσω ουρών μηνυμάτων / εργασιών όπως αυτές περιγράφηκαν στην αρχιτεκτονική του συστήματος στην ενότητα 2. Η επεξήγηση της λειτουργίας του και του πρωτοκόλλου Beanstalk δε θα αναλυθεί περαιτέρω στο παρόν έγγραφο καθώς είναι πέρα από τους σκοπούς του.

4.2.3 Input Parser

Προετοιμάζει τα αρχεία δεδομένων της προσομοίωσης με βάση τις παραμέτρους που έστειλε ο διαδικτυακός κόμβος και το αρχείο SBML του βιο-μοντέλου προς προσομοίωση. Η λειτουργία του διαφέρει ανάλογα αν ο αλγόριθμος στοχαστικής προσομοίωσης είναι ο FRM ή ο NRM καθώς στη δεύτερη περίπτωση παράγει έξτρα το αρχείο με τα δεδομένα του γράφου εξαρτήσεων των αντιδράσεων (*Dependency Graph*). Επίσης είναι υπεύθυνο για την επιλογή του κατάλληλου bitstream άρα και SoC με το οποίο θα προγραμματιστεί η συσκευή FPGA που βρίσκεται στον κόμβο εκτέλεσης των προσομοιώσεων. Η επιλογή γίνεται με βάση τον αλγόριθμο, και τον τρόπο παράλληλης επεξεργασίας που επέλεξε ο χρήστης που έβαλε την εργασία προσομοίωσης στο διαδικτυακό κόμβο καθώς και με βάση το μέγεθος του μοντέλου σε αντιδράσεις, τον αριθμό των επαναλήψεων κ.α. Τα αρχεία δεδομένων που έχουν παραχθεί μετά την ολοκλήρωση της εκτέλεσης του είναι αυτά που φαίνονται στον Πίνακα 12.

4.2.4 FPGA Programmer

Προγραμματίζει τη συσκευή FPGA με το κατάλληλο αρχείο bitstream που επέλεξε ο *Input Parser* κατά την εκτέλεση του. Για τον προγραμματισμό χρησιμοποιεί το εργαλείο Impact που περιέχεται στη σουίτα εργαλείων ISE 14.7 της εταιρείας Xilinx το οποίο βρίσκει από το PATH του τερματικού στο οποίο εκτελείται. Κάθε φορά που γίνεται ο προγραμματισμός σημειώνεται το αρχείο bitstream ώστε σε περίπτωση που ξαναζητηθεί ακριβώς το ίδιο να αποφευχθεί νέος άσκοπος προγραμματισμός.

Πρωτού προγραμματιστεί η συσκευή FPGA ελέγχεται αν ο κόμβος εκτέλεσης της προσομοίωσης είναι ενεργός (online) στο τοπικό δίκτυο και αν διαπιστωθεί ότι είναι του στέλνει αίτημα τερματισμού ώστε αυτός να παύσει να λειτουργεί. Αφού εκτελεστεί επιτυχώς ο προγραμματισμός στέλνεται νέο αίτημα με σκοπό να τον εκκινήσει (boot) με τη διαδικασία της αφύπνισης μέσω τοπικού δικτύου (Wake on Lan – WoL).

4.2.5 Results Parser

Επεξεργάζεται τα αποτελέσματα που βρίσκονται στο δυαδικό αρχείο που επιστρέφει μετά το πέρας της προσομοίωσης το FRM ή NRM SoC στη συσκευή FPGA μέσω του πρωτόκολλου PCIe ώστε να δημιουργηθούν δεδομένα και στατιστικά κατανοητά από τον τελικό χρήστη. Η επεξεργασία έχει παραλληλοποιηθεί για να εκμεταλεύεται πλήρως τους σύγχρονους πολυνηματικούς επεξεργαστές με αποτέλεσμα ο χρόνος εκτέλεσης του να ελαχιστοποιείται ακόμα και για πολύ μεγάλα αρχεία αποτελεσμάτων με μέγεθος αρκετά GB.

Αρχικά ξεχωρίζει τα δεδομένα που παρήγαγε ο κάθε επεξεργαστικός πυρήνας του SoC καθότι είναι δυνατόν ο κάθε πυρήνας να εκτελεί μια τελείως διαφορετική προσομοίωση. Έπειτα για κάθε προσομοίωση που διεξήχθη υπολογίζει τις μέσες τιμές καθώς και τα ελάχιστα και μέγιστα για όλους τους μοριακούς πληθυσμούς, εφόσον έχουν εκτελεστεί περισσότερες από μια στοχαστικές επαναλήψεις.

Τα στατιστικά αυτά μαζί με τα αποτελέσματα κάθε επανάληψης αποθηκεύονται σε μορφή CSV (Comma Separated Values) ώστε να μπορεί να γίνει περαιτέρω επεξεργασία τους από άλλο λογισμικό που έχει στη διάθεση του ο χρήστης που τοποθέτησε την προσομοίωση στο δικτυακό κόμβο. Επιπλέον το εργαλείο δημιουργεί και έξτρα CSV αρχεία με στατιστικά σχετικά με την εκτέλεση της προσομοίωσης, όπως ο συνολικός αριθμός κύκλων προσομοίωσης (reaction cycles) που εκτελέστηκαν, ο συνολικός αριθμός περιόδων δειγματοληψίας που καταγράφηκαν καθώς και ο χρόνος που χρειάστηκε για να ολοκληρωθεί η κάθε επανάληψη.

Όλα τα αρχεία που παράγονται τοποθετούνται και σε ένα συμπιεσμένο αρχείο (zip) το οποίο χρησιμοποιεί ο διαδικτυακός κόμβος ως το τελικό αρχείο των αποτελεσμάτων που εμφανίζει στο χρήστη και ο τελευταίος μπορεί να κατεβάσει (download) στον προσωπικό του υπολογιστή.

4.3 Κόμβος Εκτέλεσης Προσομοιώσεων (FPGA Worker)

Ο κόμβος εκτέλεσης προσομοιώσεων (FPGA Worker) είναι αυτός ο οποίος διαθέτει τη συσκευή FPGA η οποία προγραμματίζεται με το κατάλληλο bitsream όπως περιγράφηκε στην προηγούμενη υποενότητα, εκτελεί την προσομοίωση με βάση τις παραμέτρους που του δίνει ο κόμβος συντονισμού των προσομοιώσεων και επιστρέφει τελικά τα αποτελέσματα πίσω σε αυτόν. Για την επίτευξη των απαραίτητων λειτουργιών του σύμφωνα με τις προδιαγραφές της αρχιτεκτονικής της υπηρεσίας ιστού που αναφέρθηκαν στο κεφάλαιο 2 χρησιμοποιεί τα συστατικά μέρη *FPGA Worker*, *Serial Communicator* και *Beanstalk Client*. Στη συγκεκριμένη υποενότητα θα αναλύσουμε μόνο στα πρώτα δύο καθώς το τελευταίο αναφέρθηκε παραπάνω κατά την ανάλυση του κόμβου συντονισμού.

Κατά την εκκίνηση (boot) του κόμβου εκτελείται αυτόματα ένα ειδικό σκριπτ εκκίνησης (startup script) το οποίο δημιουργήσαμε με σκοπό να εκτελέσει με τη σειρά τις ακόλουθες λειτουργίες που είναι απαραίτητες για την ομαλή λειτουργία όλης της υπηρεσίας ιστού:

1. Δημιουργία τοπικού SSH τούνελ (tunnel) για τη θύρα 11300 της εικονικής μηχανής ώστε ο εξυπηρετητής για τις ουρές εργασίας (work queues) που βρίσκεται εκεί να είναι διαθέσιμος από τον ίδιο.
2. Εκτέλεση του εκτελέσιμου *FPGA Worker* και που είναι το κυρίως συστατικό λογισμικό

Πίνακας 13: Οι συσκευές I/O για την επικοινωνία με τη συσκευή FPGA

Όνομα	Περιγραφή
<i>command_send</i>	Συσκευή στην οποία αποστέλεται το είδος της μεταφοράς (rx, tx ή duplex)
<i>transfer_rx</i>	Συσκευή από την οποία λαμβάνονται τα αποτελέσματα της προσομοίωσης
<i>transfer_tx</i>	Συσκευή στην οποία στέλνονται τα δυαδικά αρχεία δεδομένων
<i>status_register</i>	Συσκευή από την οποία μπορούμε να διαβάσουμε τους καταχωρητές (registers) της I/O μονάδας υλικού

Το σκρίπτ εκκίνησης είναι εγκατεστημένο στο στάνταρ φάκελο /etc/init.d/ του λειτουργικού συστήματος και ο τρόπος εκτέλεσης του είναι ο ακόλουθος με επιλογές start ή stop ή restart, ανάλογα αν θέλουμε να εκκινήσουμε, να σταματήσουμε ή να επανεκκινήσουμε τις παραπάνω λειτουργίες. Το λειτουργικό σύστημα του κόμβου το εκτελεί αυτόματα με επιλογή start κατά την εκκίνηση του και με stop κατά τον τερματισμό του.

```
sudo /etc/init.d/webstoch.worker start | stop | restart
```

4.3.1 FPGAWorker

Το κύριως συστατικό το οποίο και καλεί τα υπόλοιπα για όλες τις λειτουργίες του. Αποτελεί την ενοποίηση των υπολοίπων και τα χρησιμοποιεί ως βιβλιοθήκες ενώ το ίδιο δρα ως το εκτελέσιμο το οποίο εκκινείται αυτόματα κάθε φορά που εκκινεί ο κόμβος, μέσω του σκρίπτ εκκίνησης που αναφέρθηκε παραπάνω. Περιλαμβάνει τις ακόλουθες δομές δεδομένων – κλάσεις:

- ResourceDownloader – Παραλαβή των απαραίτητων αρχείων δεδομένων του SBML βιομοντέλου καθώς και των παραμέτρων προσομοίωσης που χρειάζεται το FPGA SoC (RT.bin, ST.bin, DG.bin, MT.bin, Parameters.bin) με το πρωτόκολλο SSH - SFTP από τον κόμβο συντονισμού της προσομοίωσης χρησιμοποιώντας τη βιβλιοθήκη CURL
- FPGAWorker – Η κύρια κλάση που εκτελεί τις απαραίτητες λειτουργίες για την εκτέλεση της προσομοίωσης

4.3.2 Serial Communicator

Εκτελεί τη γρήγορη αποστολή και λήψη δεδομένων προς / από το FPGA SoC. Το λειτουργικό σύστημα διαθέτει εγκατεστημένο και φορτωμένο οδηγό (driver) που δημιουργεί τις κατάλληλες συσκευές που βρίσκονται στο /dev τις οποίες στη συνέχεια προσπελαύνει το συγκεκριμένο συστατικό για να εκτελέσει τις μεταφορές προς / από την I/O μονάδα υλικού η οποία στη συνέχεια επικοινωνεί με το υπόλοιπο SoC. Οι συσκευές αυτές φαίνονται στον πίνακα 13.

Η μεταφορές γίνονται με ασύγχρονο τρόπο χρησιμοποιώντας ξεχωριστά νήματα (POSIX threads) παραγωγού και καταναλωτή τα οποία έχουν πρόσβαση σε διαμοιραζόμενη (shared) ουρά που κρατάει τα δεδομένα στη μνήμη του λειτουργικού. Η μνήμη αυτή είναι δημιουργημένη με ειδικό τρόπο ώστε να είναι σταθερή (pinned) και το λειτουργικό σύστημα φροντίζει να μη την εναλλάξει με άλλα κομμάτια της εικονικής (virtual) μνήμης. Η κοινή χρήση επιτυγχάνεται με κατάλληλους σηματοφόρους (semaphores) ανάγνωσης και εγγραφής.

Τέλος κατά την επιστροφή των δεδομένων είναι εφικτή η καταγραφή των χρόνων εκτέλεσης για κάθε πυρήνα του FRM ή NRM SoC. Αυτό επιτυγχάνεται με το συνεχές διάβασμα (polling) της τιμής ενός συγκεκριμένου καταχωρητή (31) του οποίου το κάθε bit αντιπροσωπεύει την κατάσταση στην οποία βρίσκεται ο κάθε πυρήνας, αν δηλαδή ακόμα εκτελεί την προσομοίωση ή αν αυτή έχει ολοκληρωθεί.

5 ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Στα πλαίσια της παρούσης έρευνας σχεδιάσαμε και αναπτύξαμε επιτυχώς μια πρωτότυπη πιλοτική υπηρεσία ιστού για την παράλληλη στοχαστική προσομοίωση βιολογικών συστημάτων μέσω διαδικτύου. Η υπηρεσία επιτρέπει σε απομακρυσμένους χρήστες να υποβάλουν βιομοντέλα βιολογικών δικτύων τα οποία βρίσκονται σε SBML (Systems Biology Markup Language) μορφή, μαζί με επιθυμητές παραμέτρους για στοχαστική προσομοίωση αυτών πάνω στα συστήματα σε ψηφίδα (System on Chip – SoC) υλοποιημένα για συσκευές FPGA που αναπτύξαμε για την Ε.Ε.1. Τα υποβληθέντα αρχεία SBML αναλύονται και στη συνέχεια εκκινούν στοχαστικές προσομοιώσεις στο FPGA υλικό, είτε σε SSIP (Single Simulation in Parallel) είτε σε MSIP (Multiple Simulations In Parallel) τύπο λειτουργίας. Μετά την ολοκλήρωση της προσομοίωσης, ο διακομιστής (server) επιστρέφει τα αποτελέσματα στους χρήστες για περαιτέρω ανάλυση.

Ένας τέτοιος διακομιστής δεν υπήρχε μέχρι σήμερα και αποτελεί παγκόσμια πρωτοτυπία και πολύ χρήσιμη υπηρεσία για τη επιστημονική κοινότητα της συστημικής βιολογίας καθώς επιτρέπει την πολύ αποδοτική και επιταχυνόμενη στοχαστική προσομοίωση αληθινών βιομοντέλων σε μορφή SBML χωρίς ο χρήστης να χρειαστεί να ρυθμίσει, πόσο μάλλον να διαθέτει γρήγορους πολύ-νηματικούς επεξεργαστές ή σύμπλεγμα αυτών ή κατάλληλο SoC στοχαστικής προσομοίωσης πάνω στο FPGA υλικό. Όπως δείξαμε και στη Δράση 1.3, τα SoC στοχαστικής προσομοίωσης που σχεδιάσαμε και υλοποιήσαμε για συσκευές FPGA εκτελούν προσομοιώσεις βιολογικών συστημάτων πολύ πιο αποδοτικά από οποιοδήποτε άλλο λογισμικό προσομοίωσης όπως π.χ το COPASI. Η υπηρεσία ιστού που δημιουργήθηκε δοκιμάστηκε με επιτυχία από την ομάδα του έργου με ποικιλία SBML βιομοντέλων που είχαν ήδη χρησιμοποιηθεί και στις Δράσεις 1.3 και 2.2.

Οι βελτιώσεις που μπορούν να γίνουν σε ένα τόσο μεγάλο σύστημα υλικού και λογισμικού είναι πάρα πολλές. Κάθε ένα από τα συστατικά μέρη του θα μπορούσε να υποστηρίξει περισσότερες και ισχυρότερες λειτουργίες που θα δώσουν στο χρήστη τη δυνατότητα να εκτελεί καλύτερες στοχαστικές προσομοιώσεις. Ιδιαίτερη βαρύτητα θα θέλαμε να δώσουμε και στη μέγιστη σταθερότητα (stability) και στιβαρότητα (robustness) του συστήματος ώστε να μην είναι καθόλου επιρρεπές σε λάθη και σε ακυρώσεις των προσομοιώσεων. Ακόμα και στην περίπτωση που κάτι συμβεί θα πρέπει να είναι σε θέση να το διορθώσει (resilience) ή έστω να παρουσιάσει στο χρήστη το ακριβές πρόβλημα και βήματα για να το ξεπεράσει.

Επειδή από τη φύση της μια υπηρεσία ιστού θα πρέπει να είναι εύχρηστη και φιλική προς το χρήστη, πρώτο μας μέλημα κατά τη βελτίωση της θα ήταν ο επανασχεδιασμός του γραφικού περιβάλλοντος που παρουσιάζεται στο χρήστη. Οι όψεις (views / templates) που δημιουργήθηκαν είναι αρκετά στατικές και το σύγχρονο διαδικτυακό περιβάλλον έχει ήδη περάσει στην επόμενη εποχή (Web 2.0) ενώ ήδη σχεδιάζεται το επόμενο που θα φέρει ακόμα περισσότερες νέες τεχνολογίες άρα και δυνατότητες. Συνεπώς θα επιλέγαμε μεγαλύτερη δυναμικότητα στις όψεις με χρήση περισσότερων frontend τεχνολογιών που βασίζονται σε βαριά χρήση της Javascript που είναι ή γλώσσα επίκεντρο όλων αυτών των τεχνολογιών που αλληλεπιδρούν απευθείας με το χρήστη μιας υπηρεσίας ιστού.

Για παράδειγμα τα αποτελέσματα, πέραν από ένα συμπιεσμένο αρχείο που έχει αυτή τη στιγμή τη δυνατότητα να κατεβάσει στον προσωπικό του υπολογιστή ο χρήστης της υπηρεσίας, θα μπορούσαν να του παρουσιάζονται και γραφικά με ένα πολύ εύχρηστο interface και κατάλληλα γραφήματα (plots) των μοριακών πληθυσμών ανά πάσα χρονική στιγμή, πλήρη στατιστικά της προσομοίωσης και άλλα πολλά που θα έκαναν την ανάλυση των δεδομένων πιο κατανοητή και εύληπτη από το χρήστη.

Η πιο δύσκολη αλλά και πιο ενδιαφέρουσα επέκταση που θα επιλέγαμε να σχεδιάσουμε και να υλοποιήσουμε είναι η δυνατότητα να υποστηριχθούν περισσότεροι του ενός κόμβοι εργάτες με συσκευές FPGA, πιθανώς διαφορετικού τύπου ενώ το ιδανικό θα ήταν να υποστηρίζονται και άλλοι τύποι backend όπως πολύ-νηματικοί ή και πολύ-πύρηνιοι (multi or many-core) επεξεργαστές ή GPU επιταχυντές. Η ομάδα μας έχει ήδη αναπτύξει στοχαστικούς προσομοιωτές για τους πρώτους στο πλαίσιο της Ε.Ε.2 [26], ενώ ήδη βρίσκονται σε φάση διερεύνησης αντίστοιχοι για τους δεύτερους, οπότε είναι πλέον θέμα προσεκτικής μελέτης και σωστής ενοποίησης με το υπόλοιπο σύστημα της υπηρεσίας ιστού που παρουσιάστηκε στο παρόν έγγραφο.

Τέλος είναι αξιοσημείωτο ότι η ομάδα μας κατάφερε να ολοκληρώσει στο ακέραιο και να παραδώσει εντός των προθεσμιών που της ανατέθηκαν όλα τα παραδοτέα που ανέλαβε, στα οποία επιτεύχθηκαν όχι μόνο οι αρχικοί στόχοι όπως περιγράφονται στο τεχνικό παράρτημα αλλά και παραπάνω όπου αυτό χρειάστηκε και ήταν απαραίτητο για την κομψότητα της σχεδίασης, την ποιότητα της υλοποίησης και την αρτιότητα του αποτελέσματος συνολικά ώστε να είναι ανταγωνιστικό και να βελτιώσει σημαντικά τον αντίστοιχο ερευνητικό τομέα. Στα παραδοτέα μας, όπως το παρόν έγγραφο, περιγράφονται συστήματα υλικού και λογισμικού τα οποία διακρίνονται από διεθνή πρωτοτυπία και τα οποία ολοκληρώσαμε με πρώτο μας μέλημα την υψηλή ποιότητα και τη δυνατότητα αυτά να διακριθούν τόσο στον ακαδημαϊκό τομέα με δημοσιεύσεις υψηλού επιπέδου, περισσότερες από μια που ήταν ο αρχικός στόχος, όσο και στη συνείδηση των ατόμων που μέχρι τώρα τα έχουν αξιολογήσει τεχνικά στα αντίστοιχα συνέδρια των δημοσιεύσεων ή και χρησιμοποιήσει σε πιλοτική λειτουργία.

Είμαστε βέβαιοι ότι το επίπεδο των δυναμικών Ελληνικών επιστημονικών ομάδων όπως η δικιά μας με νέους επιστήμονες που συμμετείχαν στο πρόγραμμα ΑΡΙΣΤΕΙΑ ΙΙ είναι εξίσου υψηλό, αν όχι υψηλότερο, από αντίστοιχες ομάδες του εξωτερικού. Ελπίζουμε ότι αρκετές από τις μελέτες που ολοκληρώθηκαν με επιτυχία, με την υποστήριξη της πολιτείας θα γίνουν η αρχή για ακόμα μεγαλύτερη ερευνητική δραστηριότητα και καινοτομία στο άμεσο μέλλον, κάτι που αναπόφευκτα θα οδηγήσει σε νέες διακρίσεις αλλά και ποιοτικές θέσεις εργασίας για ανήσυχους ερευνητές που τολμούν σε τομείς τεχνολογιών αιχμής για το κοινό καλό και που τόσο έχει ανάγκη η χώρα μας να συγκρατήσει αυτή τη δύσκολη περίοδο που διανύει.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] E. Kent, S. Hoops και P. Mendes, «Condor-COPASI: high-throughput computing for biochemical networks,» *BMC Systems Biology*, τόμ. 6, αρ. 1, pp. 1-13, 2012.
- [2] C. Horuk, G. Douglas, A. Gupta, C. Krintz, B. Bales, G. Bellesia, B. Drawert, R. Wolski, L. Petzold και A. Hellander, «Automatic and Portable Cloud Deployment for Scientific Simulations,» σε *High Performance Computing Simulation (HPCS)*, 2014.
- [3] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, S. Natalia, M. Singhal, L. Xu, P. Mendes και U. Kummer, «COPASI—a complex pathway simulator,» *Bioinformatics*, τόμ. 22, pp. 3067-3074, 2006.
- [4] T. Tannenbaum, D. Wright, K. Miller και M. Livny, «Condor - A Distributed Job Scheduler,» σε *Beowulf Cluster Computing with Linux*, T. Sterling, Επιμ., 2011.
- [5] K. Sanft, S. Wu, M. Roh, J. Fu, R. Lim και L. Petzold, «StochKit2: software for discrete stochastic simulation of biochemical systems with events,» *Bioinformatics*, τόμ. 27, αρ. 17, pp. 2457-2458, 2011.
- [6] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. Doyle, H. Kitano, A. Arkin, B. Bornstein, D. Bray και A. Cornish-Bowden, «The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models,» *Bioinformatics*, τόμ. 19, pp. 524-531, 2003.
- [7] «Open SSH,» [Ηλεκτρονικό]. Available: <http://www.openssh.com>. [Πρόσβαση 29 05 2015].
- [8] «Symfony Framework,» Sensio Labs, [Ηλεκτρονικό]. Available: <http://symfony.com/>. [Πρόσβαση 29 05 2015].
- [9] «PHP,» Zend, [Ηλεκτρονικό]. Available: <http://www.php.net>. [Πρόσβαση 29 05 2015].
- [10] «MySQL,» Oracle, [Ηλεκτρονικό]. Available: <http://www.mysql.com>. [Πρόσβαση 29 05 2015].
- [11] «Doctrine,» [Ηλεκτρονικό]. Available: <http://www.doctrine-project.org>. [Πρόσβαση 29 05 2015].
- [12] «Apache HTTP Server,» Apache, [Ηλεκτρονικό]. Available: <http://httpd.apache.org>. [Πρόσβαση 29 05 2015].
- [13] «Virtual Box,» Oracle, [Ηλεκτρονικό]. Available: <https://www.virtualbox.org>. [Πρόσβαση 29 05 2015].
- [14] «Vagrant,» Hashi Corp, [Ηλεκτρονικό]. Available: <https://www.vagrantup.com>. [Πρόσβαση 29 05 2015].
- [15] «Puppet,» Puppet Labs, [Ηλεκτρονικό]. Available: <https://www.puppetlabs.com>. [Πρόσβαση 29 05 2015].
- [16] K. Rarick, «Beanstalk,» [Ηλεκτρονικό]. Available: <http://kr.github.io/beanstalkd>. [Πρόσβαση 29 05 2015].
- [17] «libSSH2,» [Ηλεκτρονικό]. Available: <http://www.libssh2.org>. [Πρόσβαση 29 05 2015].

- [18] «libCURL,» [Ηλεκτρονικό]. Available: <http://curl.haxx.se/libcurl>. [Πρόσβαση 29 05 2015].
- [19] «GNU Compiler Collection,» Free Software Foundation, [Ηλεκτρονικό]. Available: <https://gcc.gnu.org>. [Πρόσβαση 29 05 2015].
- [20] «CMake,» Kitware, [Ηλεκτρονικό]. Available: <https://www.cmake.org>. [Πρόσβαση 29 05 2015].
- [21] «Bii Code,» [Ηλεκτρονικό]. Available: <https://www.biicode.com>. [Πρόσβαση 29 05 2015].
- [22] Xilinx, «Xilinx Inc,» [Ηλεκτρονικό]. Available: <http://www.xilinx.com>. [Πρόσβαση 29 05 2015].
- [23] Xilinx, «ISE Design Suite,» Xilinx Inc, [Ηλεκτρονικό]. Available: <http://www.xilinx.com/products/design-tools/ise-design-suite.html>. [Πρόσβαση 01 Jul 2011].
- [24] Xilinx, «iMpac Overview,» Xilinx Inc, [Ηλεκτρονικό]. Available: http://www.xilinx.com/itp/xilinx10/isehelp/pim_c_overview.htm. [Πρόσβαση 01 Jul 2012].
- [25] M. Matsumoto και T. Nishimura, «Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator,» *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, τόμ. 8, αρ. ACM, pp. 3-30, Jan. 1998.
- [26] E. Kouskoumvekakis, D. Soudris και E. S. Manolakos, «Many-core CPUs can deliver scalable performance to stochastic simulations,» σε *International Conference on High Performance Computing & Simulation*, pp 517-524, 2015.