ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Παράλληλη και κατανεμημένη εκπαίδευση βαθιών νευρωνικών δικτύων

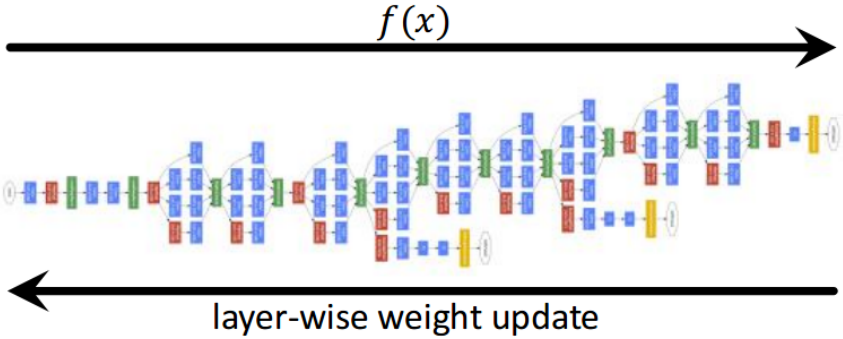
Ακαδημαϊκό έτος 2019-20

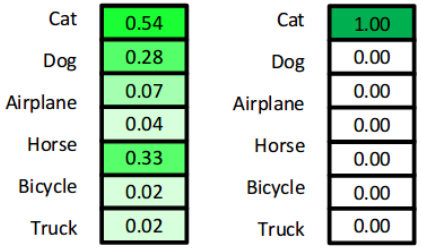
Εισαγωγή

- Βαθιά νευρωνικά δίκτυα και ανάγκη για κατανεμημένη επεξεργασία
- Παράλληλη επεξεργασία σε συστοιχίες: επικοινωνία
- Παραλληλισμός στα βαθιά νευρωνικά δίκτυα
- Τεχνικές μείωσης του χρόνου επικοινωνίας σε παράλληλα βαθιά νευρωνικά δίκτυα

Βαθιά νευρωνικά δίκτυα και ανάγκη για κατανεμημένη επεξεργασία

Βαθιά νευρωνικά δίκτυα





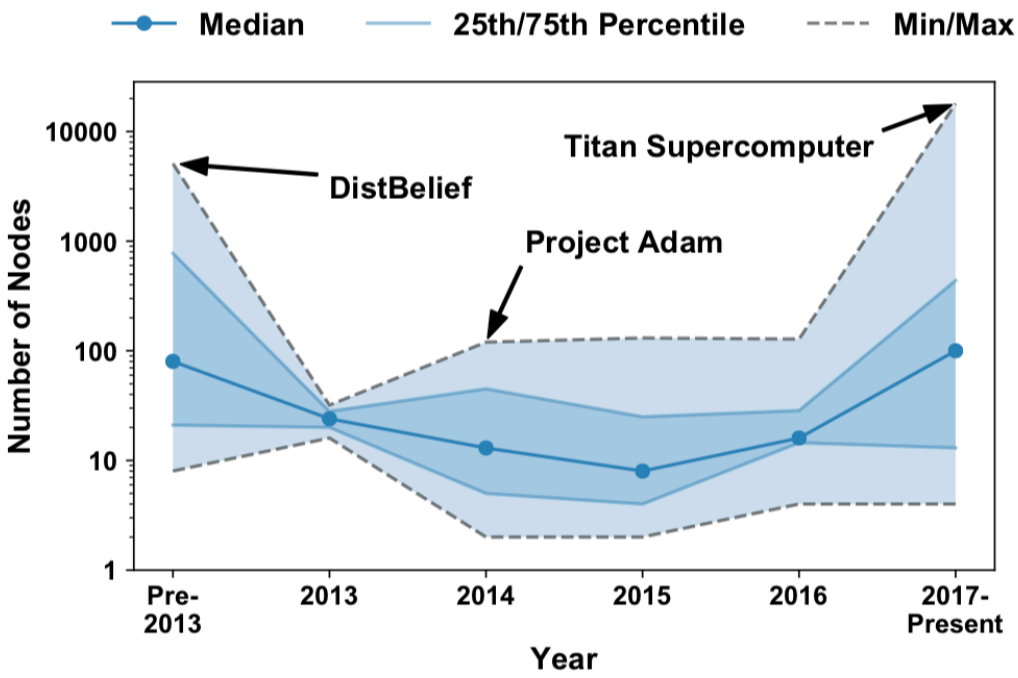
- ImageNet (1K): ~GB
- ImageNet (22K): ~TB

- 100-200 επίπεδα
- ~100M-2B παράμετροι
- 0.1-8GB για τις παραμέτρους

- 10-22k ετικέτες
- απαιτούν εβδομάδες για την εκτέλεση της εκπαίδευσης

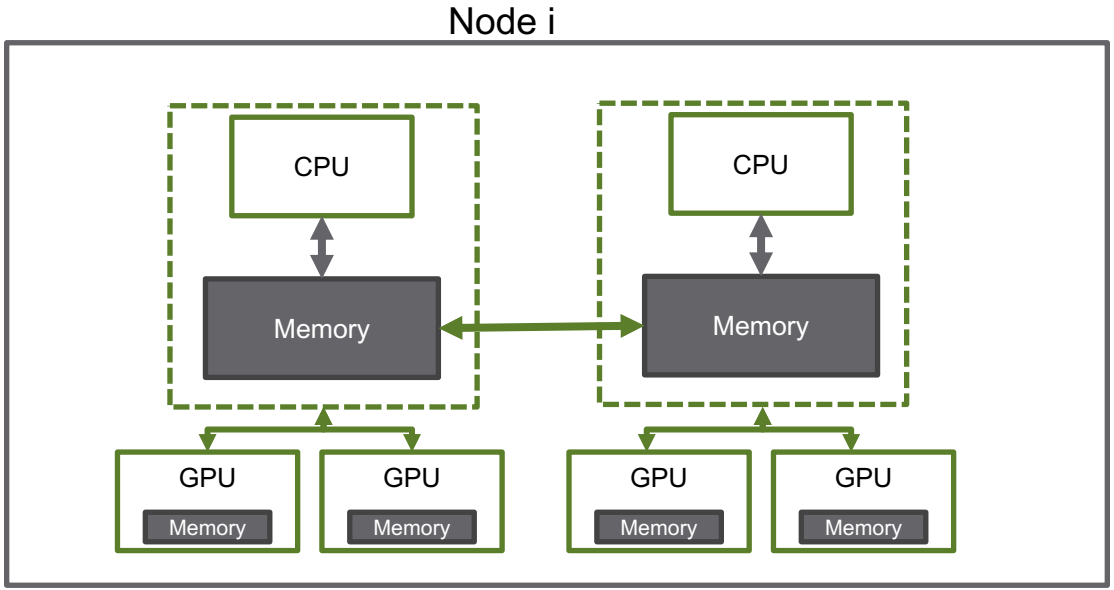
Όταν η υπολογιστική ισχύς ενός κόμβου δεν είναι αρκετή

- Η ανάγκη για μείωση του χρόνο εκπαίδευσης οδηγεί σε χρήση πολλαπλών κόμβων για την εκπαίδευση βαθιών νευρωνικών δικτύων
- Η τάση αυτή αυξάνεται τα τελευταία χρόνια



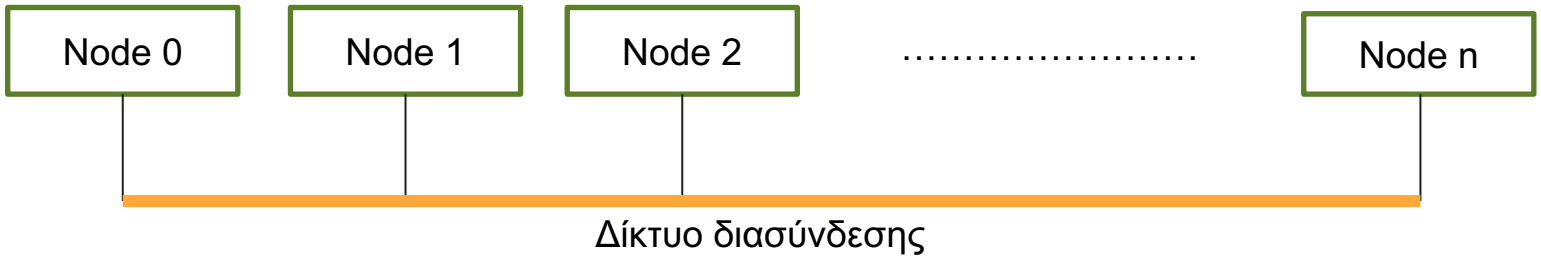
Σύγχρονες συστοιχίες (clusters/supercomputers)

- Πολλοί παράλληλοι υπολογιστικοί κόμβοι
  - Σύγχρονοι υπολογιστικοί κόμβοι: πολλαπλές CPUs, κοινή μνήμη (UMA ή NUMA), ένας ή περισσότεροι επιταχυντές



Σύγχρονες συστοιχίες (clusters/supercomputers)

- Πολλοί παράλληλοι υπολογιστικοί κόμβοι
- Κάθε κόμβος έχει τη δική του μνήμη
- Η μνήμη ενός κόμβου δεν είναι «ορατή» από άλλους κόμβους
  - Δεν είναι μοιραζόμενη
- Οι κόμβοι διασυνδέονται με κάποιο δίκτυο διασύνδεσης υψηλής επίδοσης



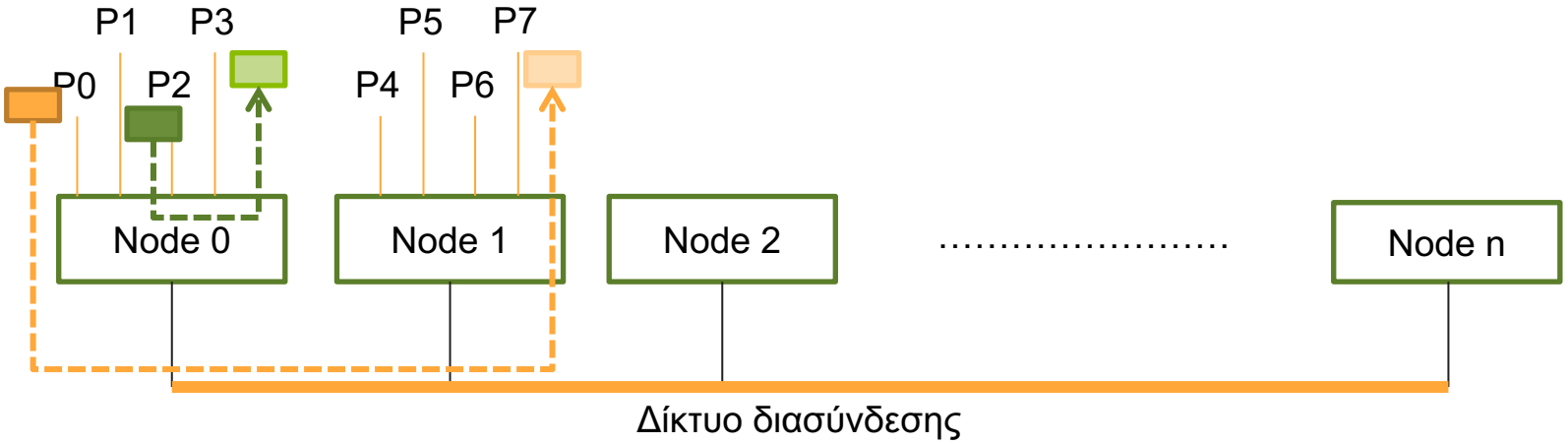
Σύγχρονες συστοιχίες

- Οι συστοιχίες υπολογιστών είναι αρχιτεκτονικές κατανεμημένης μνήμης
- Αλλαγές/Ενημερώσεις στη μνήμη ενός κόμβου δεν είναι ορατές σε άλλους κόμβους
- Η **επικοινωνία** μεταξύ των κόμβων γίνεται μέσω του δικτύου διασύνδεσης με ανταλλαγή μηνυμάτων
- Μια εφαρμογή μπορεί να αξιοποιήσει όλη τη διαθέσιμη υπολογιστική ισχύ και κατανεμημένη μνήμη με το κατάλληλο προγραμματιστικό μοντέλο

Παράλληλη επεξεργασία σε συστοιχίες: επικοινωνία

Σύγχρονες συστοιχίες και επικοινωνία

- Σε κάθε κόμβο εκτελούνται εργάτες/διεργασίες του παράλληλου προγράμματος
- Εργάτες που εκτελούνται στον ίδιο κόμβο μπορούν να αξιοποιήσουν την κοινή μνήμη
- Εργάτες που εκτελούνται σε διαφορετικο κόμβο πρέπει να ανταλλάξουν μηνύματα για ανταλλαγή δεδομένων

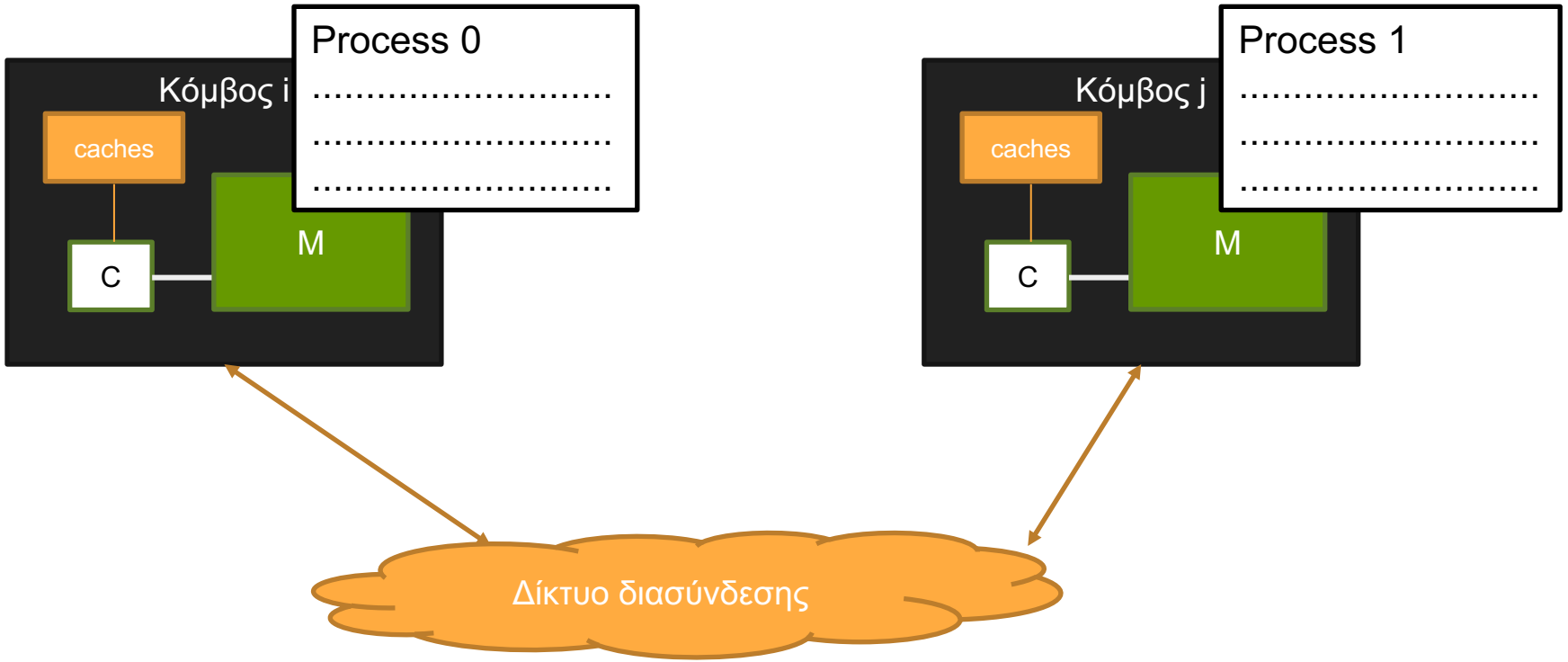


Προγραμματισμός σε κατανεμημένο χώρο διευθύνσεων

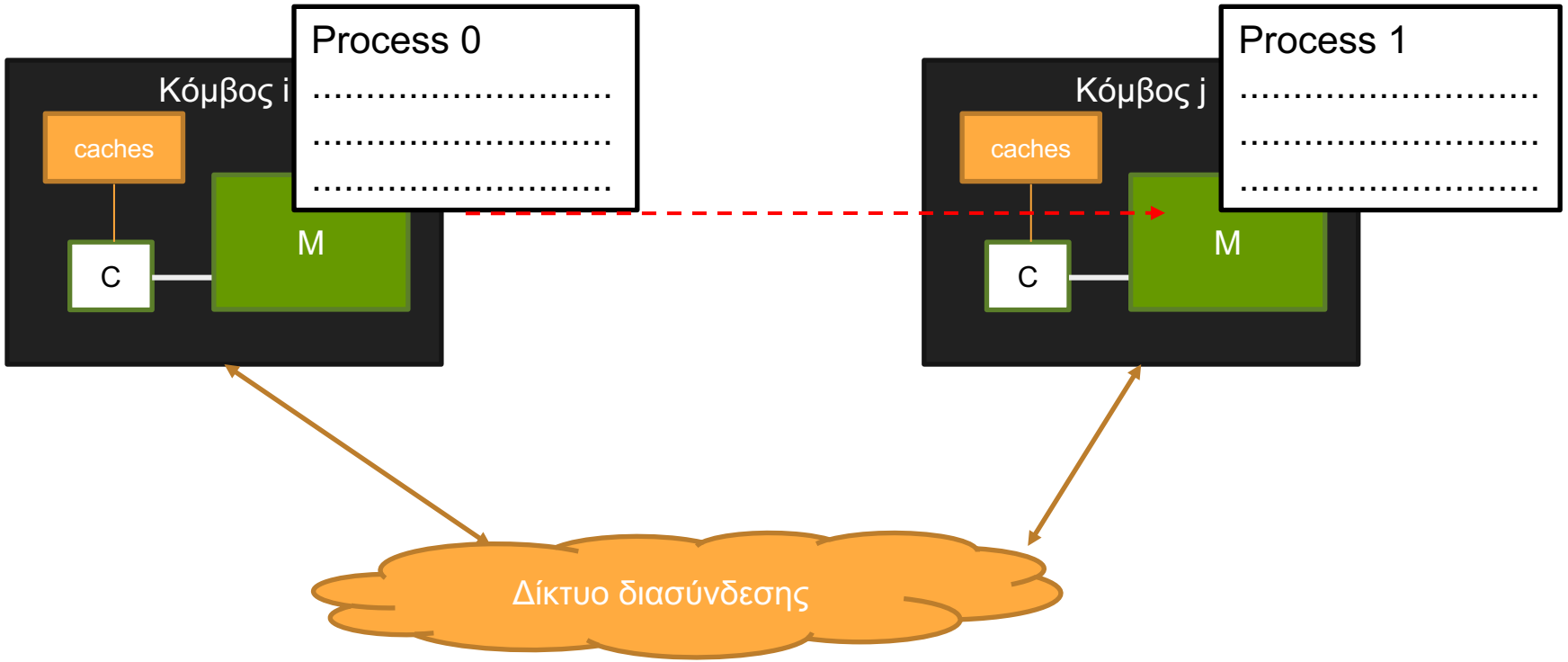
- Προγραμματιστικά μοντέλα:
  - Message Passing Interface (MPI) κ.ά.
- Η πρόσβαση σε δεδομένα που κατέχει άλλη διεργασία γίνεται με ανταλλαγή μηνυμάτων
  - Ρητές κλήσεις σε συναρτήσεις αποστολής / λήψης

- Δύσκολος προγραμματισμός!
  - Ο προγραμματιστής πρέπει να ορίσει ρητά τη μεταφορά δεδομένων μεταξύ διεργασιών

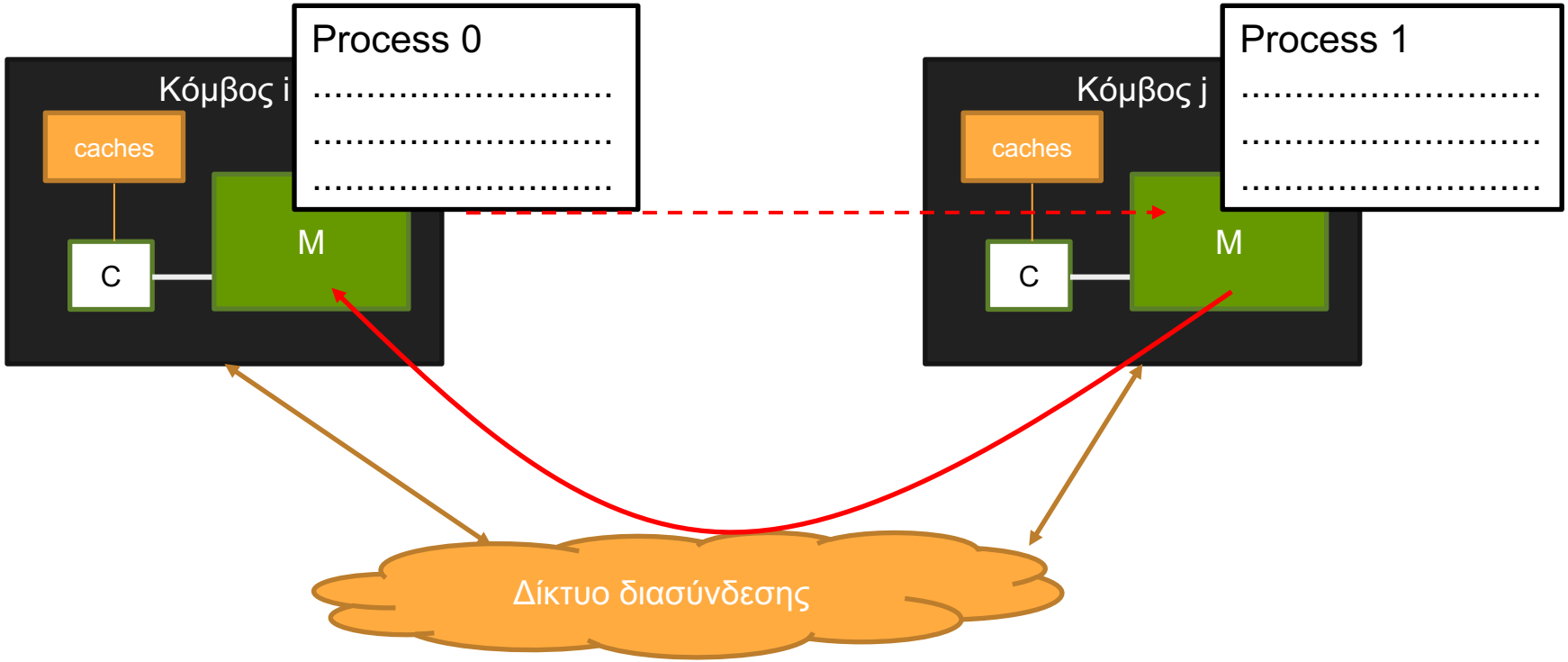
Ανταλλαγή μηνυμάτων στο MPI



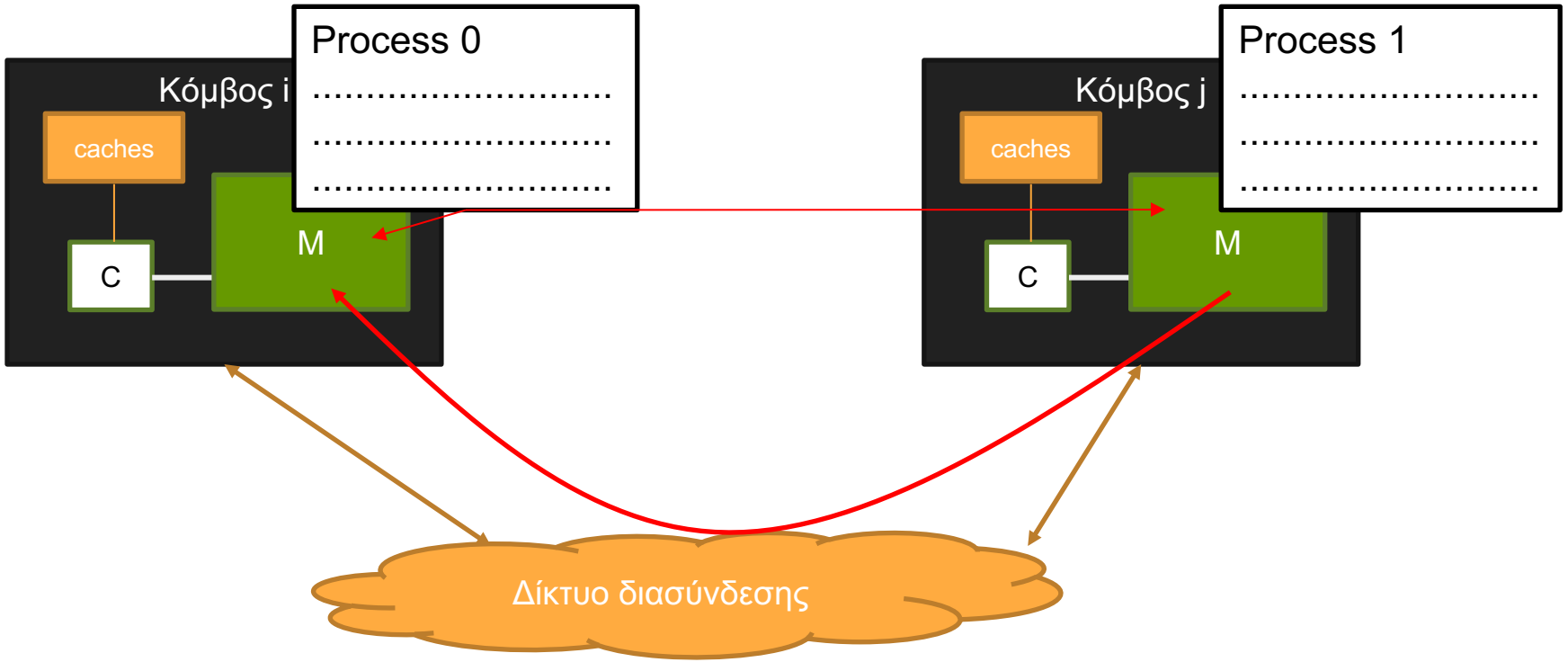
Ανταλλαγή μηνυμάτων στο MPI



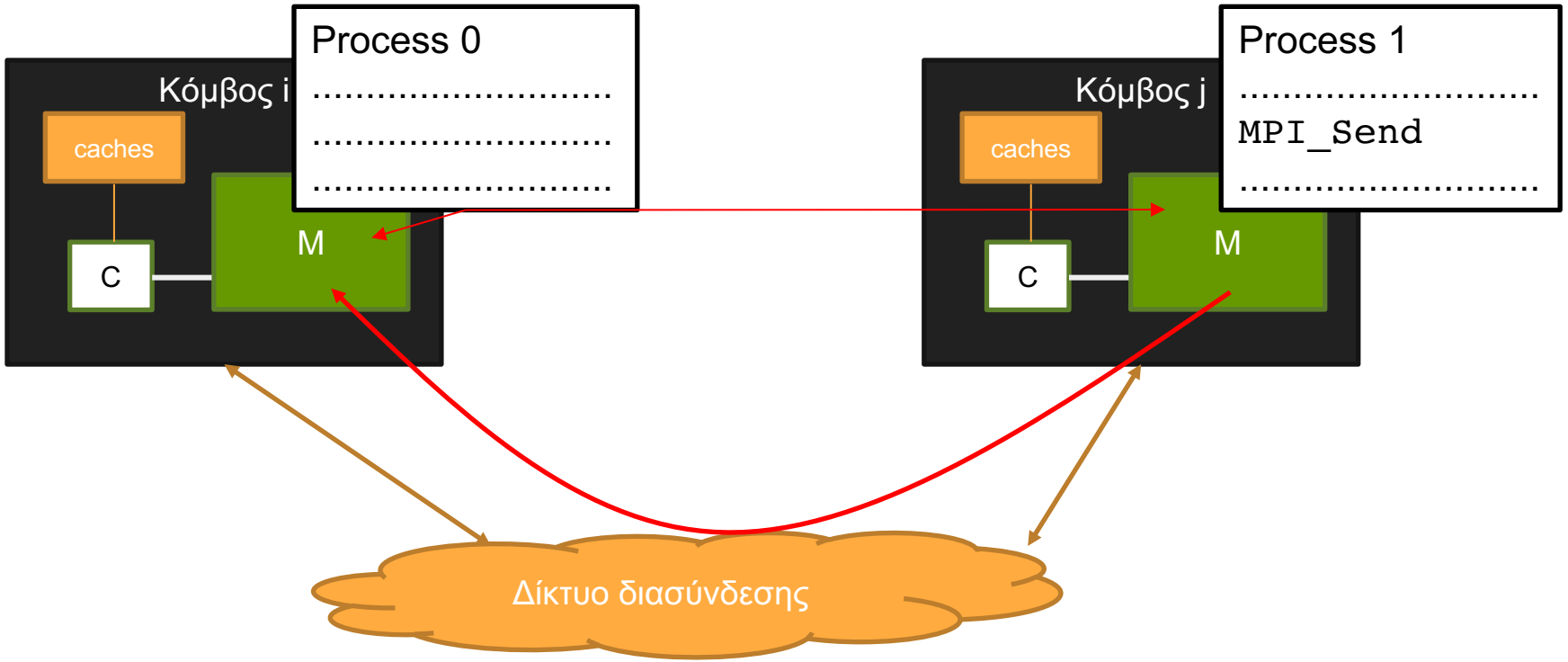
Ανταλλαγή μηνυμάτων στο MPI



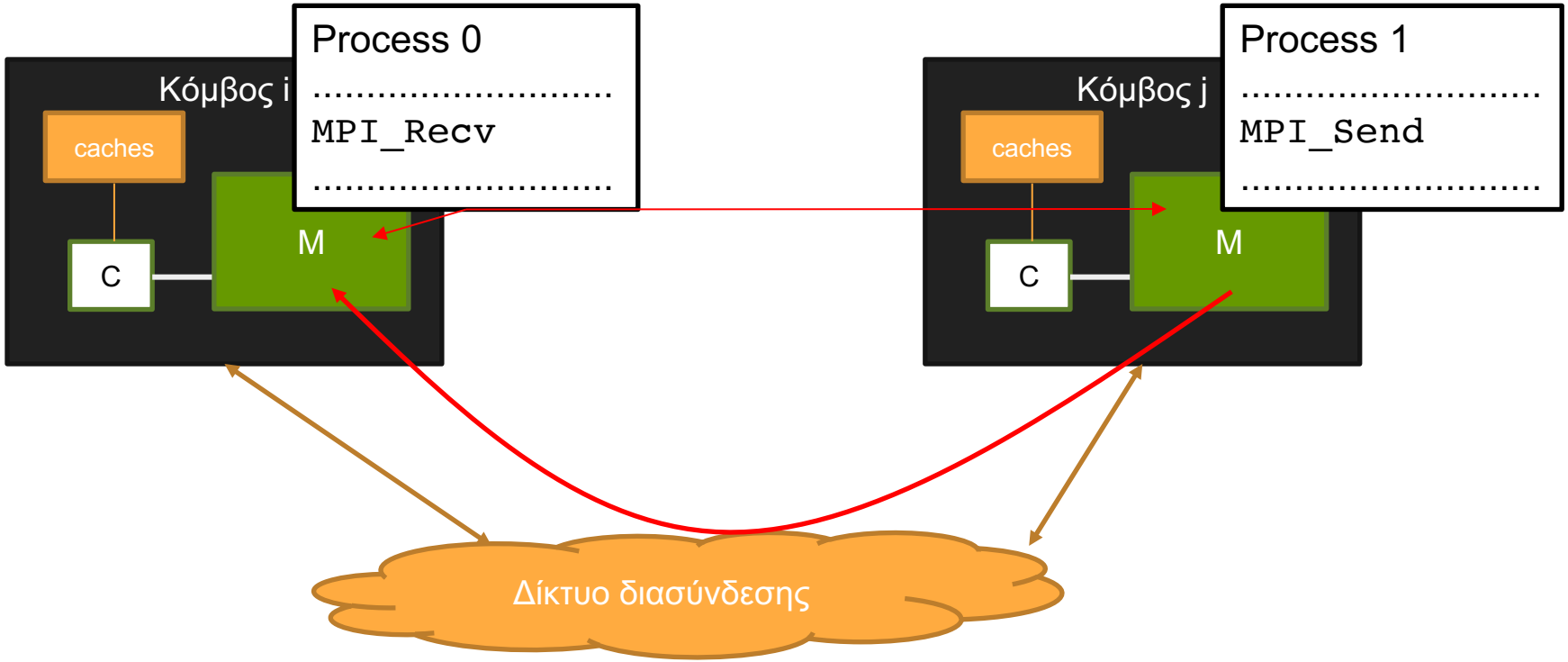
Ανταλλαγή μηνυμάτων στο MPI



Ανταλλαγή μηνυμάτων στο MPI



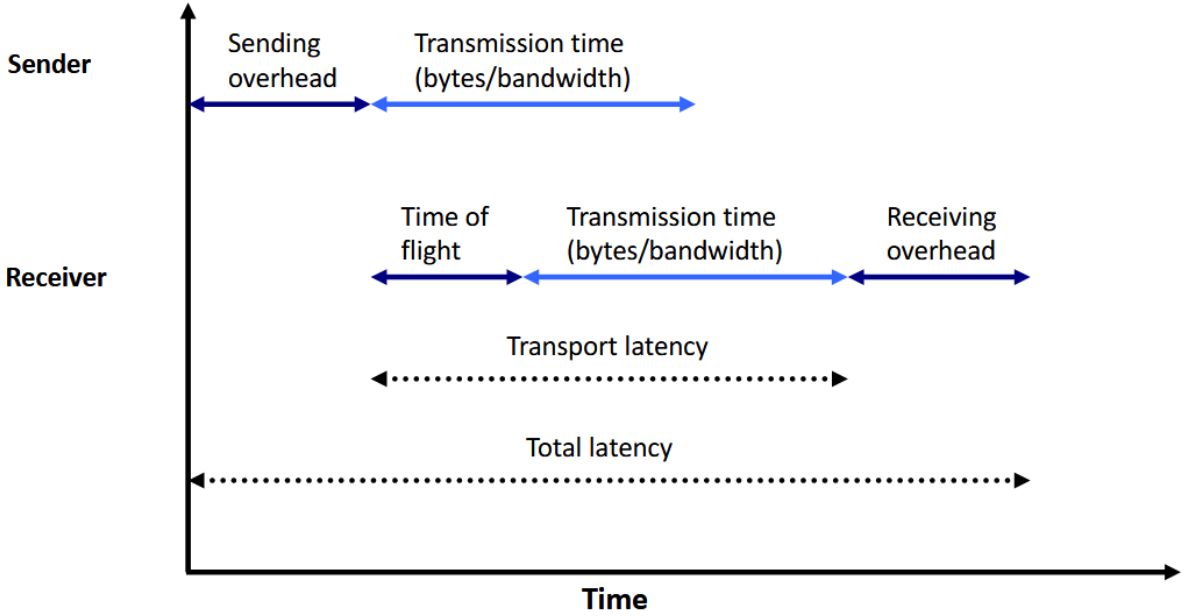
Ανταλλαγή μηνυμάτων στο MPI



Δίκτυα διασύνδεσης και κόστος επικοινωνίας

- Point-to-point επικοινωνία
  - Μία διεργασία στέλνει ένα μήνυμα (send)
  - Μία άλλη διεργασία λαμβάνει ένα μήνυμα (receive)
  - Ασυμπτωτικό κόστος: O(1)
  - Πραγματικό κόστος: εξαρτάται από τα χαρακτηριστικά του δικτύου
    - Χρόνος απόκρισης (latency - l): ο χρόνος που απαιτείται για τη μεταφορά ενός byte από μία διεργασία σε μια άλλη
    - Εύρος ζώνης (bandwidth - bw): ο ρυθμός μεταφοράς δεδομένων (bytes/sec)

Δίκτυα διασύνδεσης και κόστος επικοινωνίας



- Απλό μοντέλο για το κόστος επικοινωνίας ενός μηνύματος μεγέθους size bytes
  - t = l + size / bw
- Τα σύγχρονα δίκτυα διασύνδεσης προσφέρουν χαμηλό χρόνο απόκρισης και υψηλό εύρος ζώνης
  - Ωστόσο η επικοινωνία είναι αρκετά πιο αργή από την επικοινωνία μέσω κοινής μνήμης

Συλλογική επικοινωνία

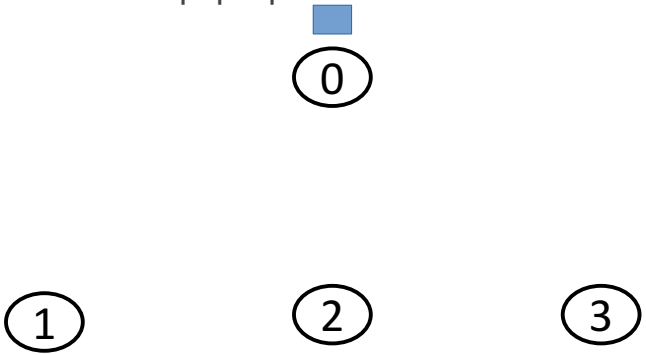
- Στις περισσότερες παράλληλες εφαρμογές, απαιτείται επικοινωνία περισσότερων των ενός διεργασιών
- Αν όλες οι διεργασίες πρέπει να επικοινωνήσουν μεταξύ τους για ανταλλαγή δεδομένων, τότε αναφερόμαστε σε **συλλογική επικοινωνία**

- Διαφορετικά σχήματα συλλογικής επικοινωνίας έχουν διαφορετικό κόστος

- *Παράδειγμα:* Σε μια εφαρμογή με **n** διεργασίες, μία διεργασία πρέπει να στείλει μία τιμή σε όλες τις άλλες διεργασίες (broadcast)
  - Ασυμπτωτικό κόστος: O(**n**), καταλήγει σε O(**logn**) με βελτιστοποίηση
  - Πώς;

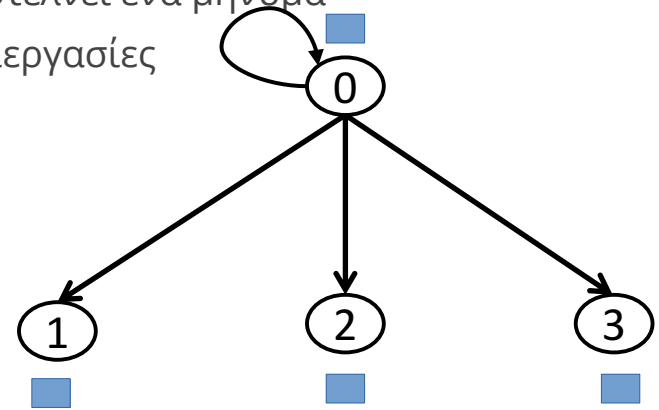
Broadcast

- Broadcast
  - Μία διεργασία στέλνει ένα μήνυμα



Broadcast

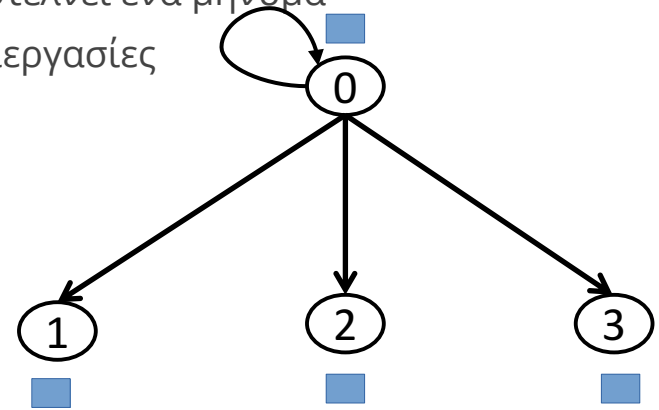
- Broadcast
  - Μία διεργασία στέλνει ένα μήνυμα
  - Όλες οι άλλες διεργασίες λαμβάνουν το ίδιο μήνυμα



- Ασυμπτωτικό κόστος: O(logn)

Broadcast

- Broadcast
  - Μία διεργασία στέλνει ένα μήνυμα
  - Όλες οι άλλες διεργασίες λαμβάνουν το ίδιο μήνυμα



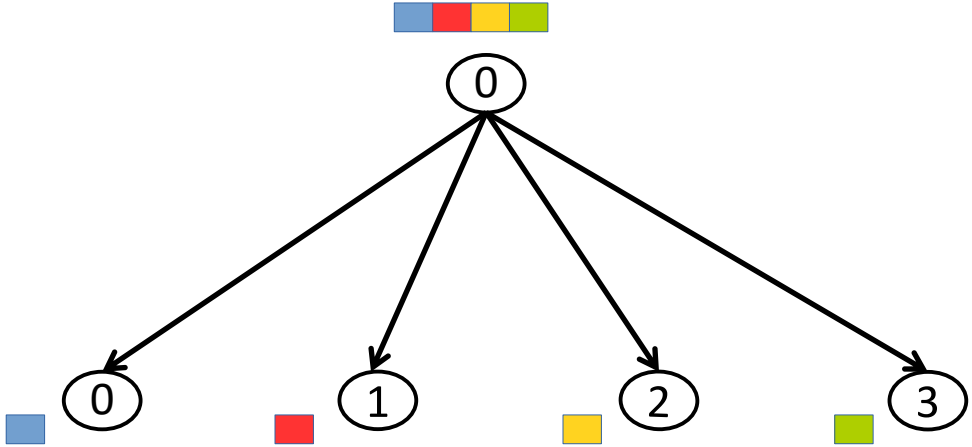
- Ασυμπτωτικό κόστος: O(logn)

Scatter

- Scatter:
  - Μία διεργασία στέλνει ένα μήνυμα

Scatter

- Scatter:
  - Μία διεργασία στέλνει ένα μήνυμα
  - Όλες οι άλλες διεργασίες λαμβάνουν μέρος του μηνύματος (με συγκεκριμένη σειρά)



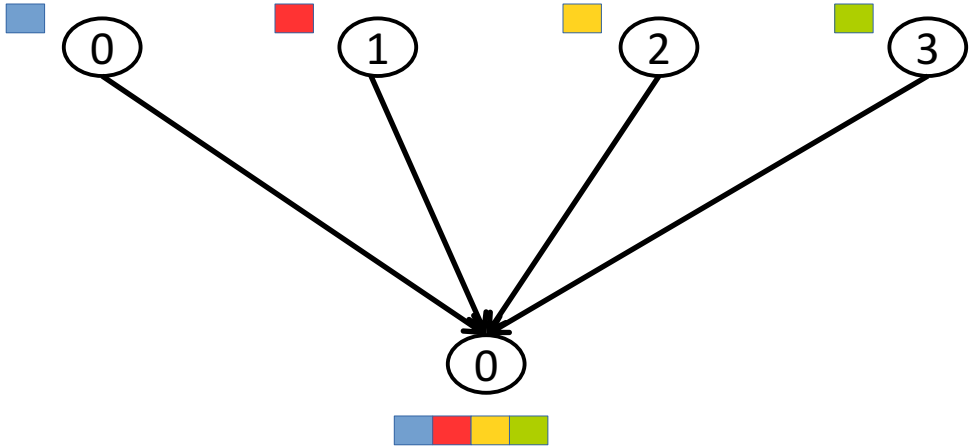
- Κόστος: O(n)

Gather

- Gather:
  - Όλες οι διεργασίες στέλνουν ένα μήνυμα

Gather

- Gather:
  - Όλες οι διεργασίες στέλνουν ένα μήνυμα
  - Μία διεργασία λαμβάνει όλα τα μηνύματα και τα συνενώνει



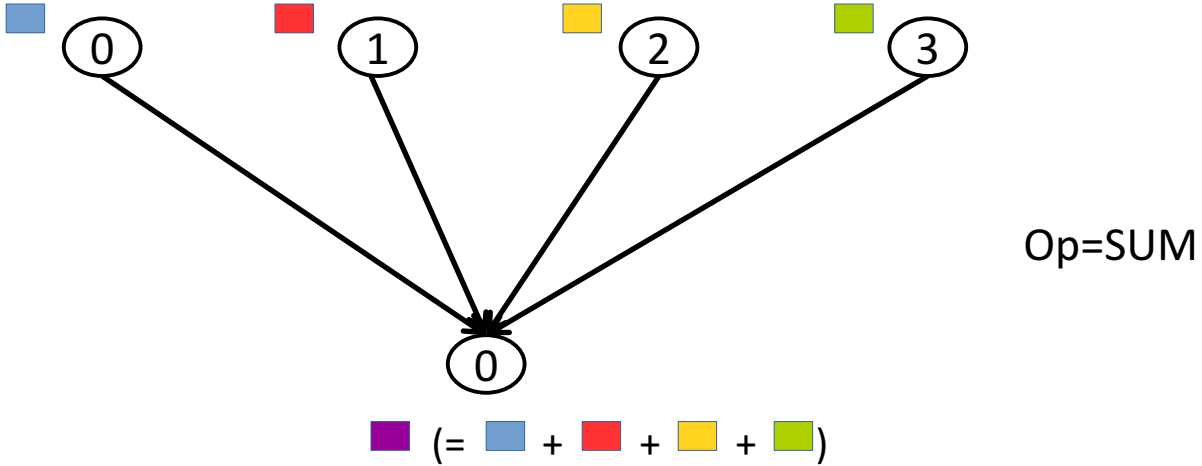
- Κόστος: O(n)

Reduce

- Reduce:
  - Όλες οι διεργασίες στέλνουν ένα μήνυμα

Reduce

- Reduce:
  - Όλες οι διεργασίες στέλνουν ένα μήνυμα
  - Μία διεργασία λαμβάνει όλα τα μηνύματα και εφαρμόζει έναν τελεστή



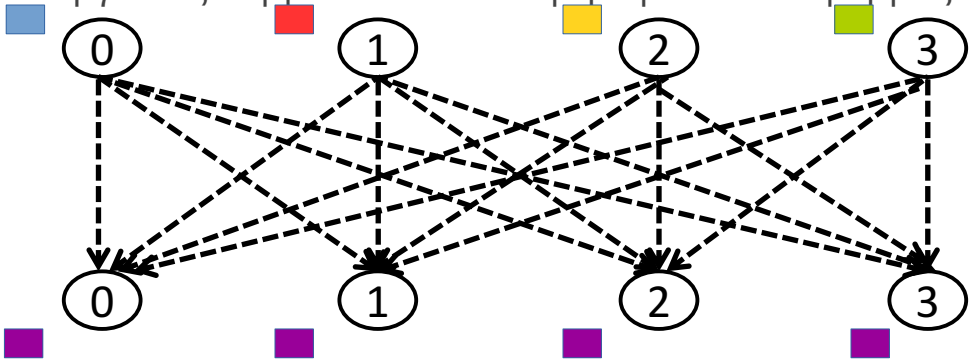
- Κόστος: O(logn)

Allreduce

- Allreduce:
  - Όλες οι διεργασίες στέλνουν ένα μήνυμα

Allreduce

- Allreduce:
  - Όλες οι διεργασίες στέλνουν ένα μήνυμα
  - Όλες οι διεργασίες λαμβάνουν όλα τα μηνύματα και εφαρμόζουν έναν τελεστή



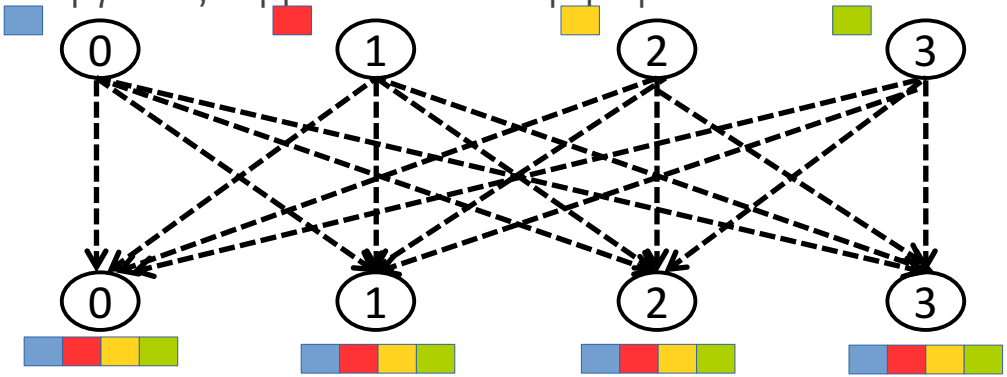
- Κόστος: O(logn)
- Υλοποιείται με Reduce + Broadcast

Alltoall

- Alltoall:
  - Όλες οι διεργασίες στέλνουν ένα μήνυμα

Alltoall

- Alltoall:
  - Όλες οι διεργασίες στέλνουν ένα μήνυμα
  - Όλες οι διεργασίες λαμβάνουν όλα τα μηνύματα

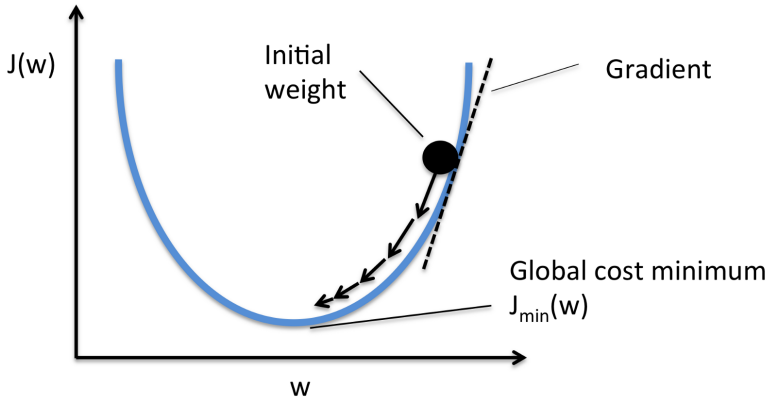


- Κόστος: O(nlogn)

Παραλληλισμός στα βαθιά νευρωνικά δίκτυα

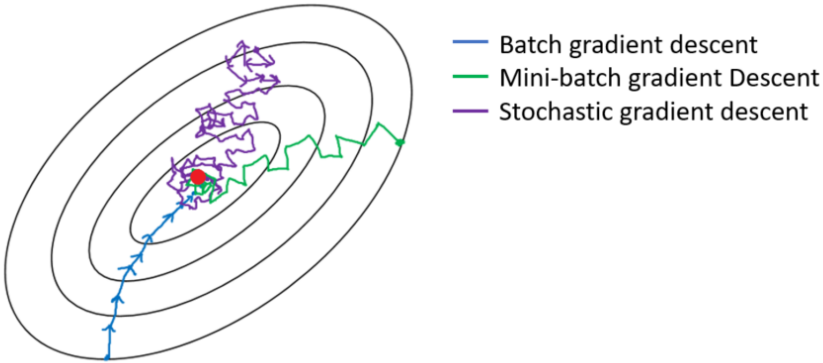
Αλγόριθμος Gradient Descent

- Επαναληπτικός αλγόριθμος βελτιστοποίησης για την ελαχιστοποίηση συνάρτησης
- Χρησιμοποιείται στην εκπαίδευση νευρωνικών δικτύων για την ελαχιστοποίηση της συνάρτησης σφάλματος
- Σε κάθε επανάληψη:
  1. Υπολογίζει την τιμή του σφάλματος για ένα δεδομένο εισόδου (forward pass) και
  2. Υπολογίζει την παράγωγο της συνάρτησης σφάλματος ως προς κάθε παράμετρο του δικτύου και ενημερώνει τις τιμές των παραμέτρων (backward pass)



Αλγόριθμος εκπαίδευσης Gradient Descent

- *Batch (BGD)*: χρήση *όλων* των δεδομένων εισόδου σε ένα πέρασμα
  - Υψηλές απαιτήσεις σε υπολογισμούς και μνήμη
- *Stochastic (SGD)*: χρήση *ενός* δεδομένου εισόδου σε ένα πέρασμα
  - Χαμηλές απαιτήσεις σε υπολογισμούς και μνήμη
- *Mini-Batch (MB-GD)*: χρήση ενός *μικρού συνόλου* δεδομένων εισόδου σε ένα πέρασμα
  - Συμβιβασμός ανάμεσα σε BGD και SGD



Αλγόριθμος εκπαίδευσης Gradient Descent

- Stochastic gradient descent

```
for e in epochs:
     for i in size(TrainingSet):
          sample z^i from TrainingSet
          a^i = feedForward(z^i)
          dW, db = computeGradient(a^i, y^i)
          W, b = updateWeights(dW, db)
```

- Mini-batch gradient descent

```
for e in epochs:
     for i in size(TrainingSet)/B:
          sample z^i:i+B from TrainingSet
          a^i:i+B = feedForward(z^i:i+B)
          dW, db = computeGradient(a^i:i+B, y^i:i+B)
          W, b = updateWeights(dW, db)
```

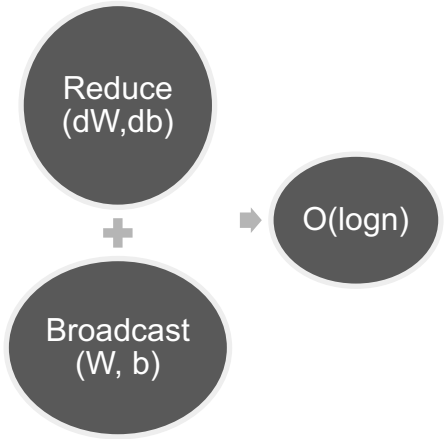
Παραλληλισμός σε επίπεδο δεδομένων εισόδου

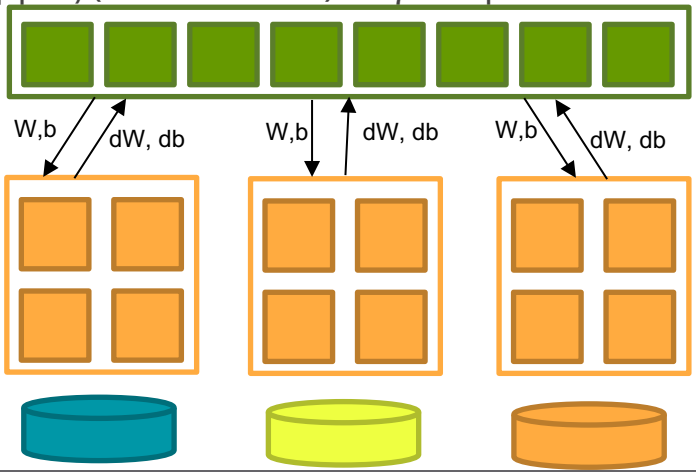
- **Data parallelism**: παραλληλισμός σε επίπεδο δεδομένων εισόδου
  - Κάθε μονάδα επεξεργασίας διατηρεί ένα αντίγραφο του μοντέλου και εκπαιδεύει τις παραμέτρους με ένα υποσύνολο των δεδομένων εισόδου
    - Κάθε μονάδα επεξεργασίας αναλαμβάνει ένα mini-batch

- *Πλεονεκτήματα*:
  - απλή και εύκολη λύση και υλοποίηση
- *Μειονεκτήματα*:
  - κάθε μονάδα επεξεργασίας διατηρεί αντίγραφο του μοντέλου
  - απαιτείται κάποιου είδους επικοινωνία για τα βάρη

Παραλληλισμός σε επίπεδο δεδομένων εισόδου

- **Data parallelism + master node (parameter server)**
  - Κάθε μονάδα επεξεργασίας διατηρεί ένα αντίγραφο του μοντέλου και εκπαιδεύει τις παραμέτρους με ένα υποσύνολο των δεδομένων εισόδου
  - Κάθε μονάδα επεξεργασίας αναλαμβάνει ένα mini-batch
  - Ένας κεντρικός κόμβος (master node) συγκεντρώνει τα αποτελέσματα





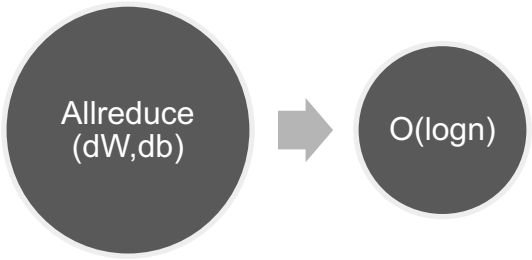
Master node - Parameter server
`W, b = updateWeights(dW, db)`
(Υπολογίζει το μέσο όρο όλων των dW,db)

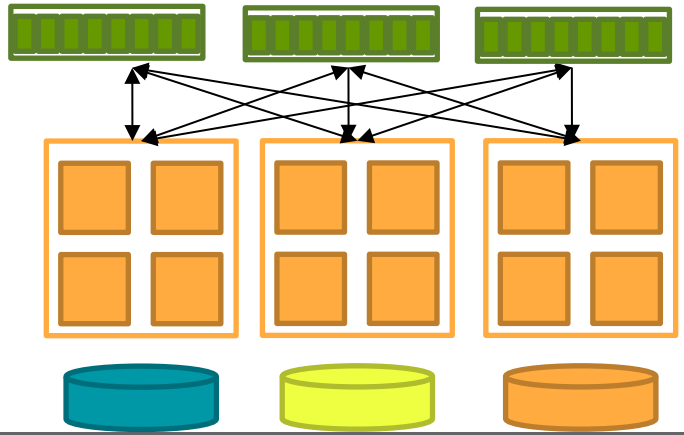
Worker nodes
Αντίγραφα του μοντέλου

Υποσύνολα του συνόλου εκπαίδευσης

Παραλληλισμός σε επίπεδο δεδομένων εισόδου

- **Data parallelism + decentralized**
  - Κάθε μονάδα επεξεργασίας διατηρεί ένα αντίγραφο του μοντέλου και εκπαιδεύει τις παραμέτρους με ένα υποσύνολο των δεδομένων εισόδου
  - Κάθε μονάδα επεξεργασίας αναλαμβάνει ένα mini-batch
  - Κάθε μονάδα επεξεργασίας διατηρεί έναν parameter server!





Worker nodes – parameter servers

```
Allreduce(dW,db)
W, b = updateWeights(dW, db)
```

Worker nodes
Αντίγραφα του μοντέλου

Υποσύνολα
του συνόλου εκπαίδευσης

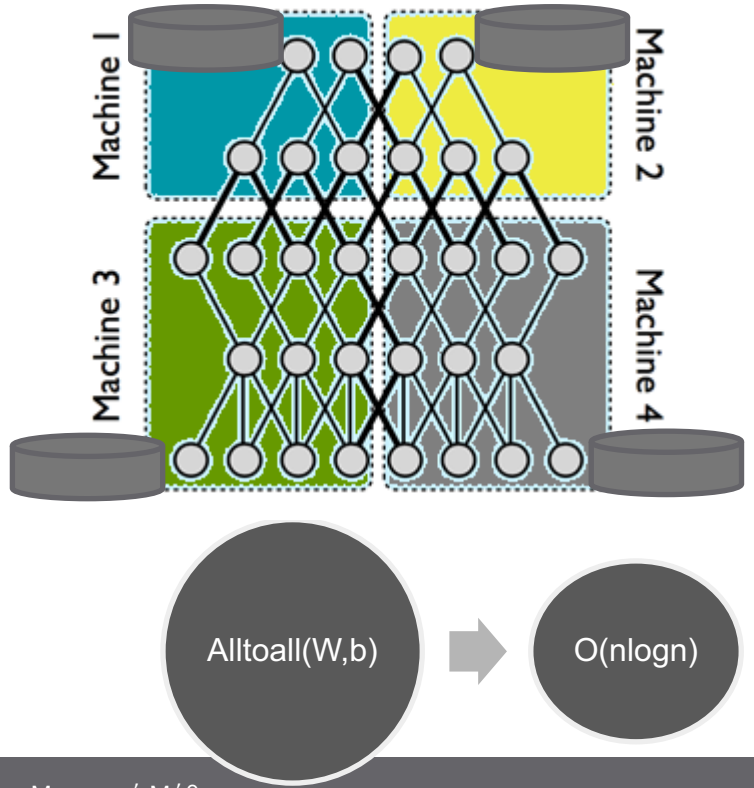
Παραλληλισμός σε επίπεδο μοντέλου

- **Model parallelism**: παραλληλισμός σε επίπεδο δεδομένων εισόδου
  - Κάθε μονάδα επεξεργασίας διατηρεί κομμάτι του μοντέλου (υποσύνολο των παραμέτρων)
  - Όλες οι μονάδες επεξεργασίας εκπαιδεύουν ταυτόχρονα το δικό τους κομμάτι του μοντέλου με το ίδιο υποσύνολο δεδομένων (mini-batch)

- *Πλεονεκτήματα*:
  - Επιτρέπει την εκπαίδευση αν το μοντέλο είναι μεγάλο (δε χωράει στη μνήμη ενός κόμβου)
- *Μειονεκτήματα*:
  - κάθε μονάδα επεξεργασίας διατηρεί αντίγραφο του τρέχοντος mini-batch
  - απαιτείται επικοινωνία κατά το back propagation

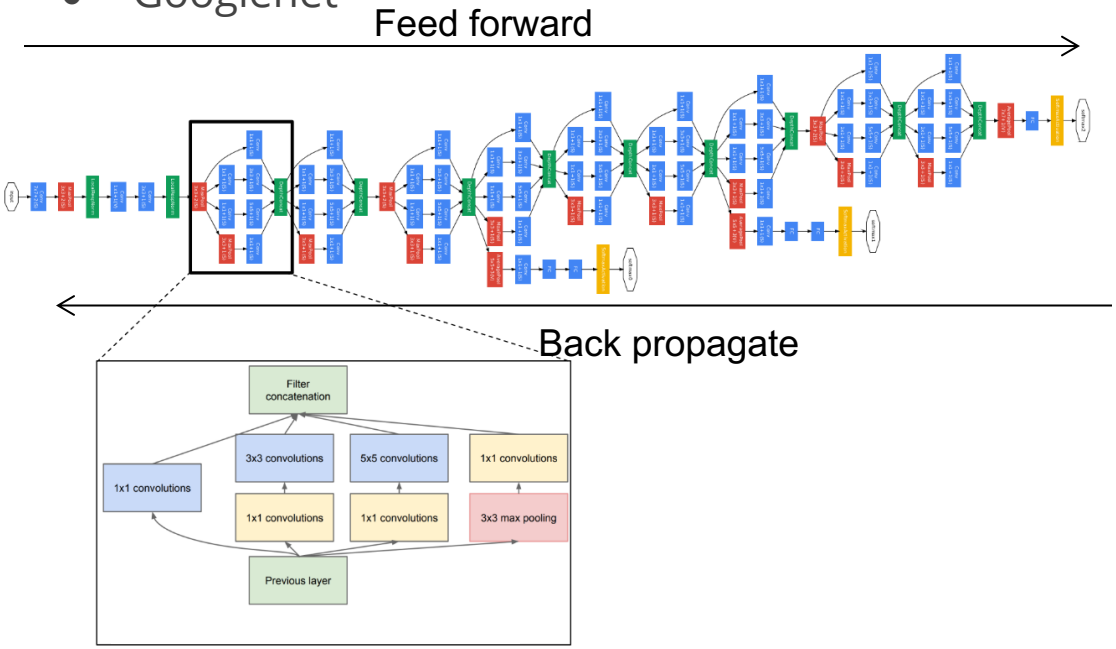
Παραλληλισμός σε επίπεδο μοντέλου

- **Model parallelism**
  - Κάθε μονάδα επεξεργασίας διατηρεί κομμάτι του μοντέλου και εκπαιδεύει τις αντίστοιχες παραμέτρους
  - Όλες οι μονάδες επεξεργασίας αναλαμβάνουν αντίγραφο του ίδιου mini-batch
  - Απαιτείται Alltoall επικοινωνία για την ανταλλαγή των παραμέτρων όταν υπάρχουν συνδέσεις μεταξύ νευρώνων



Παραλληλισμός pipeline

- Googlenet



- Δεν υπάρχει παραλληλισμός μεταξύ των επιπέδων στο forward και στο backward propagation
  - Κάθε επίπεδο τροφοδοτεί το επόμενο ή το προηγούμενο
- Depth D: το βάθος του δικτύου
  - Περιορίζει τον παραλληλισμό
  - GoogleNet: D=22

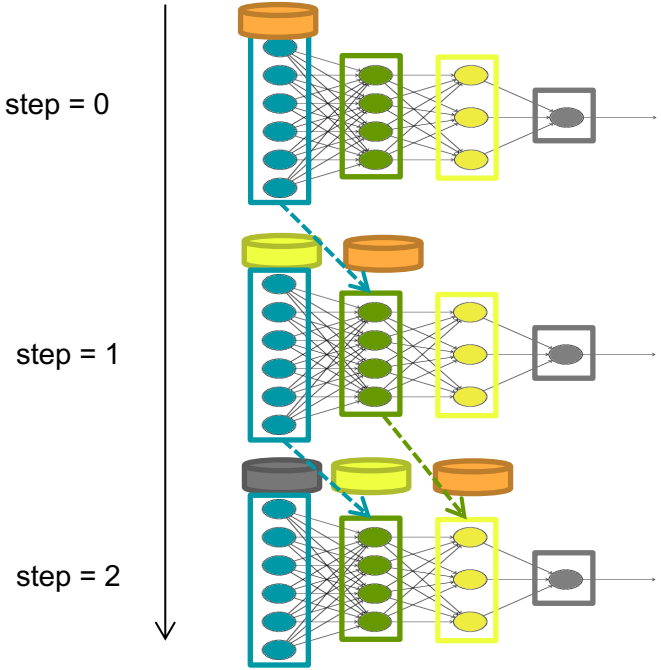
Παραλληλισμός pipeline

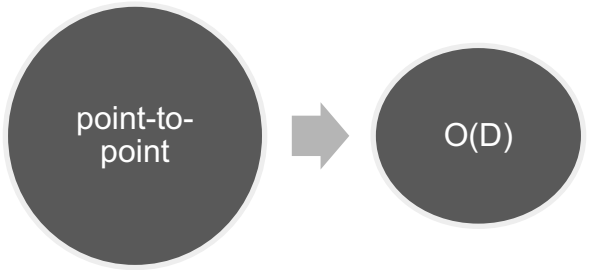
- **Pipeline parallelism**: παραλληλισμός σωλήνωσης
  - Κάθε μονάδα επεξεργασίας διατηρεί κάποιο επίπεδο του μοντέλου
  - Παράδειγμα forward propagation:
    - Βήμα t: Η μονάδα επεξεργασίας *l* εκπαιδεύει το επίπεδο *l* με το υποσύνολο δεδομένων S^i
    - Βήμα t: Τροφοδοτεί τα αποτελέσματά της στη μονάδα επεξεργασίας l+1
    - Βήμα t+1: Η μονάδα επεξεργασίας *l+1* εκπαιδεύει το επίπεδο *l+1* με το υποσύνολο δεδομένων S^i
    - Βήμα t+1: Η μονάδα επεξεργασίας *l* εκπαιδεύει το επίπεδο *l* με το υποσύνολο δεδομένων S^{i+1}

- *Πλεονεκτήματα*:
  - επιτρέπει την εκπαίδευση αν το μοντέλο είναι μεγάλο (δε χωράει στη μνήμη ενός κόμβου)
  - η επικοινωνία περιορίζεται στα όρια μεταξύ των επιπέδων
- *Μειονεκτήματα*:
  - τα υποσύνολα δεδομένων πρέπει να καταφθάνουν με συγκεκριμένο ρυθμό για να τροφοδοτείται το pipeline
  - απαιτείται αντιγραφή των υποσυνόλων των δεδομένων σε κάθε μονάδα επεξεργασίας

Παραλληλισμός pipeline

- **Pipeline parallelism**: παραλληλισμός σωλήνωσης
  - Κάθε μονάδα επεξεργασίας διατηρεί κάποιο επίπεδο του μοντέλου





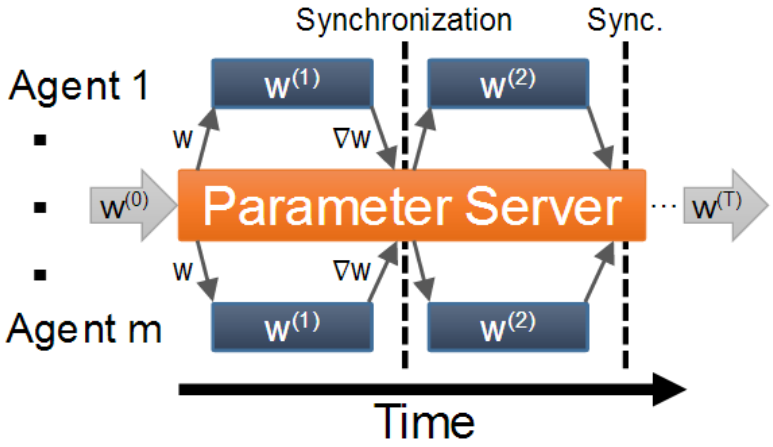
Τεχνικές μείωσης του χρόνου επικοινωνίας σε παράλληλα βαθιά νευρωνικά δίκτυα

Παραλληλισμός, επικοινωνία και συνέπεια

- Σε όλες τις περιπτώσεις παραλληλοποίησης των βαθιών νευρωνικών δικτύων (data-parallel, model-parallel, pipeline-parallel) απαιτείται:
  - Επικοινωνία για την επικαιροποίηση των παραμέτρων (W,b)
  - Συγχρονισμός για συνέπεια – όλες οι μονάδες επεξεργασίας πρέπει να έχουν την ίδια εικόνα του νευρωνικού δικτύου σε μια συγκεκριμένη φάση της εκπαίδευσης

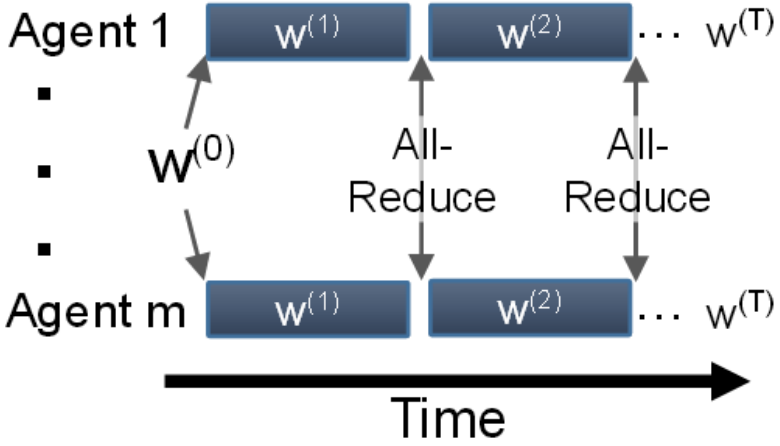
Παράλληλα, συνεπή μοντέλα

- Παραλληλισμός δεδομένων με parameter server



- Επικοινωνία + Συγχρονισμός σε κάθε βήμα μεταξύ των workers και του master

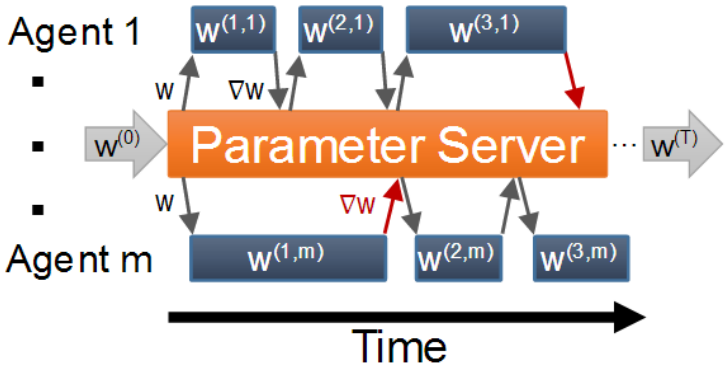
- Παραλληλισμός δεδομένων με decentralized parameter servers



- Επικοινωνία + Συγχρονισμός σε κάθε βήμα μεταξύ των workers

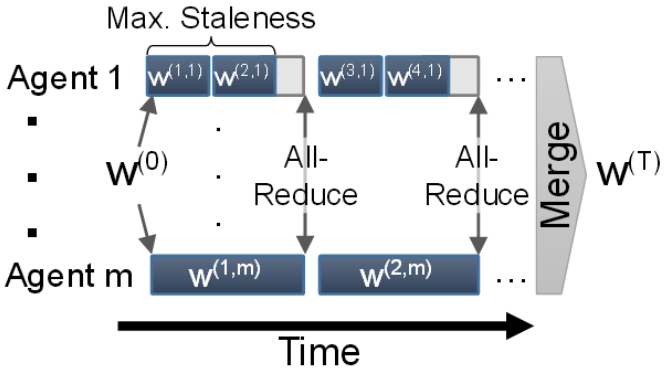
Παράλληλα, συνεπή μοντέλα με λιγότερο συγχρονισμό

- Παραλληλισμός δεδομένων με parameter server



- Συγχρονισμός μεταξύ των workers και του master μόνο όταν οι παράμετροι έχουν γίνει “stale” – δεν είναι αρκετά επίκαιρες

- Παραλληλισμός δεδομένων με decentralized parameter servers



- Συγχρονισμός μεταξύ των workers και μόνο όταν οι παράμετροι έχουν γίνει “stale” – δεν είναι αρκετά επίκαιρες

Parameter servers και κόστος επικοινωνίας

- Κεντρικός parameter server σε master node
  - + Έχει καθολική εικόνα της εκπαίδευσης
  - + Μπορεί να επιτρέψει ασύγχρονη λειτουργία των workers
  - + Μπορεί να μειώσει την επικοινωνία αν εκτελεί κάποιους υπολογισμούς αντί για τους workers
  - - Εκτελεί reduce + broadcast που δεν είναι βελτιστοποιημένη λειτουργία, σε αντίθεση με το allreduce
- Κατανεμημένοι parameter servers στους worker nodes
  - + Έκτελούν allreduce επικοινωνία που είναι βελτιστοποιημένη για τα περισσότερα δίκτυα
  - - Σε κάθε βήμα απαιτείται επικοινωνία για να υπάρχει καθολική εικόνα

Μέγεθος παραμέτρων και κόστος επικοινωνίας

- Ανεξάρτητα του σχήματος της επικοινωνίας (allreduce, alltoall, reduce+broadcast), το κόστος της επικοινωνίας εξαρτάται από το μέγεθος των μηνυμάτων
  - Μέγεθος μηνυμάτων: πλήθος και μέγεθος παραμέτρων που πρέπει να ανταλλαχθούν

- Η **συμπίεση** των μηνυμάτων βελτιώνει το κόστος της επικοινωνίας και άρα την ταχύτητα εκπαίδευσης
  - *Quantization (κβάντιση):* μείωση του αποθηκευτικού χώρου για κάθε παράμετρο με αντίστοιχη μείωση της ακρίβειας (κβάντιση σε μεγαλύτερο δεκαδικό ψηφίο)
    - Μείωση μεγέθους κάθε στοιχείου ενός μηνύματος
  - *Sparsification (αραίωση)*: κάποιες παράμετροι δεν ανταλλάσσονται αν η μεταβολή στην τιμή τους δεν είναι σημαντική (δεν ξεπερνά κάποιο κατώφλι)
    - Μείωση πλήθους των στοιχείων ενός μηνύματος

Βιβλιογραφία

- Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis, Tal Ben-Nun, Torsten Hoefler, ETH Zurich
  - https://arxiv.org/pdf/1802.09941.pdf
- Distributed Machine Learning: A Brief Overview, Dan Alistrah, IST Austria
  - PODC2018 tutorial - https://www.podc.org/data/podc2018/podc2018-tutorial-alistarh.pdf