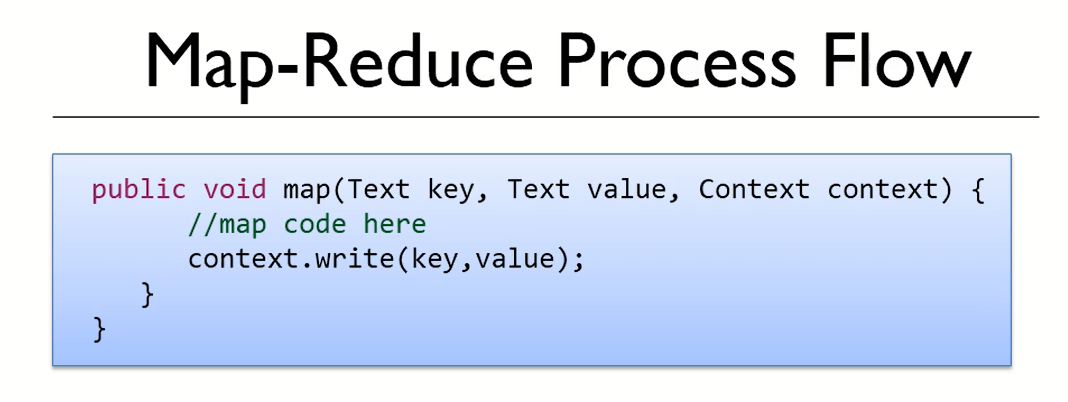
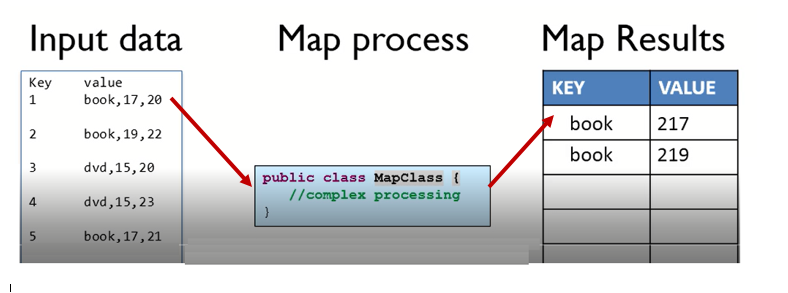
**Map-Reduse Shufle Step**

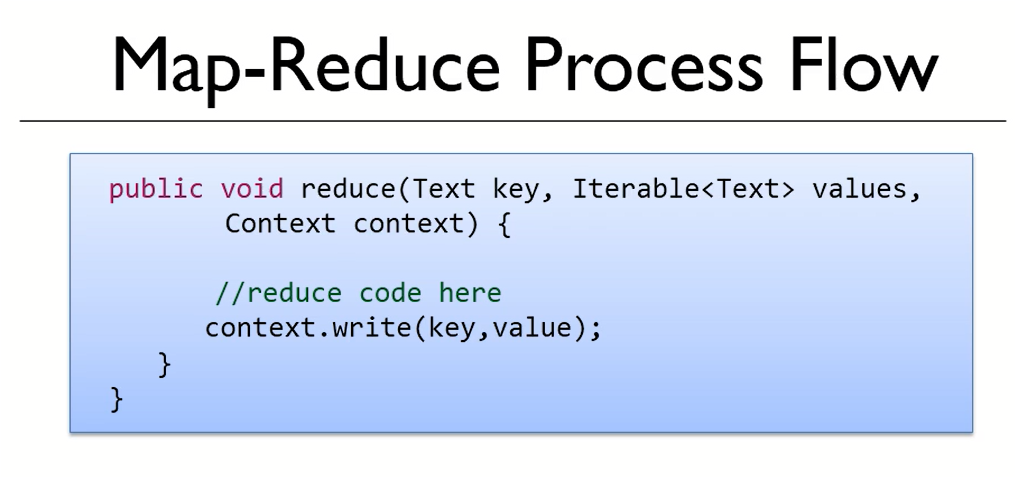


map მეთოდის შემავალი პარამეტრებია (Objekt key, Text value) და ამ მეთოდის შესრულების შედეგს, რომელიც წარმოდგენილია ასევე (key, value) წყვილით ვწერთ კონტექსტში Context.

ჩვენ შემთხვევაში როდესაც ვამუშავებდით ფაილის მონაცემებს map მეთოდის შემავალი პარამეტრებიდან key პარამეტრის მნიშვნელობა ავტომატურად განისაზღვრებოდა, როგორც ფაილში სტრიქონის ნომერი ხოლო value-ს როლში თავად სტრიქონი.



map მეთოდი ეშვება ყოველი შემავალი (key, value) წყვილისათვის და მის მუშაობის შედეგს წერს ასევე (key, value) ფორმატში

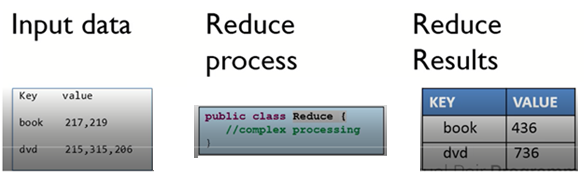


განსხვავებით map მეთოდისაგან reduce მეთოდის პარამეტრებს წარმოადგენენ

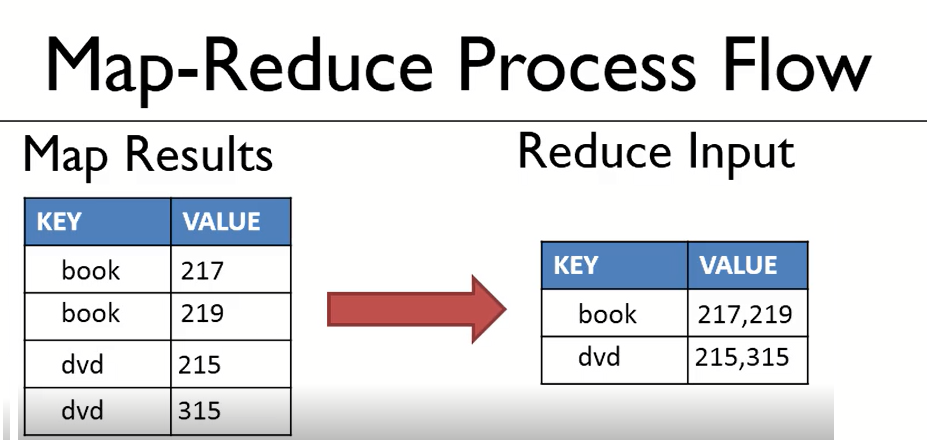


აღსანიშნავია რომ reduce მეტოდი ეშვება ერთხელ თითოეული Key-სათვის

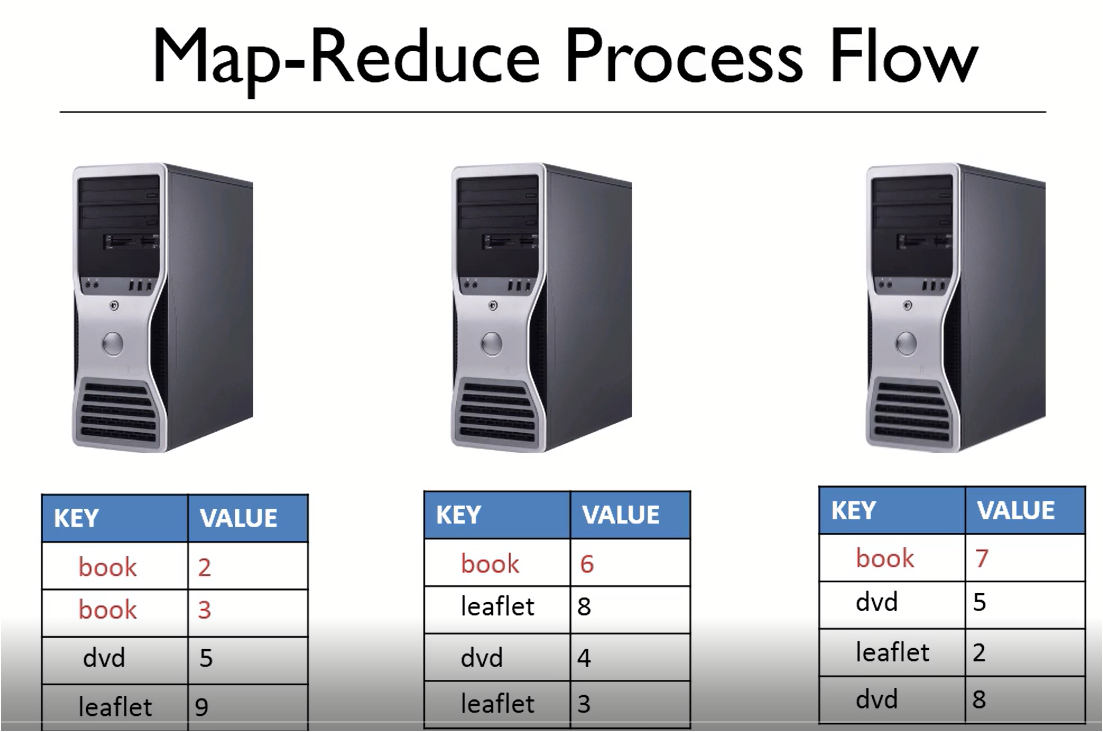
Reduce მეთოდის გაშვებამდე Hadoop აჯგუფებს ყველა განსხვაებულ Key-ს და იმ რაოდენობებს



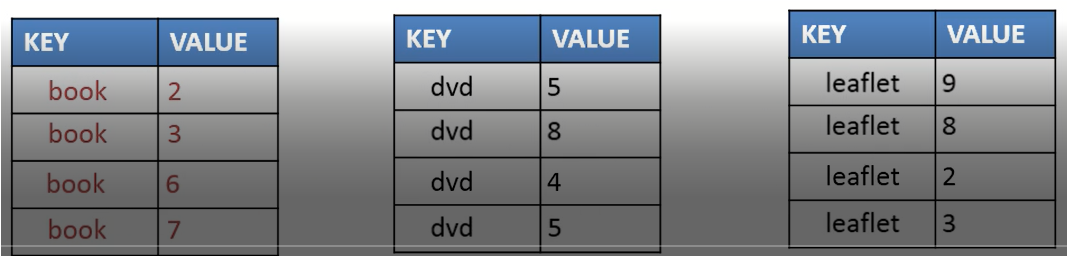
როგორც ვხედავთ map მეთოდის დასრულების შემდგომ ჩვენ გვაქვს ერთი დამატებითი საფეხური:



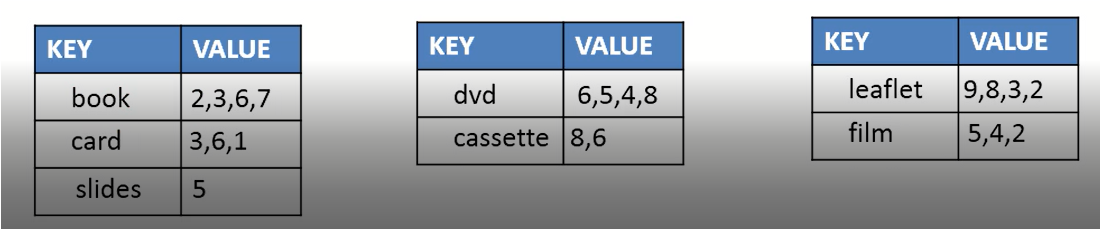
**როდესაც ჩვენ საქმე გვაქვს Hadoob-ის განაწილებულ კლასტერთან ასეთ შემთხვევაში თითოეულ Node-ზე map მეთოდის დასრულების შემდეგ გვექნება map Output**



მანამდე სანამ reduce მეთოდი გაეშვება საჭიროა რომ თითოეული გასაღების key-სათვის მნიშვნელობები დაჯგუფდეს თითოეულ Node-ზე:



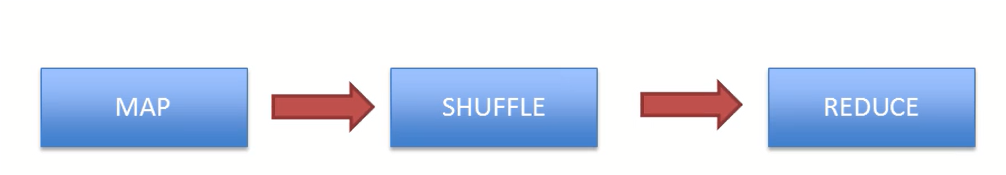
შემდგომ კი გარდაიქმნას შემდეგი სახით:



აღნიშნულ ოპერაციებს ახორციელებს Shufle პროცესი (საფეხური)

ამის შემდგომ კი შესაძლებელია reduce მეთოდის გაშვება თითოეულ ნოდზე

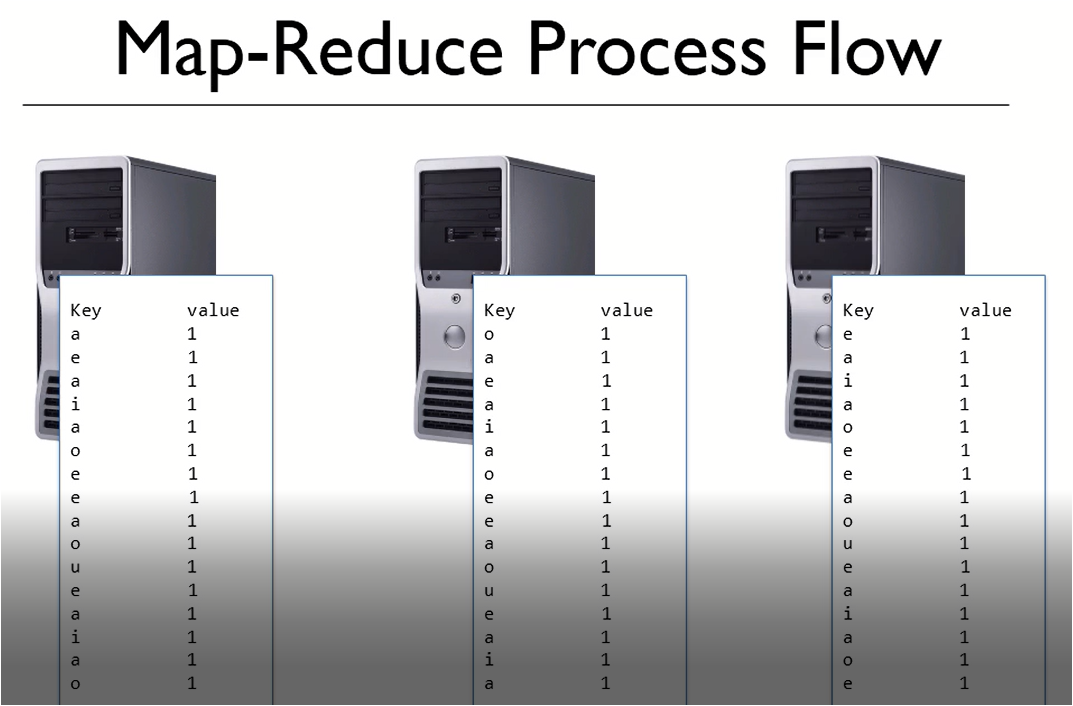
ამრიგად სინამდვილესი ჩვენ გვაქვს MapReduce პროცესის შემდეგი საფეხურები (ეტაპები):



SHUFFLE პროცესი არ იწყება მანამდე სანამ არ დასრულდება MAP პროცესი ხოლო

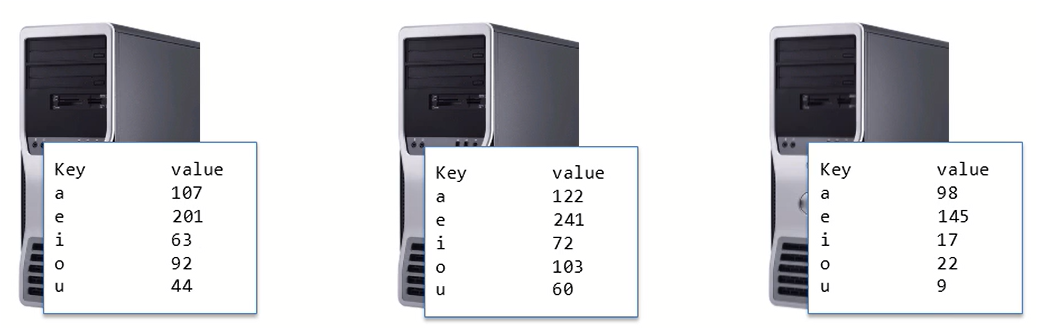
REDUCE პროცესი არ იწყება მანამდე სანამ არ დასრულდება SHUFFLE პროცესი.

დავუბრუნდეთ ხმოვნების რაოდენობების გამოთვლის ამოცანას. MAP პროცესის დასრულების შემდგომ ჩვენ გვექნება შემდეგი სურათი:

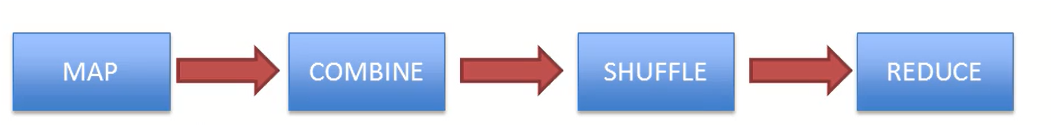


თუ ამ პროცესის შემდგომ ჩვენ დავიწყებდით მიღებული key, value წყვილების გადაჯგუფებას Hadoop-ის კლასტერის ხვადასხვა node-ებზე, ეს პროცესი გამოიწვევს ქსელში დიდი მონაცემების გადაცემას და შესაბამისად ქსელის დატვირთვას.

იმისათვის რომ თავიდან ავიცილოთ ქსელის გადატვირთვა საჭიროა SHUFFLE პროცესის დაწყებამდე მოვახდინოთ შემდეგი სახის დაჯგუფება:

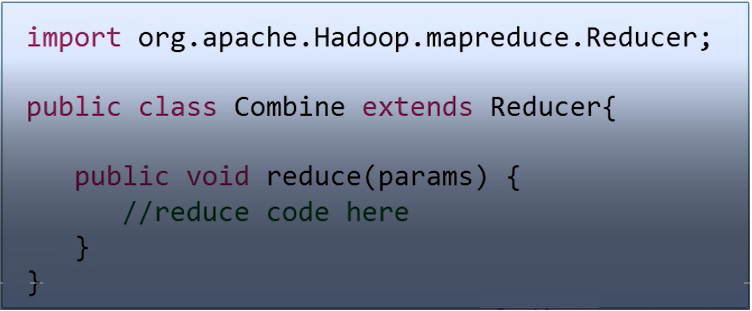


აღნიშნულს ეწოდება COMBINE- პროცესი. ამრიგად ჩვენ მივედით MapReduce პროცესის შემდეგ საფეხურებამდე:

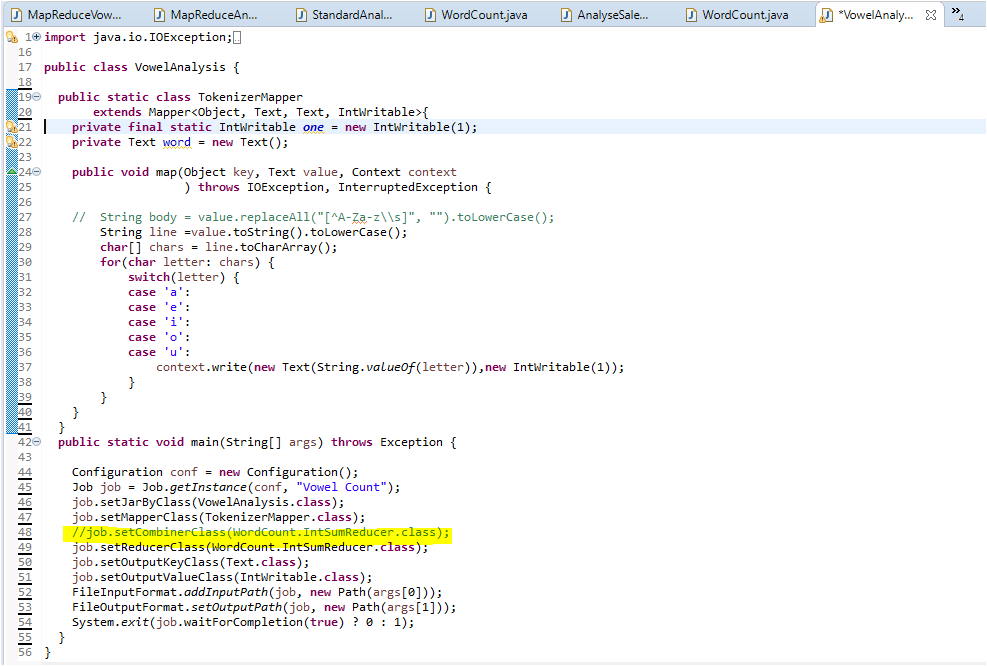


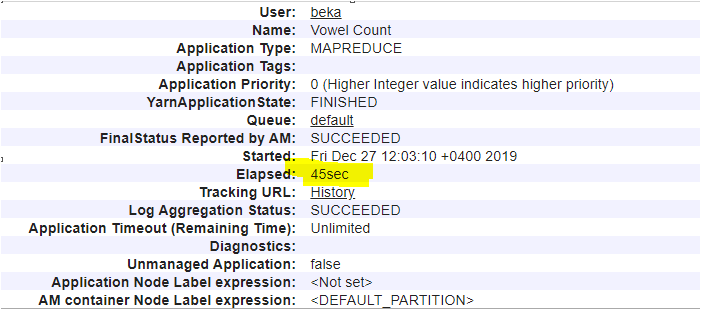
აღსანიშნავია ამასთან რომ COMBINE-საფეხურის სინტაქსი იგივეა რაც REDUCE საფეხურის სინტაქსი. მართლაც თითოეული მათთაგანი (როგორც REDUCE ასევე COMBINE) იღებს გასაღების (key-ის) მნიშვნელობას და აჯამებს შესაბამის value- მნიშვნელობებს.

ამრიგად ჩვენ შეგვიძლია გამოვიყენოთ ერთიდაიგივე პროგრამული კოდი როგორც REDUCE ასევე COMBINE საფეხურებისათვის

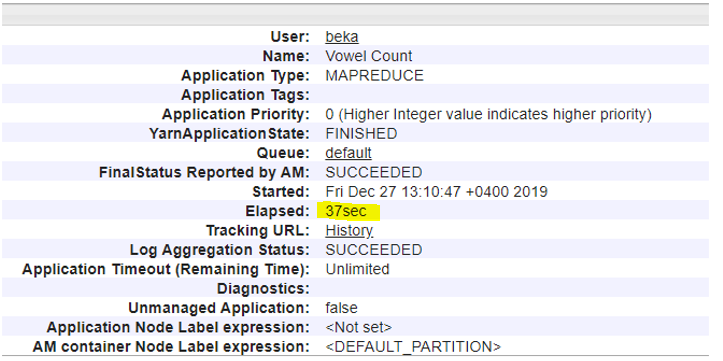


MapReduce without Combiner Step

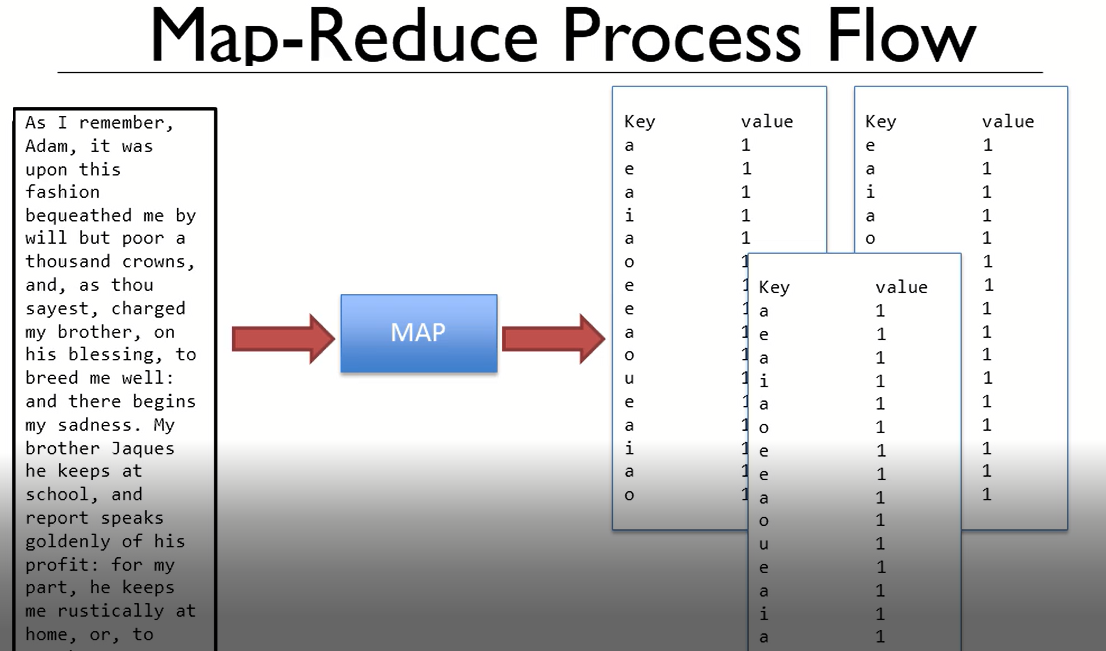


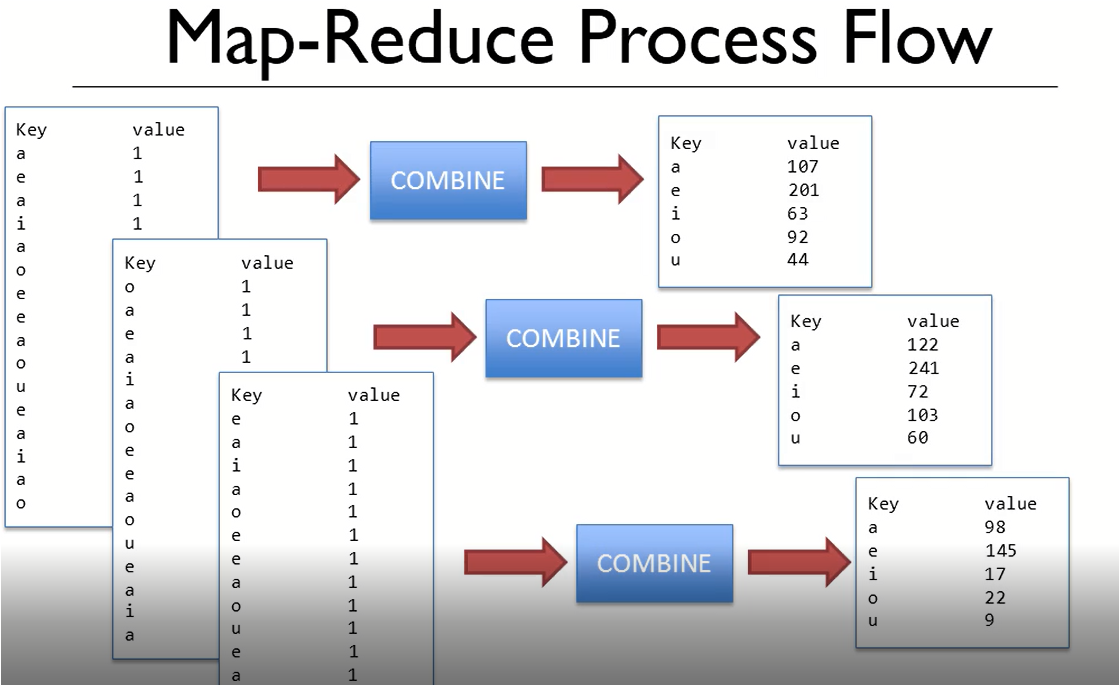


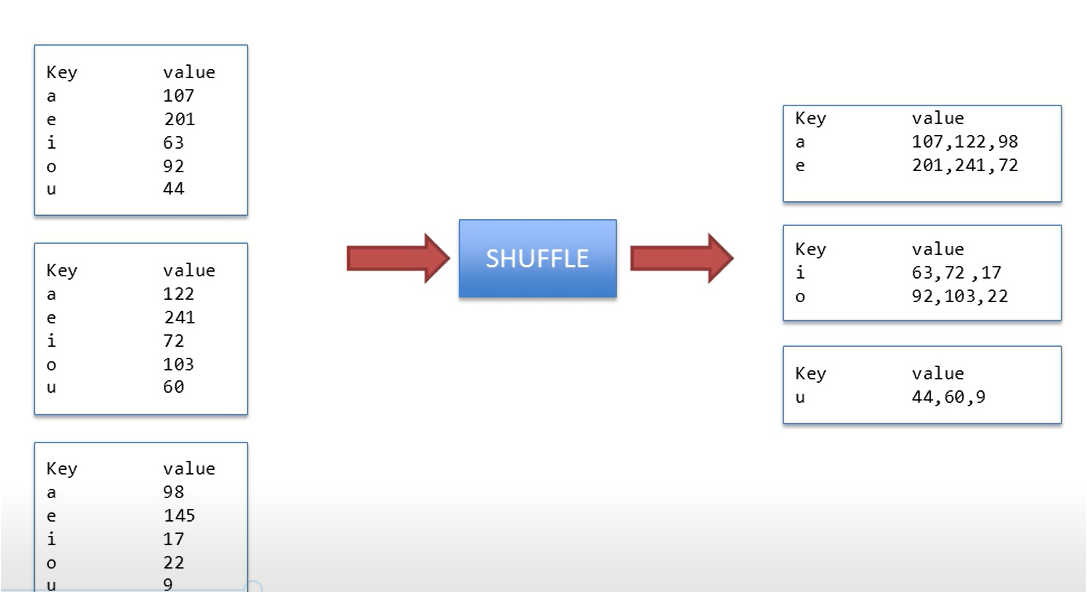
**MapReduce with Combiner Step**

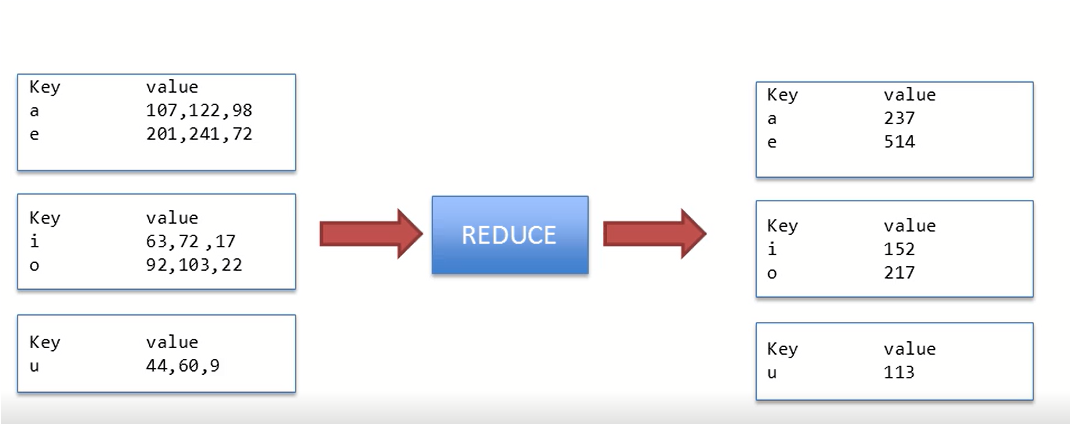


როგორც ვხედავთ COMBINE-საფეხურის გამოყენების შემთხვევაში პროგრამის მუშაობის დრო შემცირდა 30%-ით.

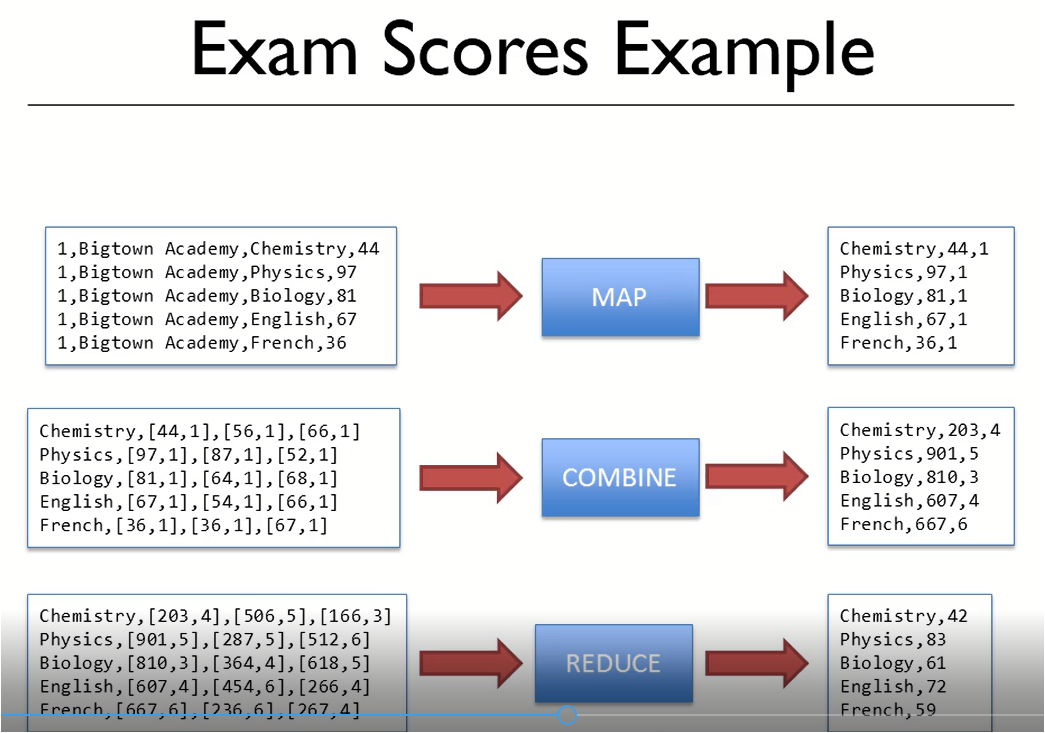
****







Calculate average Scores



ამ ამოცანაში ჩვენ გამოსათვლელი გვაქვს საშუალო ქულა თითოეულ საგანში. ზემოთ მოყვანილია MAP, COMBINE, REDUCE პროცესების მიერ შესასრულებელი ამოცანები.

ამ შემთხვევაში values[3] წარმოადგენს ჩვენ შემთხვევაში key-ს ხოლო value-ს როლში ავიღოთ მიღებული ქულა და 1-იანი რომელიც მიუთითებს იმ სტუდენტების რაოდენობაზე რამდენმაც მიიღო ეს ქულა. ჩვენ გვექნება მაგ. Chemistry,44,1. იმ შემთხვევაში თუ აღმოჩნდებოდა რომ კიდევ გვყავს ისეთი სტუდენტი რომელმაც ქიმიაში მიიღო 44 ქულა მაშინ კიდევ ერთი ასეთი ჩანაწერი გვექნებოდა

