1. Describe architecture for the solution which sends emails and:

- **May run in parallel on distinct machines**
- **As the input receives email addresses**
- **Same email address may be provided multiple times**
- **Email to the same address shouldn't be send often than once per 12 hours**

The solution would need a database that will store emails with timestamps (preferably sql as indexing can help getting the data quickly). I would consider using master slave architecture, where the server is receiving list of adresses along with the email contents or instructions what email should be send (If its only 1 type of email I dont see any point in running it on separate machines honestly).
Master solution would gather all email requests, check if database has timestamp for provided emails, if no then email adress with timestamp (now) is added, otherwise the email is scheduled for existing timestamp (+12h - this could be a parameter), and adding new entry to db with timestamp = previous timestamp+12h.
To make searching the db quicker, old timestamps (older than 12h) can be moved to the audit table, where data about what emails were sent where exactly can be stored, but at the same time the table that is often used by the server should be kept as small as possible.

2. We have list of users purchases and its details. We would like to find users with similar purchases. What solution would you recommend?

In my opinion the best approach would be to use k-means clustering method, that can easily group customers. However, grouping against single products would not be optimal, thus earlier it would be best to assert the purchasing patterns, by using for example baske analysis. Knowing common basket patterns, the k-means clustering can more precisely group customers not to single products, but to the popular groups of products. However It would be best to try both approaches as it still depends on the data.