**What are the differences between parallel, concurrent and asynchronous programming?**

10 Answers

[Jan Christian Meyer](https://www.quora.com/profile/Jan-Christian-Meyer)

[**Jan Christian Meyer**](https://www.quora.com/profile/Jan-Christian-Meyer), Ph.D. in Computer Science

97.6k Views • Upvoted by [Adam Hartline](https://www.quora.com/profile/Adam-Hartline), Programming since age 8. Corporate programming since age 25.

[Jan](https://www.quora.com/profile/Jan-Christian-Meyer) is a Most Viewed Writer in [Concurrency (computer science)](https://www.quora.com/topic/Concurrency-computer-science).

These people are working in ***parallel***:



Parallel programs distribute their tasks to multiple processors, that actively work on all of them simultaneously.  
  
This guy is ***concurrently*** juggling 8 balls:



Concurrent programs handle tasks that are all in progress at the same time, but it is only necessary to work briefly and separately on each task, so the work can be interleaved in whatever order the tasks require.  
  
This guy is ***asynchronously*** doing his laundry while reading:



An asynchronous program dispatches tasks to devices that can take care of themselves, leaving the program free do something else until it receives a signal that the results are finished.

[Written Sep 8, 2015](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming/answer/Jan-Christian-Meyer) • [View Upvotes](https://www.quora.com/api/mobile_expanded_voter_list?type=answer&key=aZmHgYYeAWZ)

[Upvote**3.9k**Downvote](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Comments**35+**](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Share](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Mike MacHenry](https://www.quora.com/profile/Mike-MacHenry)

[**Mike MacHenry**](https://www.quora.com/profile/Mike-MacHenry), software engineer, improv comedian, maker

5.2k Views

A concurrent program is one with multiple control flows that overlap in time. Concurrency changes the semantics of a program such that the independent flows of control can observe this change. For example, a web server is a concurrent program and I can observe, as a client of that web server, that I don't need for one user's request to complete before mine can begin.  
  
A parallel program is one where independent subexpressions are evaluated at the same time on different processors (potentially not even on the same machine) in order to finish a computation faster. Though this does in practice require the programmer to annotate her program to show which parts should be run in parallel this is not required in theory. Furthermore parallel computation presents not semantic change to the program.  
  
It is important to note that concurrent programs might be run on just one processor using an interleaving scheduler and they are still just as concurrent as ones that are run on multiple processors -- though they are potentially slower.  This is an aspect that[**Bernhard Støcker**](https://www.quora.com/profile/Bernhard-St%C3%B8cker) has gotten wrong in his answer. Parallel computation is **not** a special case of concurrency. One can achieve parallelism using concurrency constructs which often leads to confusion.  
  
Asynchronous evaluation is when one thread of control calls another function or method on another thread of control and instead of waiting for a return value, it carries on with its work. In many asynchronous models, that other thread, or actor, is likely to make an asynchronous call back later in order to deliver the answer. This model of computation works really well for concurrent programming due to the easy with which one can load balance work since a slight delay in one thread will not cause cascading delays throughout.

[Updated Sep 13, 2015](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming/answer/Mike-MacHenry) • [View Upvotes](https://www.quora.com/api/mobile_expanded_voter_list?type=answer&key=VFsitjqh7s4)

[Upvote**19**Downvote](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Comments**2**](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Share](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Bernhard Støcker](https://www.quora.com/profile/Bernhard-St%C3%B8cker)

[**Bernhard Støcker**](https://www.quora.com/profile/Bernhard-St%C3%B8cker), I wrote code in 8 languages. My favorites are the functional once.

2.2k Views

Concurrent Programming from Wikipedia: **Concurrent computing** is a form of computing in which several computations are executing during overlapping time periods – concurrently – instead of *sequentially* (one completing before the next starts).  
  
This means several processes are instanciated to do a job at the same time.   
  
Parallel is a special case of [Page on it.it](http://it.it/)  means you split a task in subtasks and compute them in parallel while the main process is waiting until ***all*** processes are resolved. The last point is not nessessary when having  concurrent processes.   
  
Asynchronous as a term is used when things are done outside your app, while the app proceeds instead of waiting for the reply. It is somehow a different concept.  
  
Maybe take a look into this stackoverflow answer: [What is the difference between concurrency, parallelism and asynchronous methods?](http://stackoverflow.com/questions/4844637/what-is-the-difference-between-concurrency-parallelism-and-asynchronous-methods)

[Written Feb 24, 2015](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming/answer/Bernhard-St%C3%B8cker)

[UpvoteDownvote](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Comment](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Share](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Bruno D. Rodrigues](https://www.quora.com/profile/Bruno-D-Rodrigues)

[**Bruno D. Rodrigues**](https://www.quora.com/profile/Bruno-D-Rodrigues), Coding since the time that we'd enter hexadecimal values taken from magazines

3k Views • Upvoted by [Miguel Paraz](https://www.quora.com/profile/Miguel-Paraz), Programming since 1985 at age 11.

Originally Answered: [What is the difference between concurrency, parallel programming, and asynchronous programming?](https://www.quora.com/What-is-the-difference-between-concurrency-parallel-programming-and-asynchronous-programming?no_redirect=1)

Parallel programming: 10 persons doing independent tasks  
  
Concurrency: 10 persons going to the boss asking for a new task or reporting the status of the finished one, with the boss having to coordinate who talks when. 10 persons at the coffee machines.   
  
Async: one person does a task, but sub tasks can be delegated mainly because they are external. So instead of leaving the office to go drop a mail into the post office, and sitting there waiting for the reply snail mail to arrive, the person asks a dedicated courier to go drop the mail, takes a note to not forget to follow up, and goes do something else. Later when the postman brings the reply post, delivers it into the secretary, that puts the mail into the todo box. Depending on the mail priority, it can even be put on the top of the queue, or at the end. The person eventually picks that task, recalls the previous note, and carries on doing something with that result.

[Written Aug 29, 2015](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming/answer/Bruno-D-Rodrigues) • [View Upvotes](https://www.quora.com/api/mobile_expanded_voter_list?type=answer&key=3u0OusmakBx)

[Upvote**28**Downvote](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Comments**1+**](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Share](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Christian Baune](https://www.quora.com/profile/Christian-Baune)

[**Christian Baune**](https://www.quora.com/profile/Christian-Baune), Versatile expert programmer with a thing for Maths !

1.7k Views

Let say you want to make the sum of the first 30 terms of :  
x2+\floor(x3)x2+\floor(x3)  
  
**In a parallel fashion :**

1. **in** parallel (on nodes):
2. r1=calculateSumOfRange(1 to 10, eqn );
3. r2=calculateSumOfRange(11 to 20, eqn );
4. r3=calculateSumOfRange(21 to 30, eqn );
5. result = r1 + r2 + r3;

The tasks are submitted to nodes (that can be threads or physical devices) by partitions. So each node does a part of the work in parallel.  
Now we have multiple CPU cores thus we can achieve real parallelism.  
  
**In a concurrent fashion without promises:**

1. **for**(i **from** 1 to 30):
2. adder.submit(i,eqn);
3. result = adder.getResult();

The heavy work (computing the formula) is deported to an external thread that will usually be a consumer feed by a consumer.  
adder.getResult() is blocking.  
  
  
**In asynchronous :**

1. calculationService.calculate(1 to 30,eqn).**then**( r -> result = r);

The heavy work is deported to an external thread and we provide a way to be notified.[Futures and promises](https://en.wikipedia.org/wiki/Futures_and_promises)  
  
**Examples :**  
When you do a search on Google, you are using a service using parallel programming. They query a lot of nodes then merge the results.  
  
When you use a website featurin AJAX, you use asynchronous programming. Requests are done in background and the JavaScript program get notified when resources are ready.  
  
When you open a file, usually the "open file" dialog is an OS call which block the calling program until you answered. This is simple concurrency.  
  
I introduced a slight bias : concurrency means blocking. No it does not. Concurrency is making part of a program appear to work in parallel. Parallelism is really doing it. (multiple cores)  
  
One goes further by requiring that in parallelism you solve problem by partitioning them. (like the sum example I gave)   
Grid computing is a really good example of parallelism.  
  
See answer of [Mike MacHenry](https://www.quora.com/profile/Mike-MacHenry) for the details.

[Written Aug 29, 2015](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming/answer/Christian-Baune) • [View Upvotes](https://www.quora.com/api/mobile_expanded_voter_list?type=answer&key=q0V9pGV6s9Y)

[Upvote**1**Downvote](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Comment](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Share](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Kevin Cameron](https://www.quora.com/profile/Kevin-Cameron)

[**Kevin Cameron**](https://www.quora.com/profile/Kevin-Cameron), C since the 80s, parallel processing (Inmos, CSP), SMP in the 90s, C++ since.

908 Views

The concept of "synchronous" is more of a machine/hardware level thing where things need to be done at certain times, and turns up in software as the difference between polling and reactive programming - i.e. waking up on a timer to check something, or waking upon an interrupt to service things.  
  
The overlap is in that preemptive concurrent multithreading is usually handled by the same hardware/interrupt mechanisms, i.e. a timer will make the OS suspend a thread (by interrupt) and switch to another without any synchronization of the threads, whereas if one thread wakes up another and stops itself the swap would be considered synchronized (not preemptive).   
  
In the preemptive case the threads are concurrent in that the OS is trying to run more than one task at once (logically, by interleaving them). If the tasks are parallel they would also be concurrent but actually executing at the same time.

[Written Oct 31, 2015](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming/answer/Kevin-Cameron) • [View Upvotes](https://www.quora.com/api/mobile_expanded_voter_list?type=answer&key=LWnkL82MEPZ)

[Upvote**1**Downvote](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Comment](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Share](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[James Burgess](https://www.quora.com/profile/James-Burgess-26)

[**James Burgess**](https://www.quora.com/profile/James-Burgess-26), James Burgess is a technology writer and software developer at @ Valuecoders.com

525 Views

**Parallel and concurrent** are effectively the same principle as you correctly surmise, both are related to tasks being executes simultaneously although I would say that parallel tasks should be truly multitasking, executed "at the same time" whereas concurrent could mean that the tasks are sharing the execution thread while still appearing to be executing in parallel.  
  
**Asynchronous** methods aren't directly related to the previous two concepts, asynchrony is used to present the impression of concurrent or parallel tasking but effectively an asynchronous method call is normally used for a process that needs to do work away from the current application and we don't want to wait and block our application awaiting the response.  
  
Example, getting data from a database could take time but we don't want to block our UI waiting for the data. The asynch call takes a call-back reference and returns execution back to your code as soon as the request has been placed with the remote system. Your UI can continue to respond to the user while the remote system does whatever processing is required, once it returns the data to your call-back method then that method can update the UI (or hand off that update) as appropriate.  
  
James  
Software developer at [www.valuecoders.com](http://www.valuecoders.com/)

[Written Sep 15, 2015](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming/answer/James-Burgess-26) • [View Upvotes](https://www.quora.com/api/mobile_expanded_voter_list?type=answer&key=doa3pPfW0KD)

[Upvote**4**Downvote](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Comment](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Share](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Chekkal Samir](https://www.quora.com/profile/Chekkal-Samir)

[**Chekkal Samir**](https://www.quora.com/profile/Chekkal-Samir), A computer programmer enchanted by symbiose between big data and statistics

594 Views

- Parallellism is like shopping in a Mall, every person is moving and selecting products (independent of others).  
- Concurrency occur when those person want to pay, then the cashier become the shared resource over which they are competing.  
Asynchronous try to enhance the wait time over a shared resource in a concurrent scenario. For example in a fictional store we could imagine that the cashier will just take your adresse and immediatly dispach your command for another team to package the order and send it afterward to your adresse where you will pay the total; and the cashier will take the next command more quickly than normal.

[Written Sep 9, 2015](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming/answer/Chekkal-Samir) • [View Upvotes](https://www.quora.com/api/mobile_expanded_voter_list?type=answer&key=mHx8q7JjOJj)

[Upvote**1**Downvote](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Comment](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Share](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Yutong Zhao](https://www.quora.com/profile/Yutong-Zhao-1)

[**Yutong Zhao**](https://www.quora.com/profile/Yutong-Zhao-1), Schrödinger

819 Views

One of the best things I've ever read is Rob Pike's post on the nuances between parallelism and concurrency:  
  
[Concurrency is not parallelism](http://blog.golang.org/concurrency-is-not-parallelism)  
  
The key distinction is that *concurrency* is a pattern, where as *parallelism* is a mode of execution.  
  
Suppose I'm doing database query, my server is in Hong Kong and my DB is in Africa. While I'm fetching data from the DB, my server can work on some other problem (eg. write out some log files).  
  
Parallelism on the other hand, is the mode of execution. I can achieve the above task by having one thread write out log files, and another thread execute the database query. Alternatively, I can just use one thread entirely, and have it context-switch from the DB query to the log writing.  
  
I generally lump asynchronicity and concurrency in the same bin.

[Written Sep 9, 2015](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming/answer/Yutong-Zhao-1) • [View Upvotes](https://www.quora.com/api/mobile_expanded_voter_list?type=answer&key=X0CaHiSHIia)

[Upvote**2**Downvote](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Comment](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Share](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming)

[Keith Allpress](https://www.quora.com/profile/Keith-Allpress)

[**Keith Allpress**](https://www.quora.com/profile/Keith-Allpress)

437 Views

Parallel means that the processing to acheive a single goal is divided up ***physically*** on the hardware. It is defined in contrast to sequential processing, where the locus of processing can be always identified as being located in space  and time.  
  
We communicate via sequential language, one person speaks (the where) and one word at a time (the when).  However the brain is parallel hardware, we cannot (easily) say exactly when and where the meaning originates.  
  
Parallel processing hardware does not necessarily mean the problem itself can be decomposed into parallel logic at some logical scale, but certainly problem solving or acheiving a logical goal is greatly assisted if the problem itself is decomposable this way.    
  
Some degree of logical parallelism becomes acheivable at the level of machine code even for sequential processing at larger scales, so the modern CPU has parallel pipelines. As scale increases, caching and scheduling algorithms introduce parallel elements, but to fully acheive it requires a special hardware paradigm.  
  
Concurrent and synchronous have straightforward meanings.   
  
Concurrent means con+current = "*together*+*at the same time*".   
Together here means the items are ***logically*** bound together in some manner. They just also happen to occur at the same time. Concurrency is purely an issue of logical decomposition into sub problems. Loops provide a good illustration of this. The use of uncounted loops in sequential  computing is seldom justified by the  logical structure of a problem. The sequential programmer simply uses loops as a crude tool to acheive what really is a parallelic subtask. The program is a network, not a tree, but the sequential loop is chosen to implement her subnet of parallel logic, because that is often all she has. The key concept is  that there are nodes where the processing must come together again in such a way that logical integrity is maintained. So concurrent processing is not merely putting  processing tasks on different processors at the same time, it is the management of the logic entailed when these results ***intersect*** with eachother.   
  
Asynchronous has an ***engineering*** origin, it means a+syn+chronous = "*not+together+by the clock*". Time has the sense here of timing,  as in soldiers marching.  Notice that now there is nothing at the logical level, it is purely a timing issue. To an engineer, synchronous means timing, and asynchronous means a communication is not timed to a clock signal. Timing gives predictability, whereas asynchronous communications gives flexibility in choosing when a communication can happen, and therefore where control can reside.

[Written Nov 11](https://www.quora.com/What-are-the-differences-between-parallel-concurrent-and-asynchronous-programming/answer/Keith-Allpress)