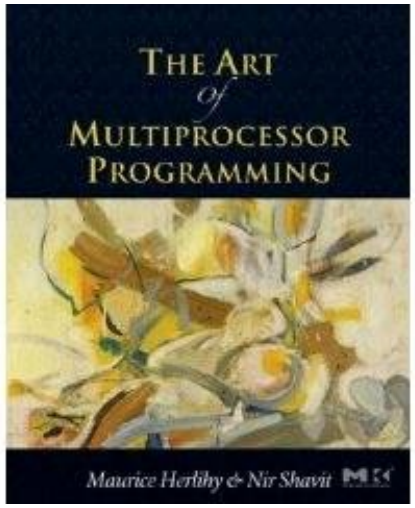
Introduction

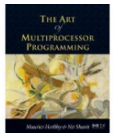


The Art of Multiprocessor Programming

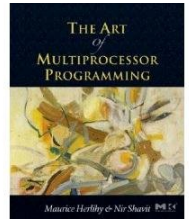
by Maurice Herlihy & Nir Shavit

Course Information

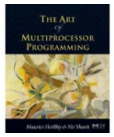
- Instructor :
  - **Prof. Hyungsoo Jung (정형수),**
  - **hyungsoo.jung@gmail.com**
- TA :
  - Please look behind you !!!
  - 오규석
  - 김영진



Course Information

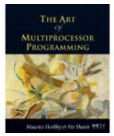


- Textbook :
  - **The Art of Multiprocessor Programming**
- Grading :
  - Only one paper exam (30% ~ 40%)
  - A series of programming projects (**VERY IMPORTANT**) (60% ~ 70%)
- Late and Missing Work
  - We do **NOT** strictly accept any late projects, but give partial credits for unfinished work, if it makes sense.

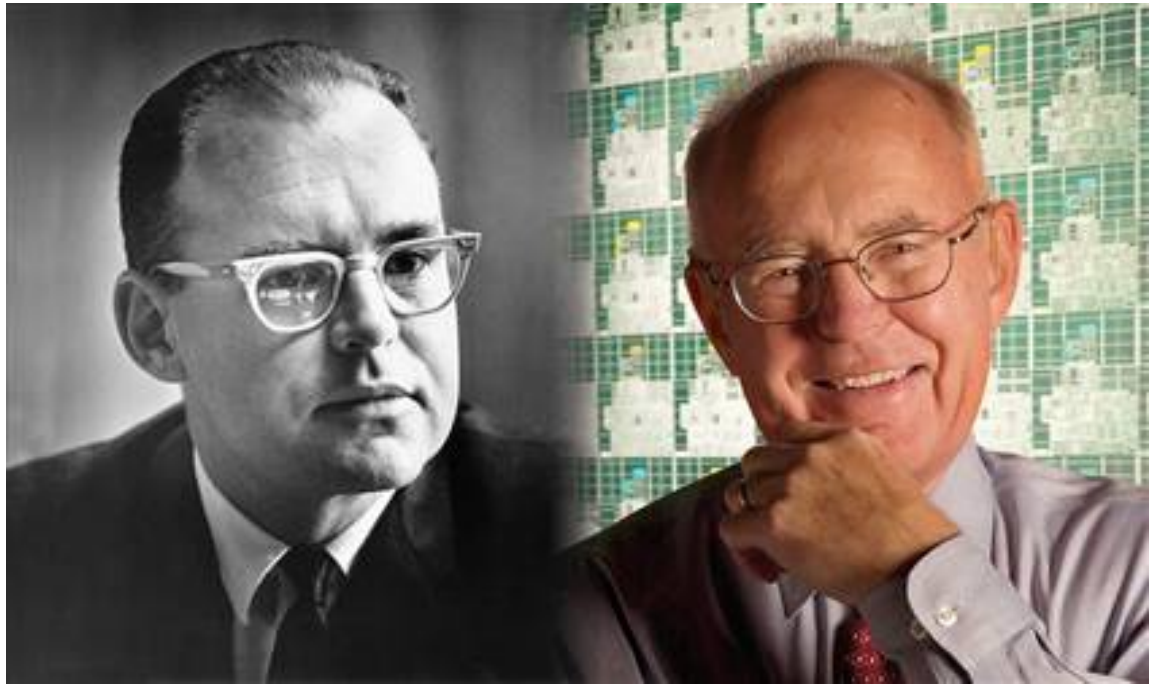


Hardware Development Trend

- Processor cores
- Network bandwidth
- Storage medium (HDD & SSD)

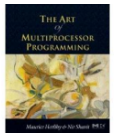


Moore's Law (processor cores)

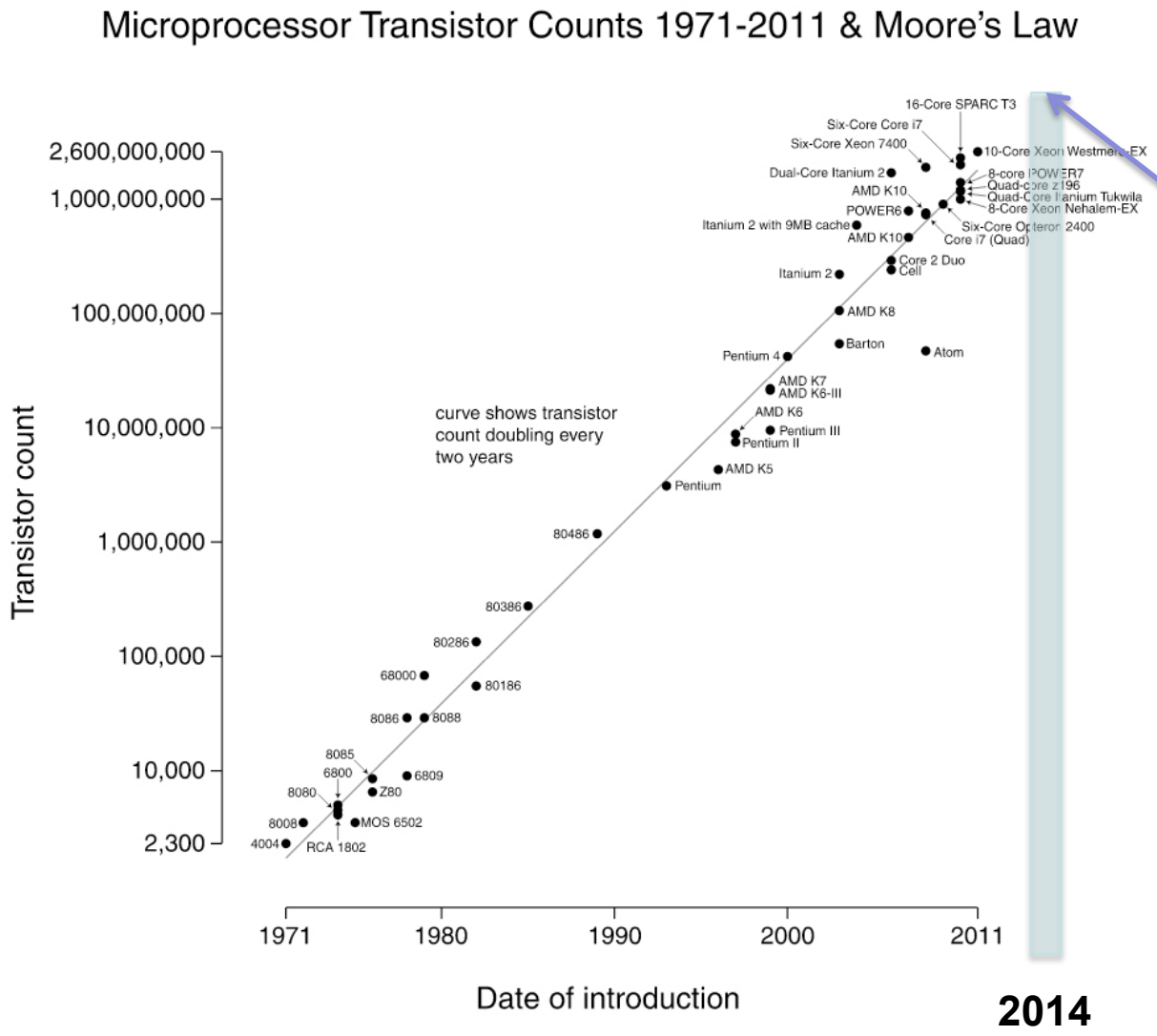


*"The number of transistors incorporated in a chip will approximately **double** every 24 months."*

--Gordon Moore, Intel co-founder



Moore's Law (processor cores)

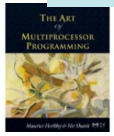


2014: IBM's System z processor 5.7GHz and with 2.75B transistors.

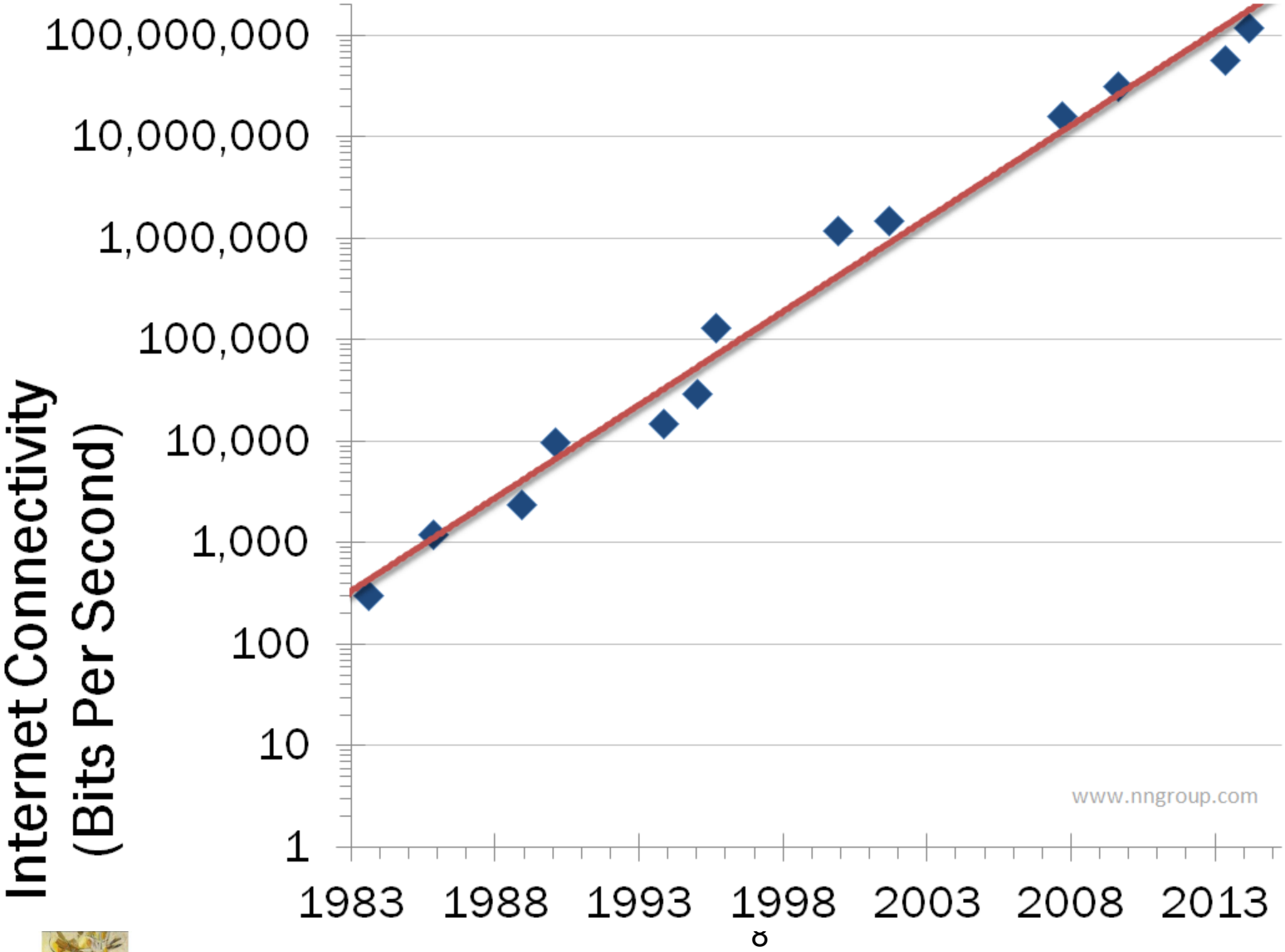
Nielsens' Law (network bandwidth)

"Users' bandwidth grows by 50% per year (10% less than Moore's Law for computer speed)."

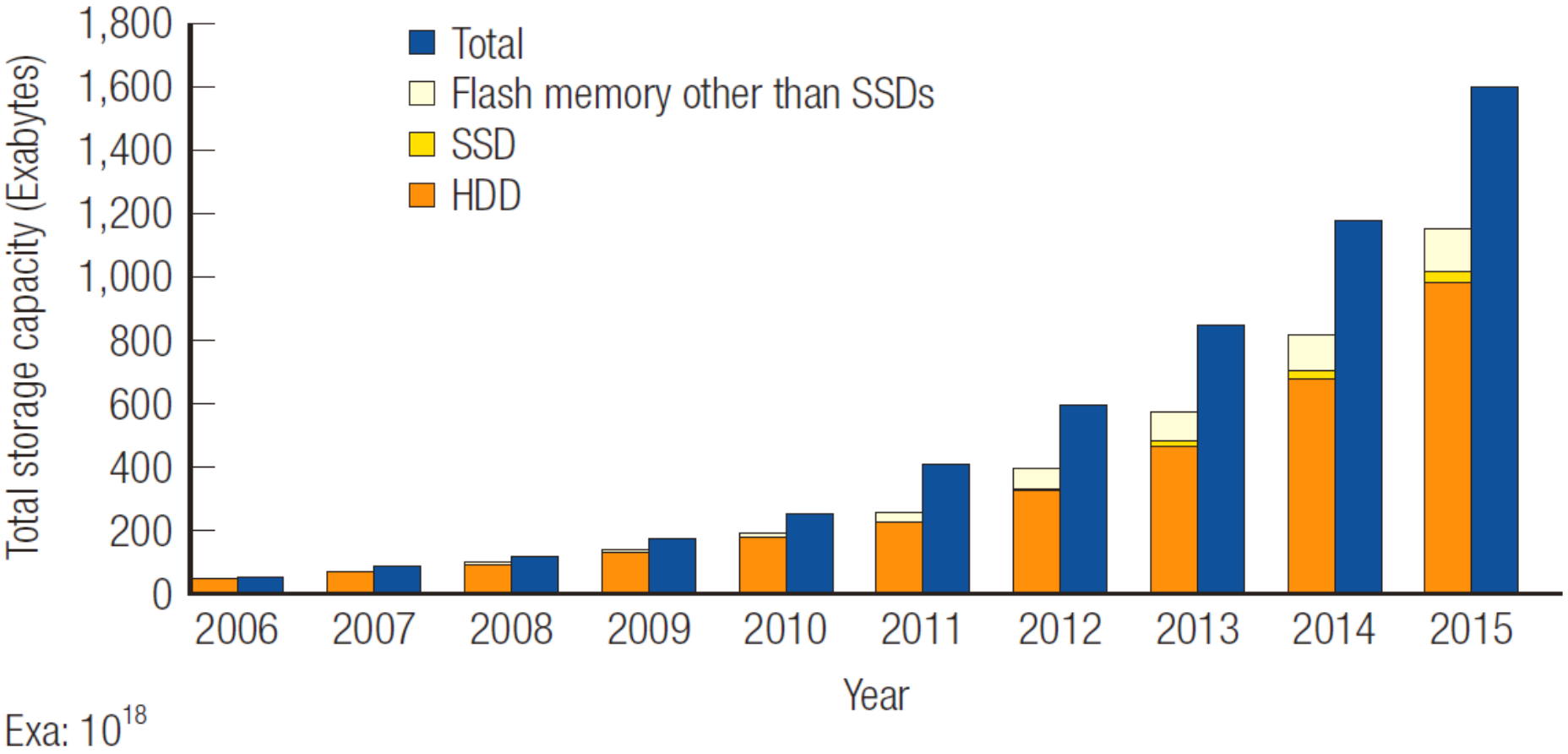
--Jakob Nielsen, **Nielsen Norman Group**



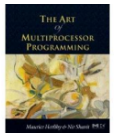
Nielsens' Law (network bandwidth)



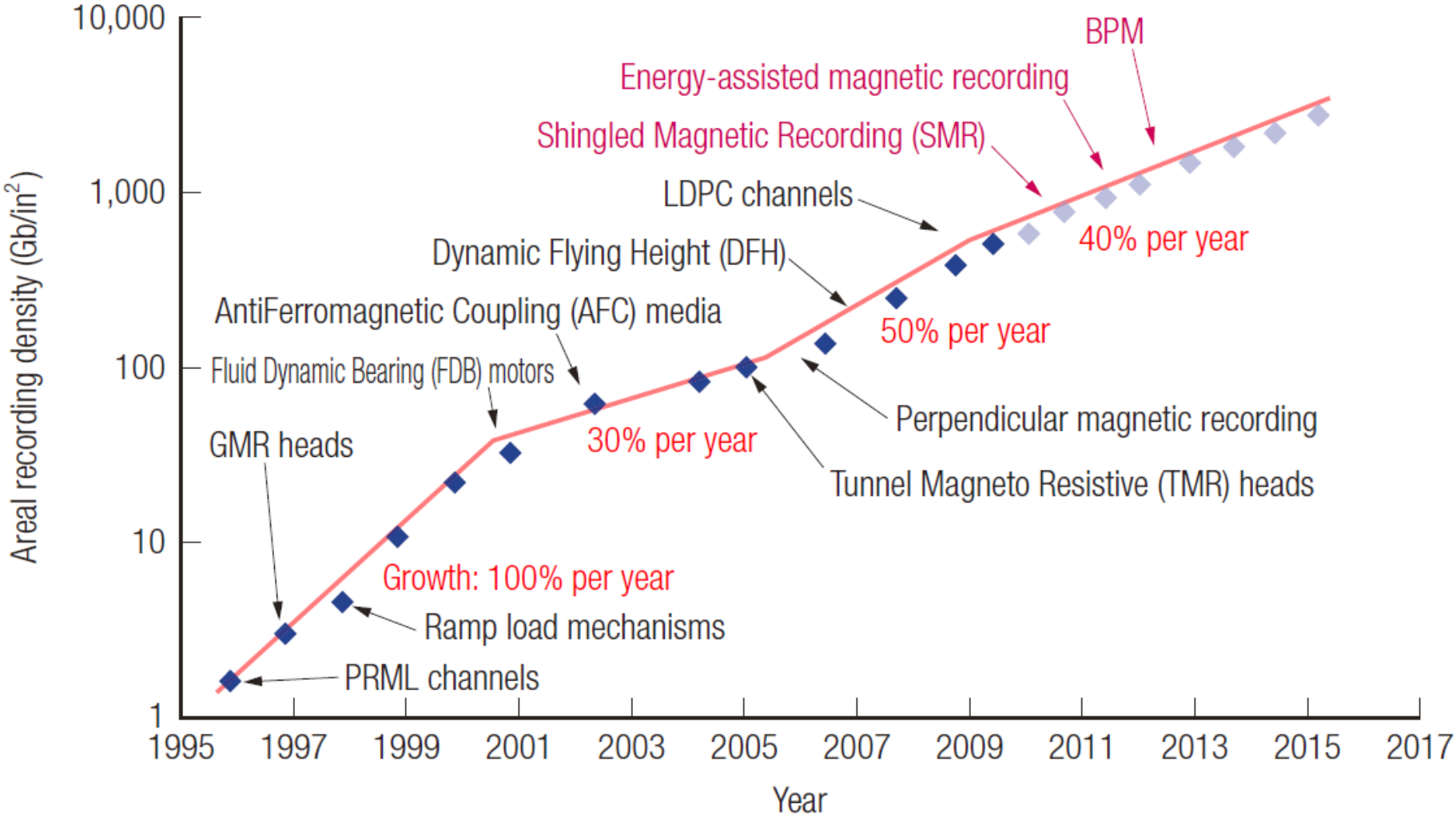
Storage medium (HDD & SSD)



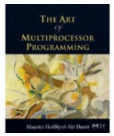
Exa: 10^{18}



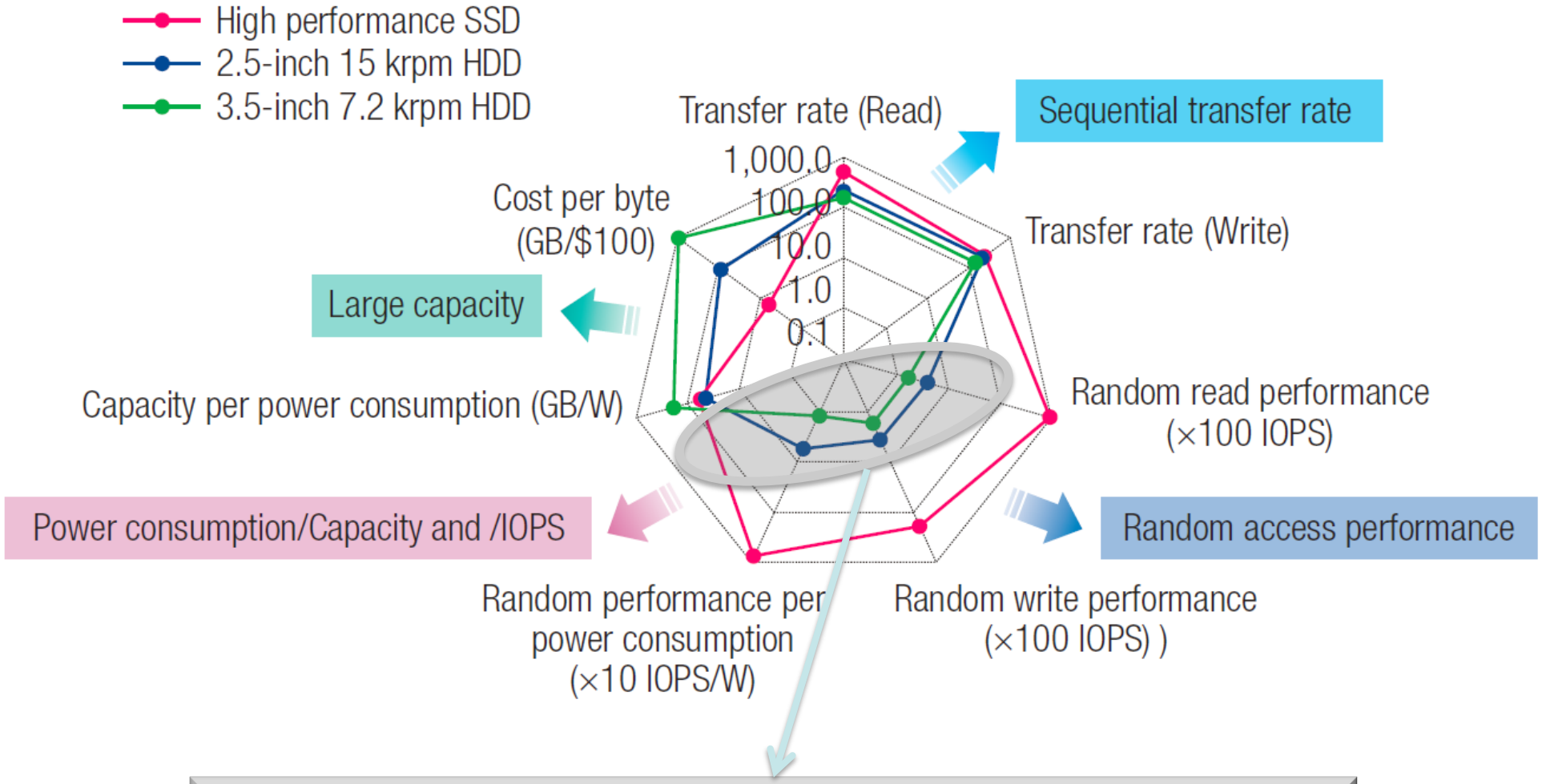
Storage medium (HDD & SSD)



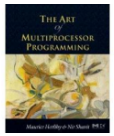
Trends in areal density of HDDs and innovation technologies



Storage medium (HDD & SSD)

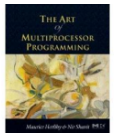


Many legacy system software designs are based on the characteristics of HDD.

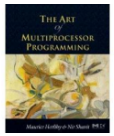


Trend Summary

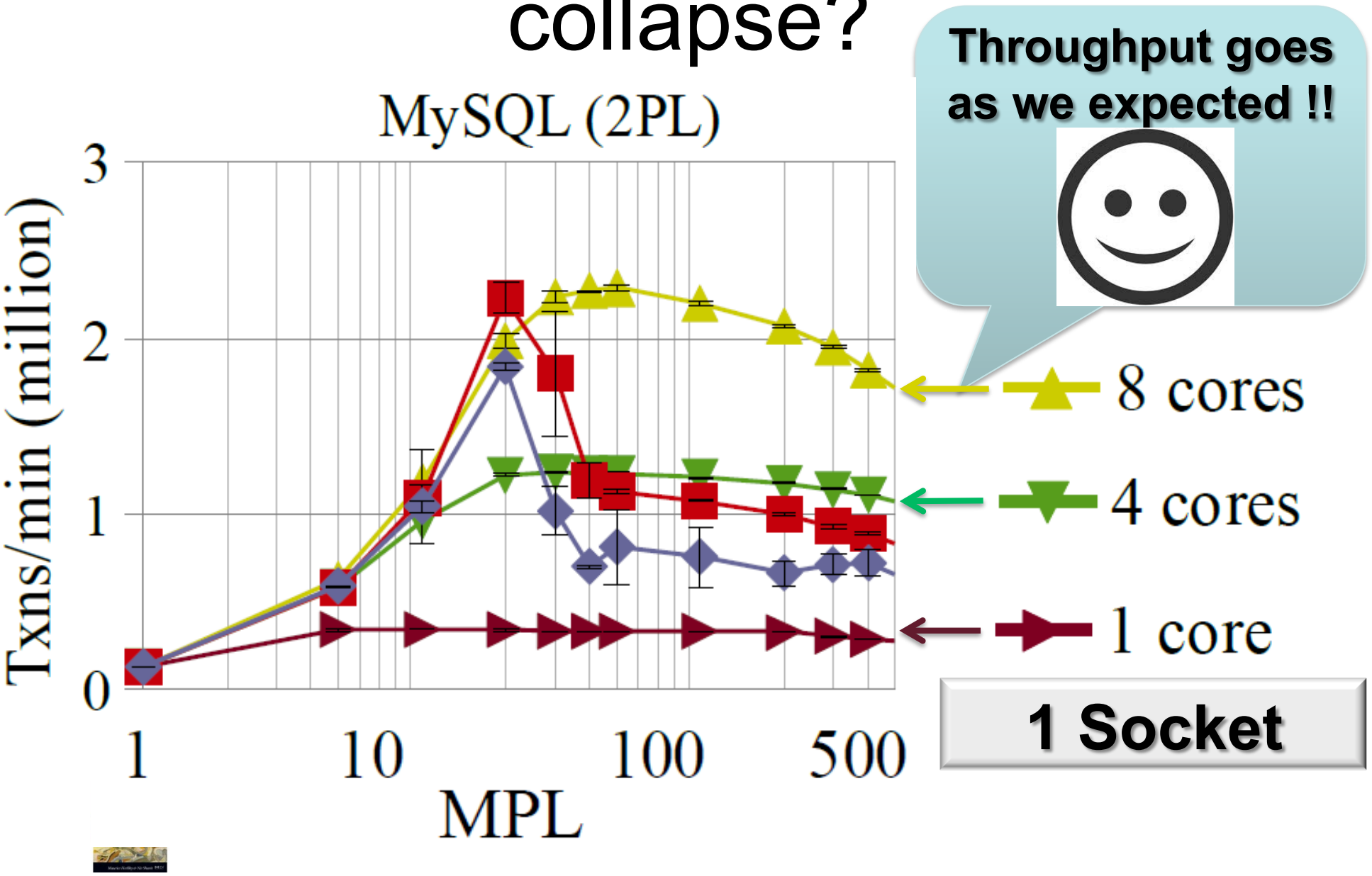
		Annualized Growth Rate	Compound Growth Over 10 Years
Nielsen's Law	Internet bandwidth	**50%**	57×
Moore's Law	Computer power	**60%**	100×



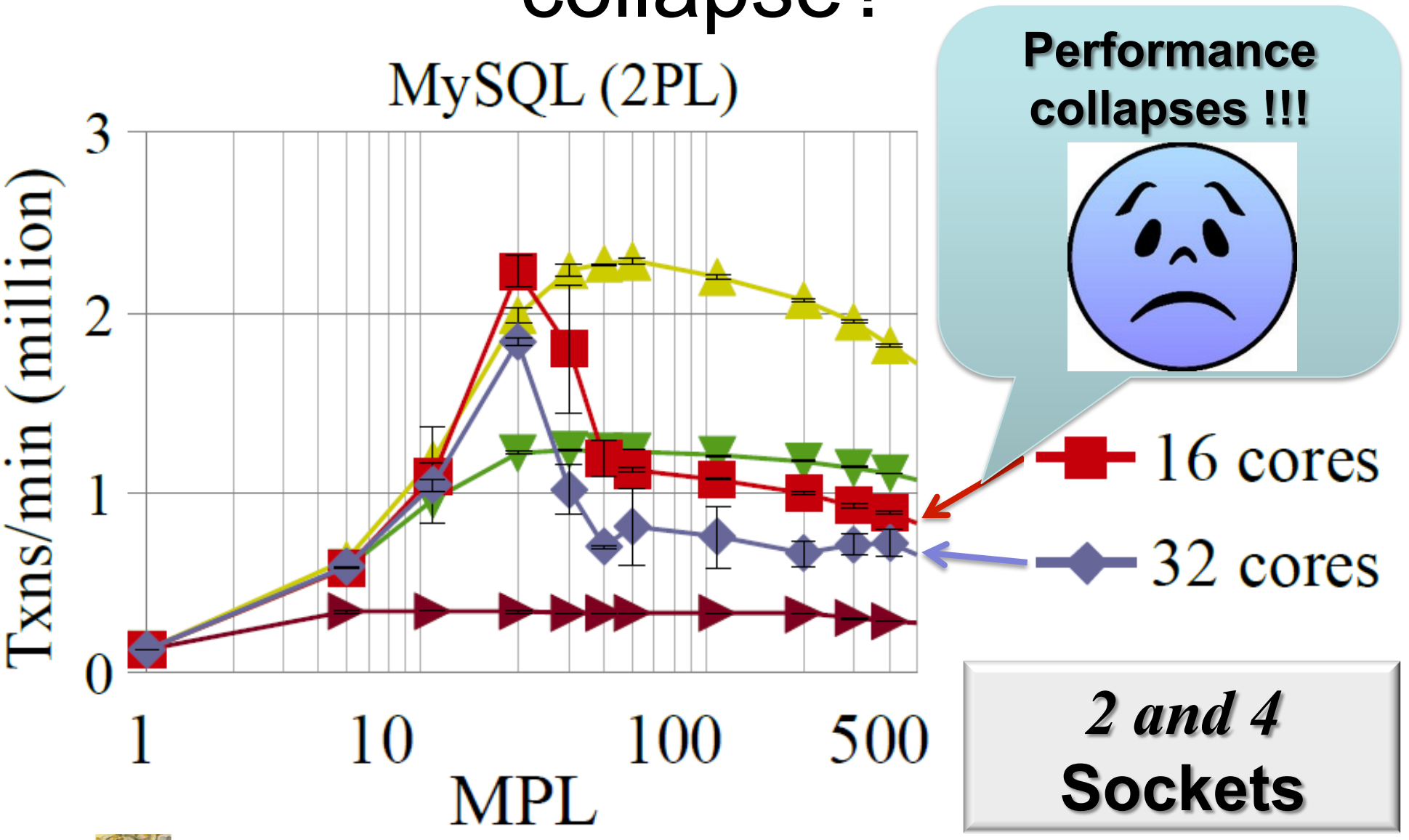
Problems in System Software



How bad is the performance collapse?



How bad is the performance collapse?

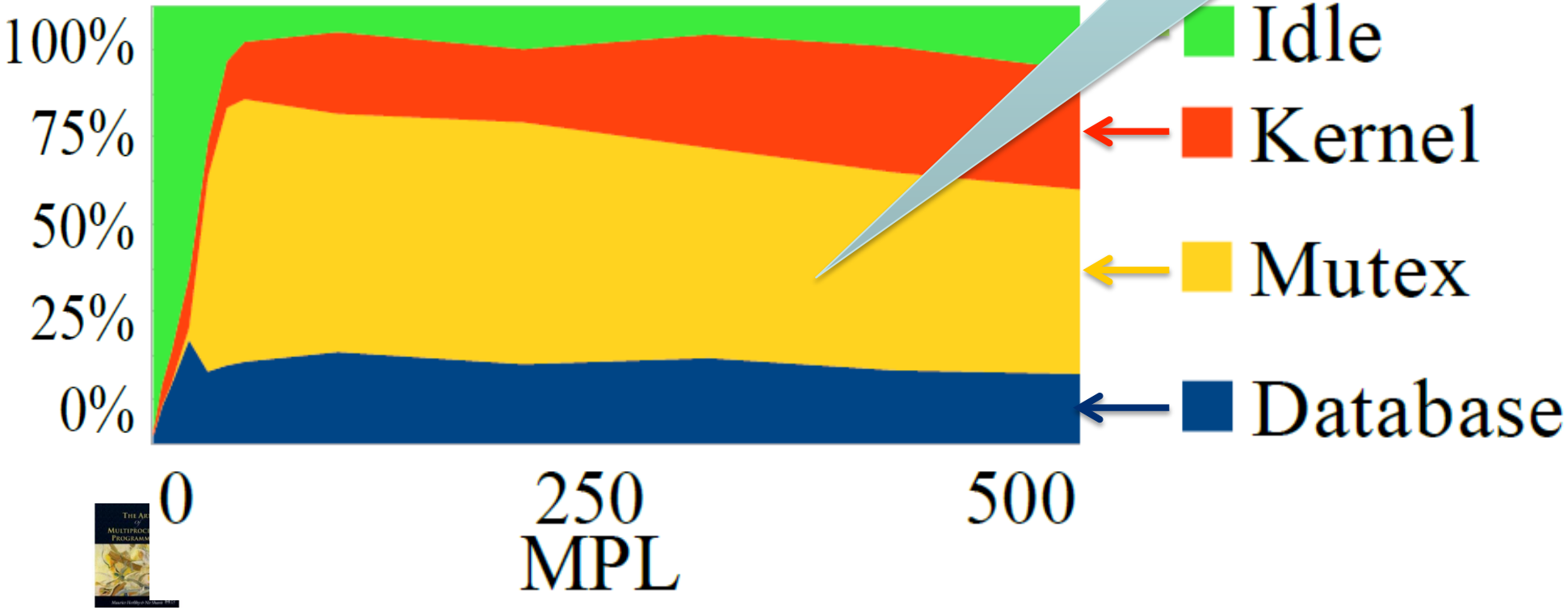


What causes this collapse ?

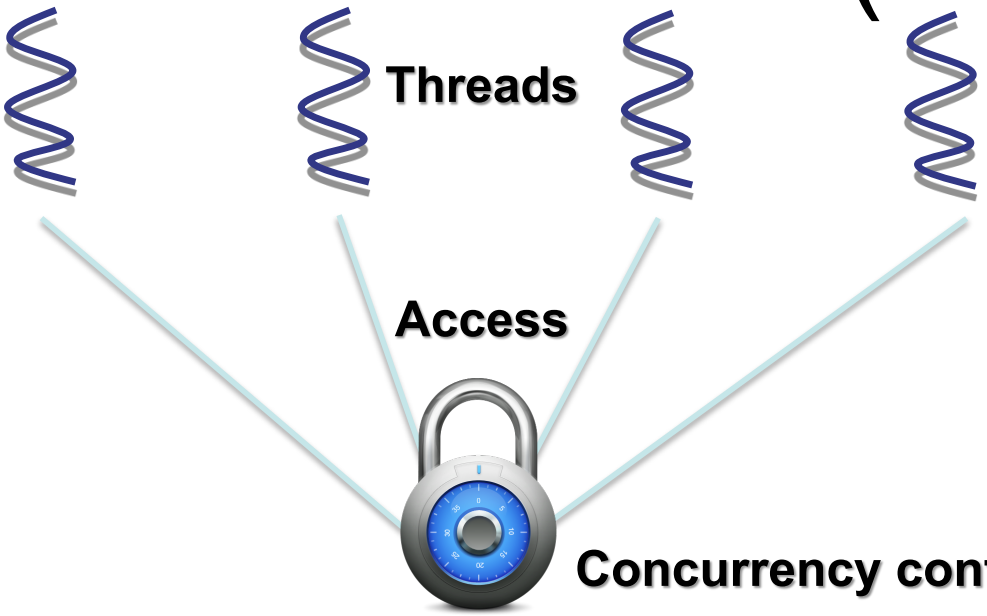
Let's profile databases to peek a little bit deeper inside the system.

Profiling:
read-only queries under "SERIALIZABLE" isolation on 32 cores on 4 sockets

Latch contention is the cause !!!



Mutex (latch)



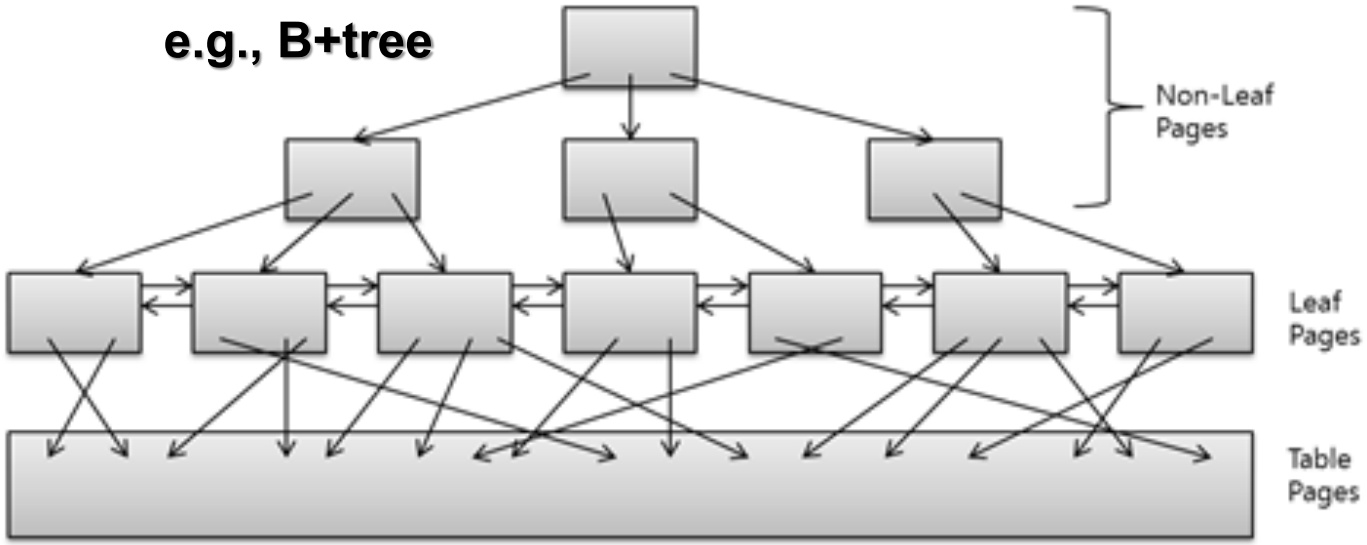
Duration is usually very short

Spin-waiting on contention

This works fine as long as the duration is really short.

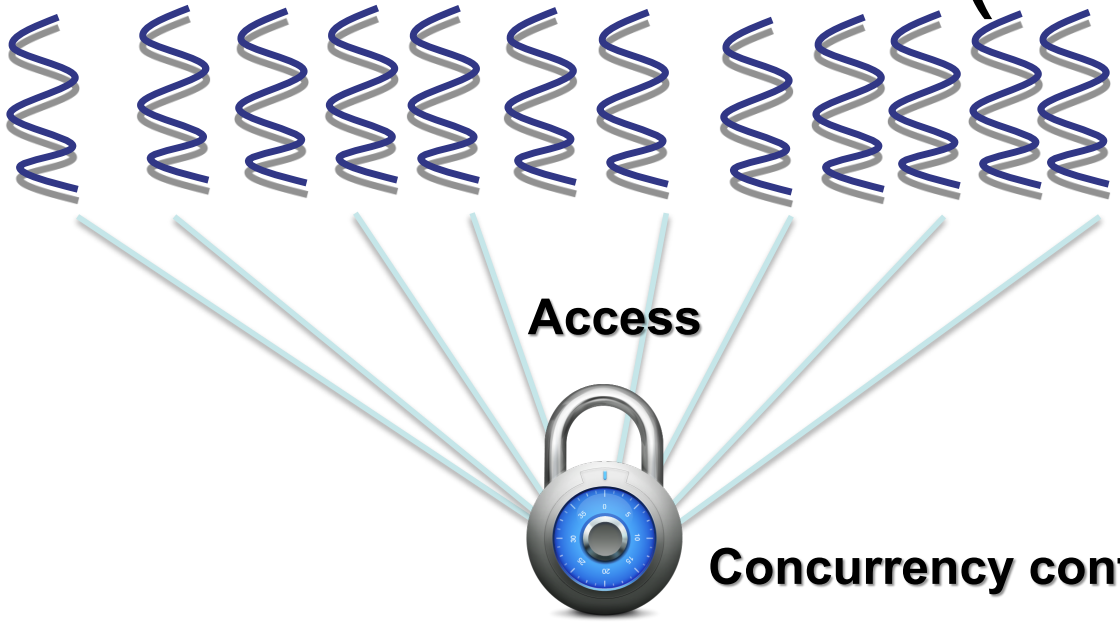
Concurrency control by latches

e.g., B+tree



In-memory Data Structures

Mutex (Latch)

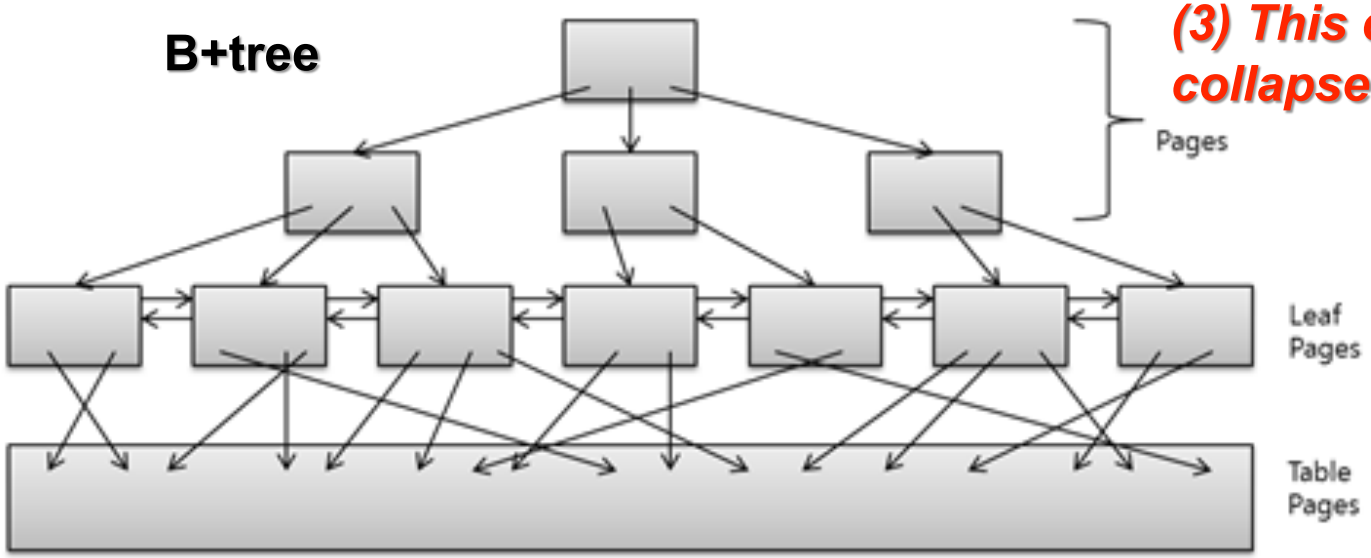


In high contention :

(1) latch duration gets longer

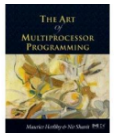
(2) spin-waiting incurs the cache invalidation storm on multicores!!!

(3) This causes performance collapse !!!!



Goal

- Design software (especially core data structures and algorithms) to have multicore scalability !!!



Sequential Computation