other steps. Why longer to perform than any step in computer system? How to overlap? Bottleneck is: how to balance? Good: to hide or avoid performance.

Directing for Performance

1. Performance of Computer System
2. Intro. GFS (Next Class)

Asenja

CSE 536 Fall 05
1. Explore technological innovation to develop new processes.
2. Develop new techniques to improve efficiency.
3. Complement examination.
4. Speed up procedures.
5. Battey billing.

Reasons for change:
- Technology
- Economics

However, due to limited technology, c. 9. speed 011.

Considerable achievement to overcome b.n.s.
Improvement may not be clear.

1. Innovation competency change

2. Be a part of the startup ecosystem

Challenges to designing a system for part.

- Shadily, fairness.
- Need to share resources.

more limits => more breadth => more

- Rel. bi:11
- Algebraic

other kind of limit

speed vs. pace consumption
Refer to worksheet.

1. Throughput: Rate of use for chimp
2. Current pending o/p
3. Latency: Delay b/o c/o change
4. Utilization:
   - % CPU used
   - % Resource
5. Capacity: Size/amount

2. Fournier

in terms of peak

1. Performance requirements are expressed

Performance metrics
\[ T_S = \frac{1}{T_I} \]

**Second system (one feed forward)**

\[ T_S \geq \min (T_A, T_B) \]

Throughput

\[ L_S \geq L_A + L_B \]
and the next slower. In the scenario, we determined the path. As the scenario progressed, we must consider:

we can hope for in reality -

to grow - how much improvement

A systems approach to decisions is part

decision today.

increasing throughput may not require increased, concentrated for these in principle, or

throughout it is certain.

Then is no direct relationship by

Concurrent execution
1. Theoretical Approach to improve flows:

   - Law of diminishing returns

   \[ \text{Expected Return} = \text{Expected Return} \times (1 - \text{fundamental}) + \text{fundamental} \]

   2. Identify bottlenecks

   3. Product import gap analysis

   4. Implement part enhancement

   5. Measure to vital part enhancement

   6. If so what aspect of the

   7. is needed

   8. In measurement of nor part

   9. Measure this part as a system

   Therefore Approach to improve flows
2. Reducing latency by using concurrency:

- Key

Any latency = Free x Latency_fixed + Free x Latency

Free = Rate of Free

1. Reducing latency by exploiting overlap

Technique
Improving throughput: concurrency

1. If latency cannot be improved, then try to hide it.

```
1 -> 2 -> 3 ... -> n
```

- Pipeline / assembly line
- Overlapped execution

$n$-stage pipeline

\[
\text{throughput (concurrent)} \leq n \times \text{throughput of serial execution}
\]
\[ 0 \rightarrow D \rightarrow \text{invc} \rightarrow \text{serv} \rightarrow \text{outc} \]

\[ D : \text{any protocol delay} \]

\[ \sigma : \text{Copy delays over load} \]

\[ D = \sum \frac{1-c}{c} \]

\[ p = \frac{\sigma}{\sigma + D} = \frac{1}{1 + \frac{D}{\sigma}} \]

\[ \text{with copiers through invc in x} \]
Overload can be handled:

1. Increasing capacity
2. Shedding load
3. Deferring requests
4. Batch processing
5. Enforce quota on sources

\[ g = f + h \]

Fifty b.n.

1. Combining multiple requests into one request
2. Proven delay for very fine processing
Speduction: part can go in evidence.

Good is that the result can be different if recurring the evidence.

Increase opportunities for testing.

Reduce latency by recording.

Absorbed work.

Create opportunities to reduce.

Provided additional.

2. Both