The Streamline Computer — or a science of computer architecture

Jun Makino
Department of Planetology, Kobe University
FS2020 Project, RIKEN AICS

Talk Overview

- How we can make the R&D of computer architecture "Scientific"?
- "Streamline"
  - An example: "The Streamline Aeroplane"
  - Another example: The Carnot Cycle
  - The meaning of "Streamline" for computer architecture
- "Streamline" for Application-Specific architectures
  - Regular Grid/Particle/Dense Matrix/Irregular Grid
- General-purpose?
- Summary

How we can make the R&D of computer architecture "Scientific"?

Well, what am I talking about?

Haven't H&P made computer architecture "quantitative", in other words "Scientific"?

Basic methodology of H&P:

1. Make a set of applications
2. Try piecemeal improvements on existing architectures and measure the differential improvements

How we can make the R&D of computer architecture "Scientific"?

Well, what am I talking about?

Haven't H&P made computer architecture "quantitative", in other words "Scientific"?

Basic methodology of H&P:

1. Make a set of applications
2. Try piecemeal improvements on existing architectures and measure the differential improvements

- This approach appears to be "quantitative", but not "Scientific".
- Let's look at truly scientific disciplines

"Streamline"

- **An example: "The Streamline Aeroplane"**
- **Another example: The Carnot Cycle**
- **The meaning of "Streamline" for computer architecture**

The Streamline Aeroplane

B. Melville Jones, The Streamline Aeroplane, Journal of the Royal Aeronautical Society, 33(1929)

THE STREAMLINE AEROPLANE

BY B. MELVILL JONES, A.F.C., M.A., F.R.A$_E$.S.

Ever since I first began to study Aeronautics I have been annoyed by the vast gap which has existed between the power actually expended on mechanical flight and the power ultimately necessary for flight in a correctly shaped aeroplane. Every year, during my summer holiday, this annoyance is aggravated by contemplating the effortless flight of the sea birds and the correlated phenomenon of the beauty and grace of their forms.

We all possess a more or less clear ideal of what an aeroplane should look like; a kind of albatross with one or two pairs of wings—depending on whether we live in Germany or Britain. In our more sanguine moments we even—like Alice and the cat—see the wings without the albatross. But progress towards this ideal, so far as the general purposes craft is concerned is, we must all admit, painfully slow. It has seemed to me that a contributory factor to the slowness of this evolution has been the lack of any generally understood and easily visualized estimate of what could be achieved were the difficulties in the way of realizing the ideal form overcome.

Albatross

Sopwith Camel

(UK WWI fighter aircraft)

They look different

Albatross is clean, and Sopwith Camel is, well, not.

How we can quantify the difference?

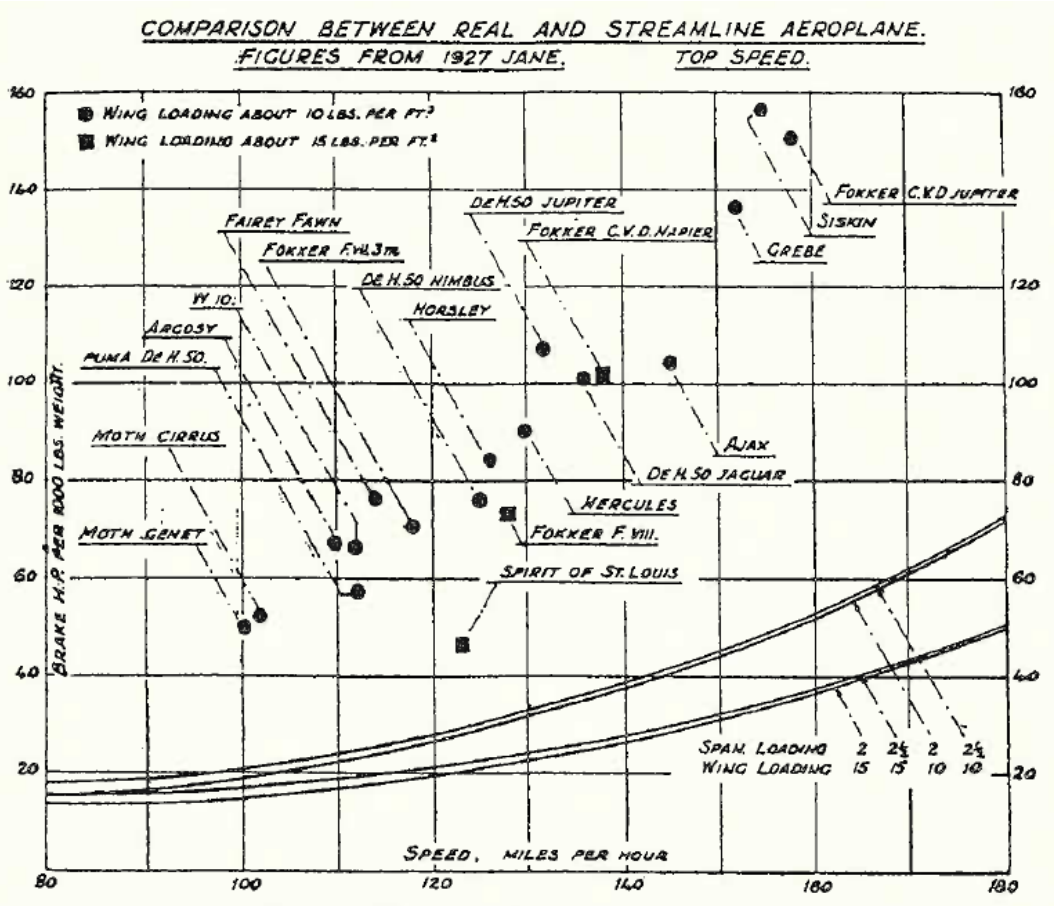
"Looks clean" is not quite enough for science.

Question here: How much we can reduce the aerodynamic drag?

Answer (from fluid dynamics):
There are terms that can be reduced and terms that cannot.

$$\text{Drag} \begin{cases} \text{induced} \quad \text{drag} - \text{remain finite} \\ \text{parasite drag} \begin{cases} \text{Pressure drag} - \text{can be reduced down to zero} \\ \text{Frictional drag} - \text{limit due to total area} \end{cases} \end{cases}$$

Result of measurements



horizontal axis: velocity
vertical: power normalized by weight
Solid curves: theoretical limits
(different curves with span- and area-load)
Points: real aircrafts. Best one: Spirit of St.Louis

Split of St. Louis

Still more than three times the limit
Further possible improvements include:
engine cowl, retractable gear, cantilever wing

Modern aircrafts

A glider

Boeing 787

Even the B787 looks quite smart and close to ideal one.

Another example: The Carnot Cycle

Question here: How much "usable" work we can extract from a heat source

(Final theoretical) Answer: The first and second laws of thermodynamics.

Actual efficiency cannot exceed that of the Carnot Cycle:

$$\eta_c = \frac{T_h - T_l}{T_h}$$

where: T_h temperature of high-temperature source
T_l temperature of low-temperature source(ambient)

Some examples

	T_h(C)	η_c	η_c
Modern NG	**1500**	**0.83**	**0.60**
Nuclear (LWR)	**330**	**0.52**	**0.33**

One need to go to high temperature to achieve high efficiency

The meaning of "Streamline" for computer architecture

- In the above two examples of aircrafts and heat engines, the goal is the energy efficiency.
- For computers, ultimate goal is the energy efficiency.
- For modern machines, at least for HPC machines, the cost of electricity is becoming higher than the hardware cost. Thus, energy efficiency directly determines the available computing power.

Thus,

For a specific calculation, there is a theoretical lower limit for the required energy, and a streamline computer is defined as a machine which achieved that minimum required energy.

(So I'm talking about HPC, not computing in general)

Can we define the "lower limit"?

1. Lower limit depends on the semiconductor technology.
2. Even if we assume that there is a lower limit for a specific application, each application requires specific architecture to realize the lower limit. It is clearly impossible to build a machine for each application, and thus such lower limit is practically useless.
3. Since the algorithms used for applications will change, the lower limit will also change, and we cannot define the lower limit as the long-term target.

We'll discuss each point now.

Point 1

Lower limit depends on the semiconductor technology.

- **Well, in the post-Moore era, the semiconductor technology doesn't evolve as fast as it did in 20C.**
- **Therefore, now it is meaningful to ask: What is the minimum energy consumption for a given semiconductor?**
- **We should be able to give simple and fundamental answer as in the case of aircrafts and heat engines, and such answer should be the basis for the scientific theory for computer architecture.**

Point 2

Even if we assume that there is a lower limit for a specific application, each application requires specific architecture to realize the lower limit. It is clearly impossible to build a machine for each application, and thus such lower limit is practically useless.

- **This was certainly an meaningful argument in 20C. General-purpose machine built with the latest technology outperformed application-specific ones in a a few years.**
- **Now in the post-Moore era, this will no longer happen.**
- **On the other hand, it becomes prohibitingly expensive to make an ASIC for a specific application. We need something else.**

Point 3

Since the algorithms used for applications will change, the lower limit will also change, and we cannot define the lower limit as the long-term target.

- **Computational Science has now the history of 70 years, and basic algorithms for various problems has now become sort of stable.**
- **There will be many changes in details, but the basic concepts like regular grid, irregular grid, particles and graphs will remain unchanged.**
- **Many new methods for parallelization are now being developed, but they are mostly solutions for the problem that hardware is becoming more complex, and does not lead to the reduction of operation count.**

Classification of power consumption

Aircrafts:

- **Drag**
  - **induced drag – remain finite**
  - **parasite drag**
    - **Pressure drag – can be reduced down to zero**
    - **Frictional drag – limit due to total area**

Computers (for HPC)

- **Energy consumption**
  - **Combinatorial Logic for Arithmetic operation**
    - **Dynamic**
    - **Static(leak)**
  - **Storage(Memory, Register)**
  - **Datamovement(Clock, Latch, Wires)**
  - **Controllogic(instructiondecodeetc)**

Dynamic power for arithmetic operations cannot be eliminated. Everything else can.

Expected criticisms

- Data movement is essential for computation and its cost should not be ignored.
- Universality is more important
- This is clearly an extreme argument with little practical meaning.
- Even when we specify an application, we cannot make "others" zero.

In the following, we'll take a look at the last one.

Streamline computers for specific applications

Let's consider

1. Regular grid (neighbor communication only, explicit stepping)
2. Particles
3. Dense Matrix
4. Irregular grid

Regular grid

- For explicit stepping, we can construct a specialized pipeline for arithmetic operations, which would minimize the main memory access.
- Modern high-order, high-accuracy schemes require very large numbers of operations per step per grid point. Thus, memory access cost can be made small.

Particles

- **Operations per particle per step is huge, of the order of 10^4 or more.**
- **Specialized pipeline for particle-particle interaction is always possible.**
- **Thus memory access cost can be made negligible.**

Dense Matrix

- Most operations can be transformed to matrix-matrix multiplications
- By blocking, memory access of matrix-matrix multiplications can be minimized.

Irregular grid

- This is problematic
- Classical CG requires large amount of memory access.
- Multigrid is even more problematic
- On the other hand, in some of modern parallel methods, locally dense matrices are used.
- Will we be using irregular grids and iterative solvers forever?

Streamline computers for specific applications

- Seems possible except for irregular grids.
- iterative methods on irregular grids are and will be problematic on large-scale parallel machines. We probably will need something else.

Thus, we can measure the difference between the theoretical limit and real machines, by measuring the power consumption of arithmetic units and total power consumption.

Where are we now?

Example: 28nm

- **GRAPE-X: 30GF/W, PEZY-SC 25GF/W**
- **AMD FirePro S9150 11GF/W (board)**
- **Intel Xeon E5-2650L (1.8GHz, 8core, 70) 1.65GF/W**

If we measure FPU combinatorial logic only, probably the result would be around 100GF/W.

In other words, even when double-precision operation is required, there is a difference of a factor $3 \sim 60$. For single or half precisions, the difference is even larger.

2020

- TSMC N7+
- Should achieve 6-8 times better power efficiency compared to 28HPM
- 600GF/W
- Even when we consider loss in DC/DC conversion, 300GF/W (DP) should be possible.
- Expected numbers of processors with N7+ are 10-30GF/W. Still a factor of 10 gap.

Streamline General-Purpose computer

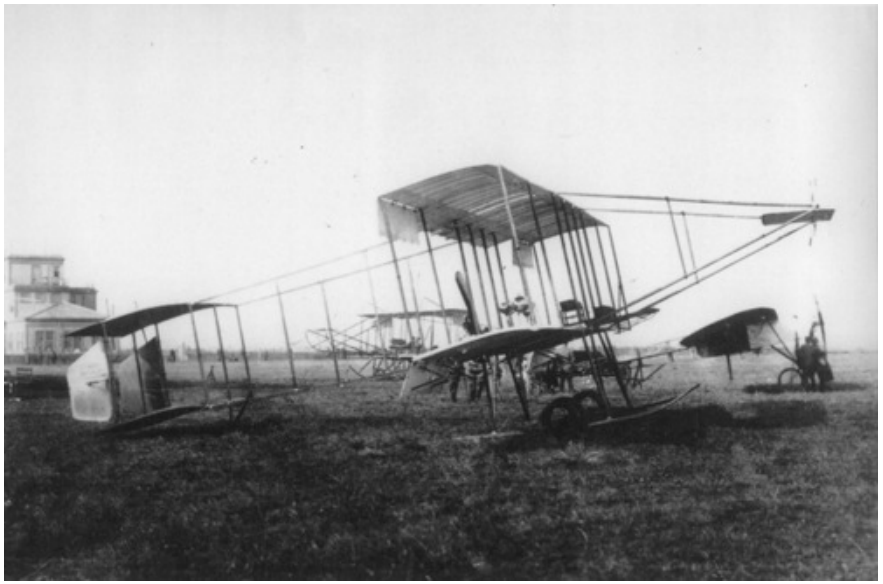
- We can define a streamline computer for each applications
- For general-Purpose computer, one can simply measure the difference with the theoretical limits for several "typical" applications and take whatever mean one like.
- For a given set of applications, there must be an optimal architecture. In other words, this is mathematically well-posed problem.
- So "general purpose" might difficult, but "Multi-purpose" is certainly possible.

Summary

- Concepts like "Streamline Aeroplane" and "Carnot Cycle" played extremely important roles as guiding principles.
- They are important because they define the theoretical limit in what we can do.
- There is no such clear guiding principle which defines the theoretical limit in computer architecture.
- In this talks, I tried to define such theoretical limit, for large-scale numerical calculations.
- It is defined as the power consumption of combinatorial arithmetic logic.
- current computers are far from the limit, by a factor of 10 to 100 or more.

- **We can define and even design "Multi-purpose" streamline computers.**

Computers now and Ideal Streamline computer



Computers now

Ideal Streamline computer