Reinventing Computing

Specialization Theory
(Abstract)

André DeHon
andre@ai.mit.edu

MIT AI Lab
545 Technology Square, Cambridge, MA 02139
April 30, 1995

Informally, we know that special-purpose hardware is faster than or smaller than general-purpose alternatives. Similarly, special-purpose software is often faster and smaller than more general alternatives. In order to better understand the link between specialization and performance, we can adapt many basic concepts from information theory [Sha48]. Specifically, we develop a *specialization theory* which relates the performance of a piece of software or hardware to the information-theoretic entropy of the data it processes. Once this basic link between information content and operational latency is established, we can use it to guide the engineering of high-performance systems in software and hardware. In some restricted domains, we can immediately apply results and concepts from information theory to quantify the benefits of specialization or the costs of generality.

Example

Let us consider conditional branching in a conventional, pipelined microprocessor. Many modern processors allow the branch opcode to statically encode a prediction as to whether or not the branch will be taken. When the branch is correctly predicted the branch costs a single cycle to execute. When the branch is mispredicted, the pipeline is flushed and refilled with instructions along the non-predicted branch. Consequently, the branch costs d cycles instead of one. Typically, d is roughly equal to the pipeline depth of the processor.

The branch nominally handles one piece of information – *i.e.* the bit which says whether or not to take the branch. However, the information content in this bit may be much less than one bit of information. In fact, the lower the information content in the bit, the greater the predictability of the branch and the higher the performance which can be achieved with static branch prediction.

Note here, that we are using the entropy of the quantity under question as the metric of its information content. Recall from information theory (see [Sha48] or an information theory text such as [CT91]), that, in general, the entropy of a variable X is:

$$H(X) = -\sum_{x \in \text{Alphabet}(X)} p(x) \log p(x) \qquad (1)$$

For a binary variable X (Alphabet$(X) = \{0,1\}$), with $p = Pr\{X = 1\}$ the entropy is:

$$H(X) = -p \log_2 p - (1-p) \log_2 (1-p) \qquad (2)$$

Entropy measures the uncertainty in a random variable. Intuitively, the entropy of a variable gives us a lower bound on the number of bits we will need to encode an average instance of the variable in an optimal encoding scheme.

The following table summarizes the entropy and average branch costs as a function of various branching probabilities.

P_{branch}	Information Content H(branch decision)	Branch Latency ($d = 3$)
0.50	1	2
0.10	0.47	1.2
0.90	0.47	1.2
0.01	0.08	1.02
0.99	0.08	1.02

Acknowledgments: This work is conducted under the direction of Principal Research Scientist Thomas Knight Jr. This research is supported by the Advanced Research Projects Agency of the Department of Defense under Rome Labs contract number F30602-94-C-0252.

Observations

On a wide variety of examples, we can make the following observations:

- The achievable performance on an application is limited by the information content inherent in the problem's input data. The lower the information content, the higher the achievable performance. *e.g.*, The lower the entropy of the branch decision, the lower the average branch cost.
- The actual performance of an implementation depends on the information content as handled by the implementation. If the implementation acts like there is more information present than there really is, the implementation will run correspondingly slower than an optimal implementation for said information content.
- Implementation building blocks with rigid interfaces limit one's ability to optimally match the information content processed by an implementation to the information content inherent in a problem and data set.
- Information content is often much lower than the maximum entropy information condition which an isolated, primitive operation might encounter because:
  - *Static* information inherent in the overall problem structure limits information content.
  - Typical use is not uniform making some values more likely than others. Deviations from the flat distribution of values, the maximum entropy state, have lower entropy and hence reduced information content.
  - Correlations, either static or probabilistic, between nominally distinct values can reduce total information content. *e.g.*, two values x and y may be related to each other in some functional manner or they may be highly correlated in practice. In such a case, the combined information content of the two variables is less than the sum of the individual information contents.

Specialization Principle

Based on these observations, we can informally formulate the principle result of specialization theory as:

> *The latency of an operation is roughly proportional to the information content processed by the operation.*

While many pragmatic and subtle issues complicate the precise formulation and application of specialization theory, this informal statement is broad and intuitively conveys the fundamental result of the theory. [DeH95] provides more detailed development of specialization theory.

The branch prediction example is certainly not an isolated one. Most of the performance enhancements in modern processor architectures, including caches, virtual memory, prefetch, block transfers, and exceptions, exploit conditions where the entropy of the data handled is significantly lower than the quantity of bits processed. Similarly, many circuit families, operating system architectures, and compiler optimizations improve their performance by exploiting conditions of low average information content.

Specialization theory provides us with an intuitive and uniform understanding of these techniques. In restricted domains, it allows us to be quantitative about the opportunity for and benefits of specialization. A focus on the information content present in applications benefits optimization of architectural organization, program structure, and data structures. At all levels of design, specialization theory focuses attention on the real issues associated with achieving high performance, exposing opportunities where specialization can be employed profitably.

References

[CT91] Thomas Cover and Joy Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., New York, 1991.

[DeH95] André DeHon. Specialization Theory. Transit Note 123, MIT Artificial Intelligence Laboratory, March 1995.

[Sha48] Claude E. Shannon. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27:379–423, 623–656, 1948.