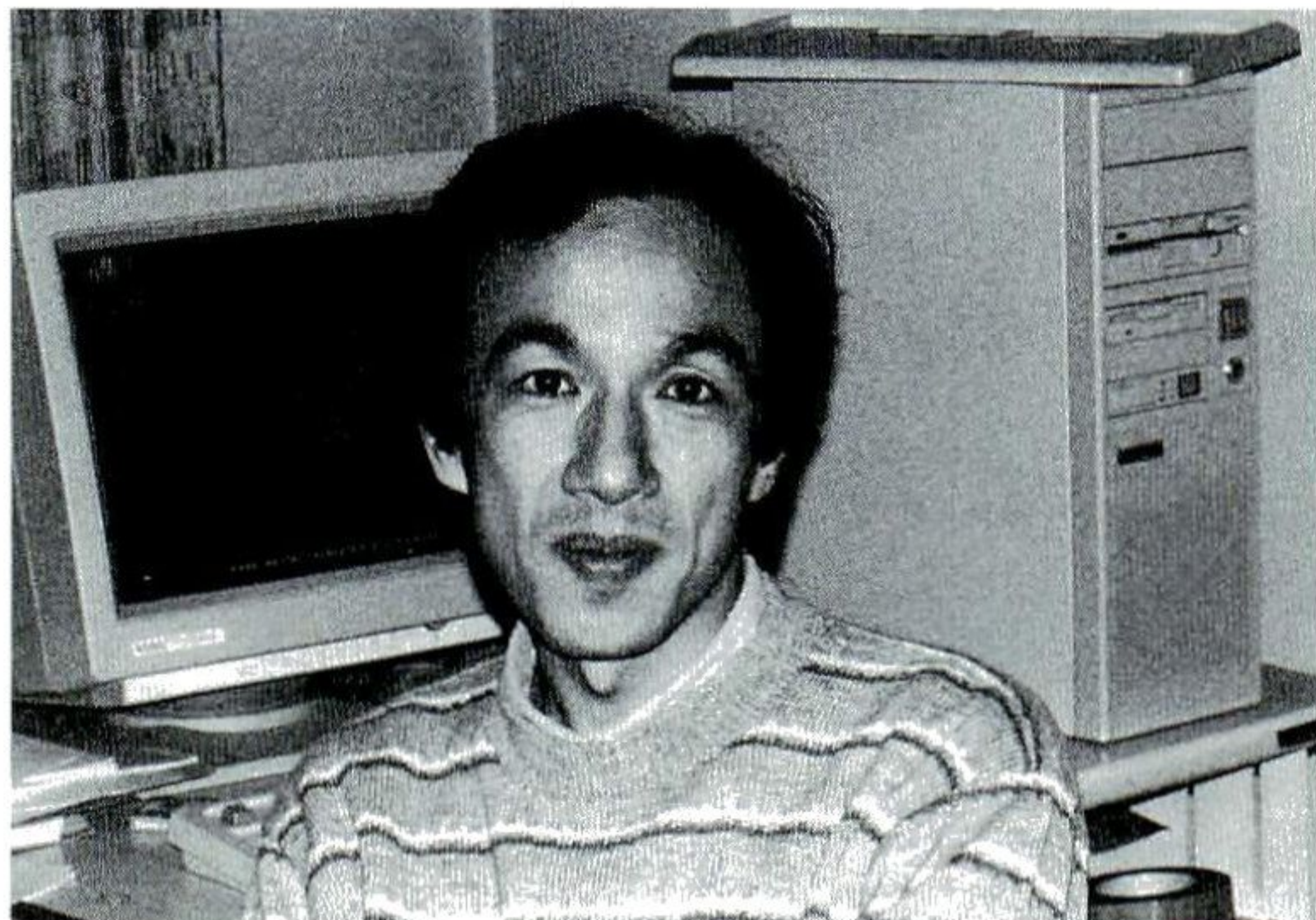




たくさんのコンピュータをどう使う？

—— 柴山研究室～情報科学科 ——



柴山 悦哉 助教授

「コンピュータはソフトウェアがなければただの箱である」と言われるように、コンピュータにとってソフトウェアは必須の存在です。しかし、ハードウェアがなければ、ソフトウェアの開発もまた不可能なのです。最近、ハードウェア技術の進歩によってコンピュータシステムに変化があらわれ始めています。その1つが、仕事を複数のコンピュータに分散させて処理する分散システムの登場です。今後は、このようなコンピュータシステムを有効に利用するためのソフトウェアが求められていくことでしょう。このような状況をみて、柴山研究室では並列分散的に動作するソフトウェアについて研究しています。

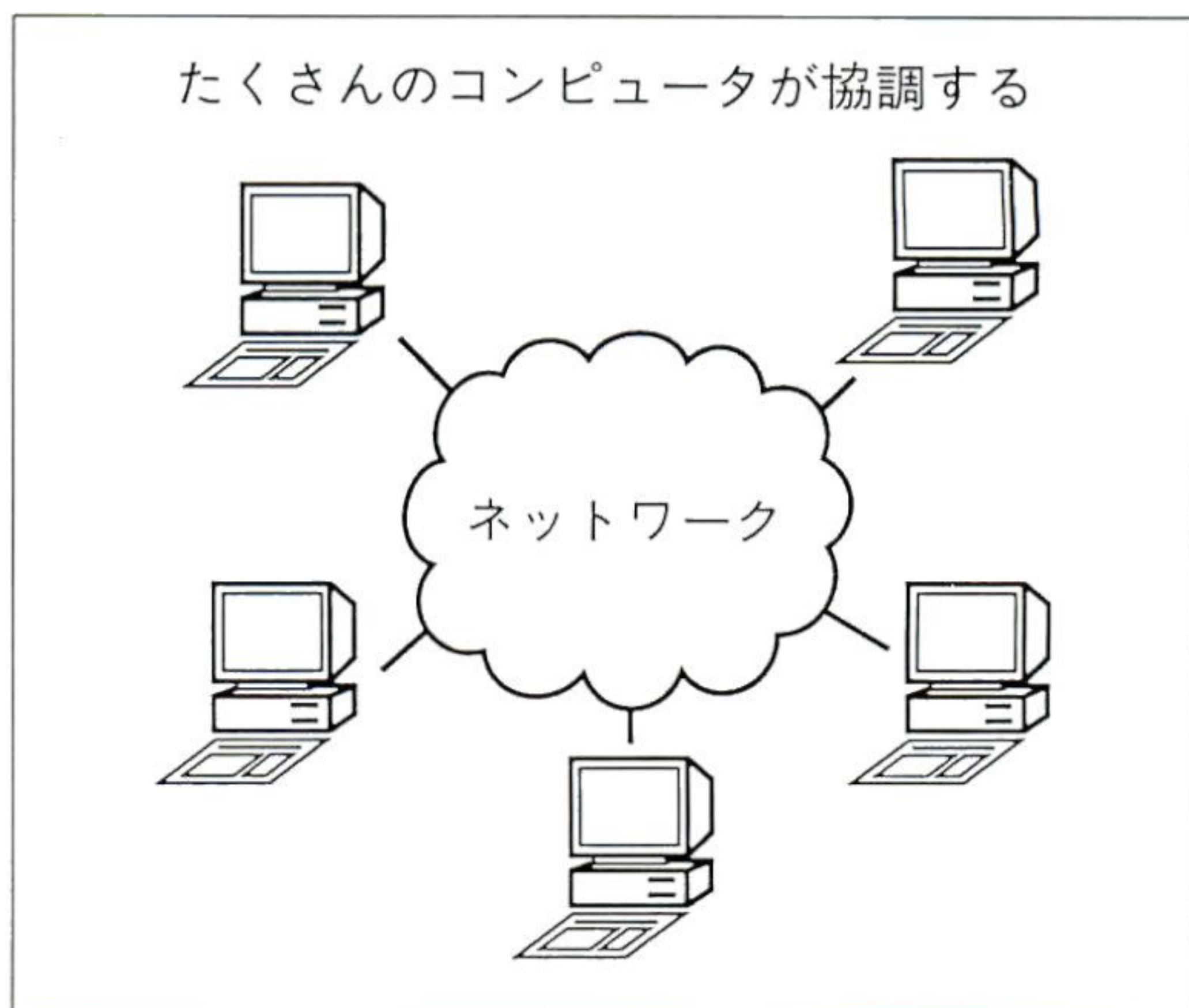


これからのコンピュータシステム

コンピュータの普及と共に、コンピュータどうしをつないで、互いに利用するためのネットワーク技術が発達してきています。しかし、今のネットワークは、基本的にコンピュータの間で情報のやり取りに利用されているだけです。このネットワークをもっと有効に利用できないでしょうか。分散システムというものがこの解答の1つとして考えられています。この分散システムでは、ネットワークにつながっているコンピュータが協力して仕事をします。つまり、あるコンピュータが、大きな仕事をしているとき、その仕事が分割できるようなものならば、その一部を他のコンピュータに手伝ってもらうことによって、仕事を早く終わらせることができるのです。このように、分散して存在しているコンピュータが、ある目的のために協調する。それが分散システムなのです。

かつては、同じ能力をもつコンピュータを2台作るより、その2倍の能力をもつコンピュータを1台作るほうが安くすむ時代がありました。つまり、コンピュータの能力を集中させることに経済

面でのメリットがあったのです。そのため、当時のコンピュータシステムというと、どこか中央に高い能力を持ったコンピュータがあり、そこにいくつもの端末機をつなげて、それをみんなで利用するという形態が主流でした。このような形態のコンピュータシステムを集中型のシステムと言



分散システム

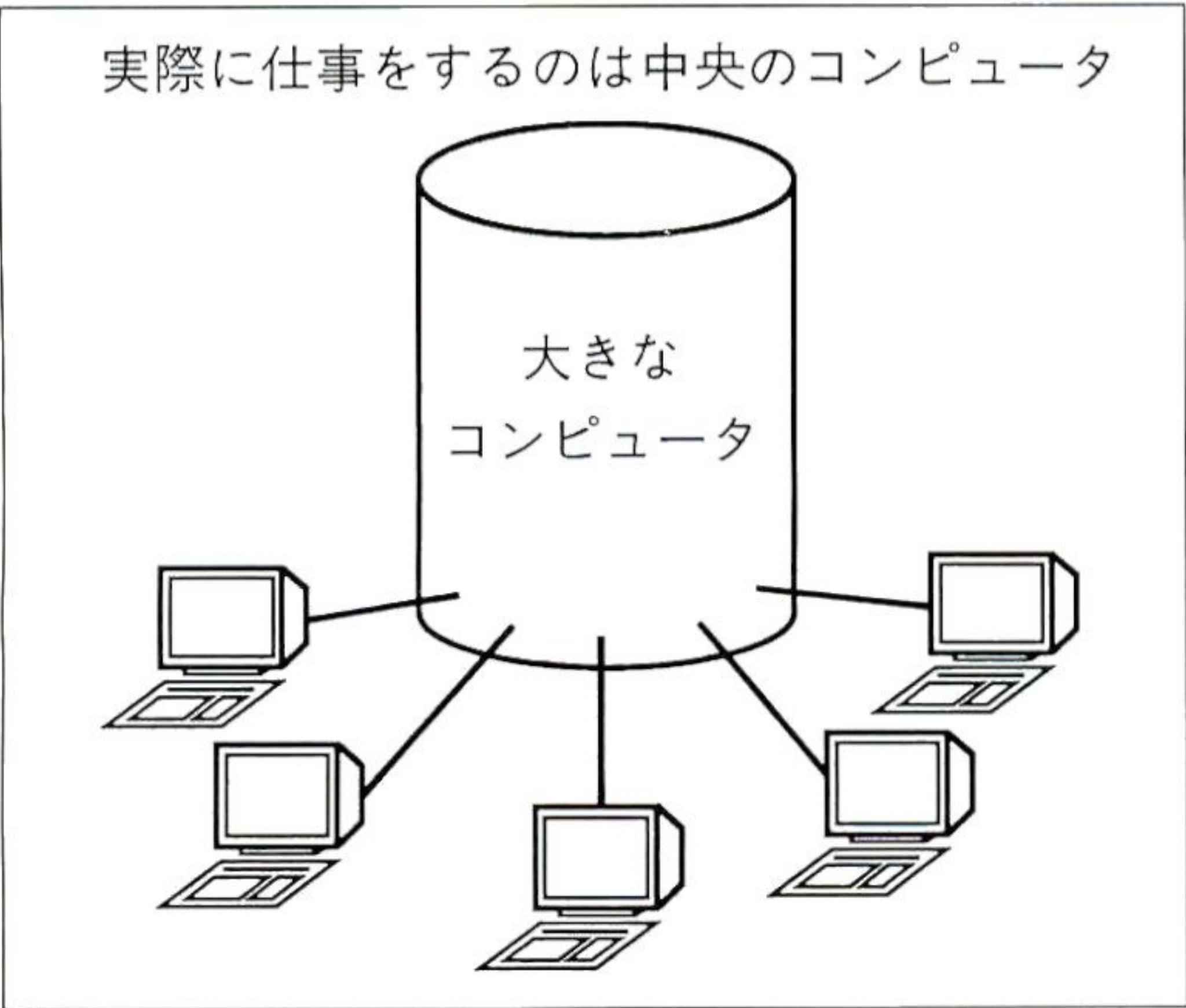
ます。

しかし、ハードウェアの技術の進歩によってこの状況は変わってきています。1台のコンピュータの能力を上げるには限界が見えてきましたが、既存のコンピュータの方は、より安く作れるようになってきたのです。そして現在、あらたに高い能力を持つコンピュータを開発するよりも、既存のコンピュータを量産するほうが、経済的にみて効率が良い時代になっています。

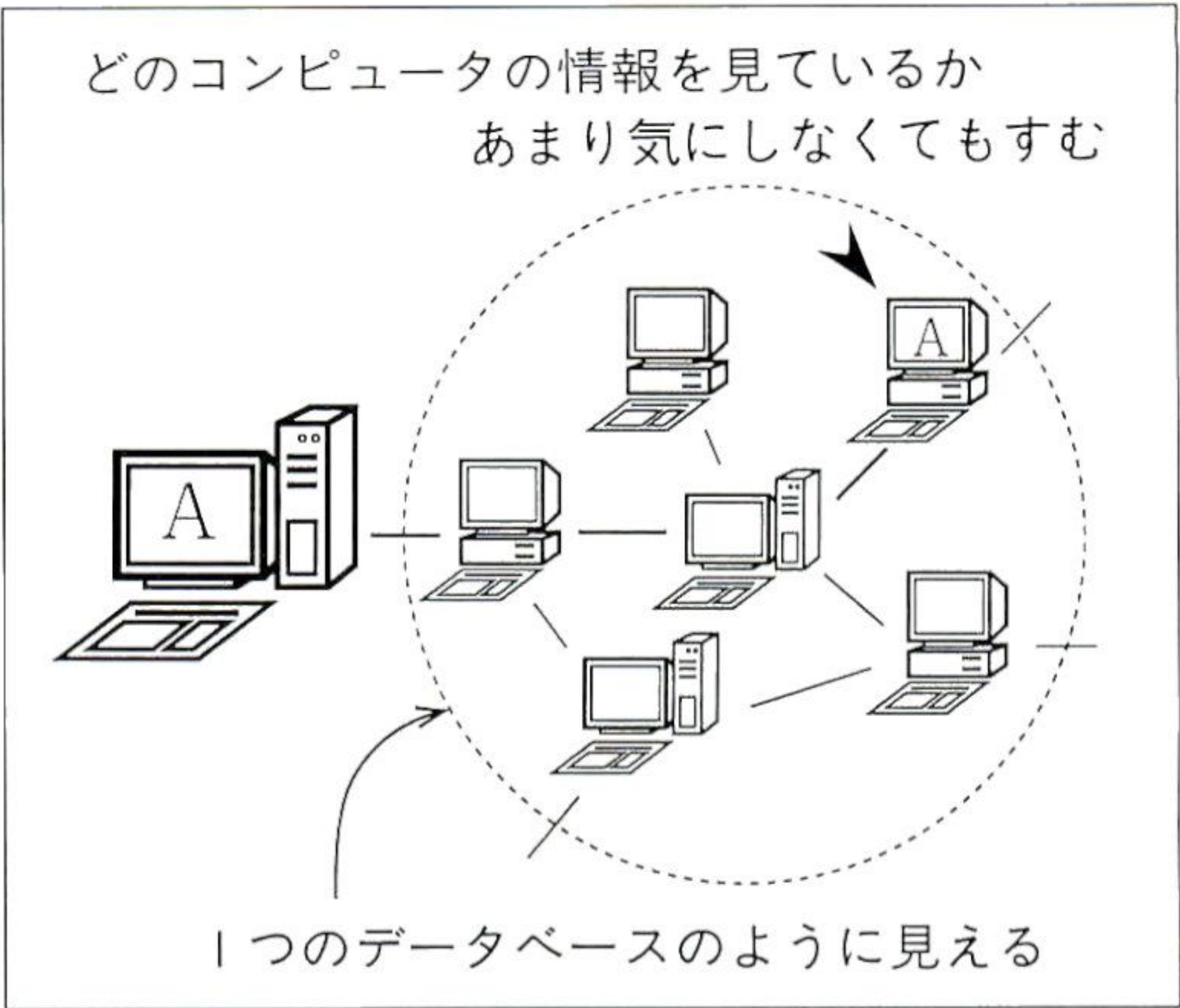
このような中で、たくさんの小型のコンピュータを接続し、集中型のシステムのように大きな処理能力を持ったシステムが作れないだろうかと考えられました。そして、複数のコンピュータに仕事を分散させて処理する分散システムが、その利点と共に注目を集めることになったのです。

1つのコンピュータになんでも任せていた形態から、その仕事を複数のコンピュータに分散させて処理する形態に変える。これによってどのような利点が生まれるのでしょうか？

1つは、故障に強くなるということです。集中型のシステムでは、1つのコンピュータがすべての処理を受け持っていました。そのため、もしも中央のコンピュータが故障したら、その時にはシステムすべてが使用不能になってしまいます。しかしこれを複数のコンピュータに任せるという形にすれば、仕事は複数のコンピュータで分散して処理されます。そのため、そのうちの1つが壊れても、すこし能力は落ちるかもしれませんが、システム全体が使用不能になるようなことはありません。このような故障に対する強さは、金融や医



集中型のシステム



インターネットでは

療などの分野では大変重要になるでしょう。

もう1つは、拡張がしやすくなることです。集中型のシステムでは、システム的能力を上げたいと思ったら、中央のコンピュータをまるごと変える必要があります。これに対して、分散システムでは、小型のコンピュータを増設するだけでシステム的能力を上げることができるのです。これにより今までシステムを構成していたコンピュータを無駄にせずにすみません。これらの利点は、ともに分散システムの柔軟性を示しているといえるでしょう。

他にも分散システムにはメリットがあります。それはシステムを構成するそれぞれのコンピュータを地理的に離して配置できることです。それぞれのコンピュータは離れているにもかかわらず、全体としては高い能力を持ったシステムになるのです。一方、たいして大きな処理能力を必要としない仕事なら、通信をせずに1つのコンピュータだけで処理することもできます。これにより通信コストや通信時間を節約することが可能となるのです。

このようにいろいろな利点を持つ分散システムですが、今の分散システムは、どのような段階にあるのでしょうか。ごく初歩的なものであればすでに実用化されています。

最近、インターネットと言うものが何かと騒がれています。このインターネットを利用するためのソフトウェアは、ネットワークを意識せずに、あたかも1つの巨大なデータベースを検索するように、世界中のコンピュータから情報を取り

寄せることができるのです。

このようにネットワークを意識せずに、複数のコンピュータ上に存在する情報を取り寄せるようなソフトウェアは確立されています。しかし「情報を取り寄せる」だけでなく「1つの巨大なコン

ピュータのように計算する」ところまでになると、まだ世間では見かけることはできません。これを実現させることが今後の分散システムの課題であるといえそうです。



協調作業するオブジェクトたち

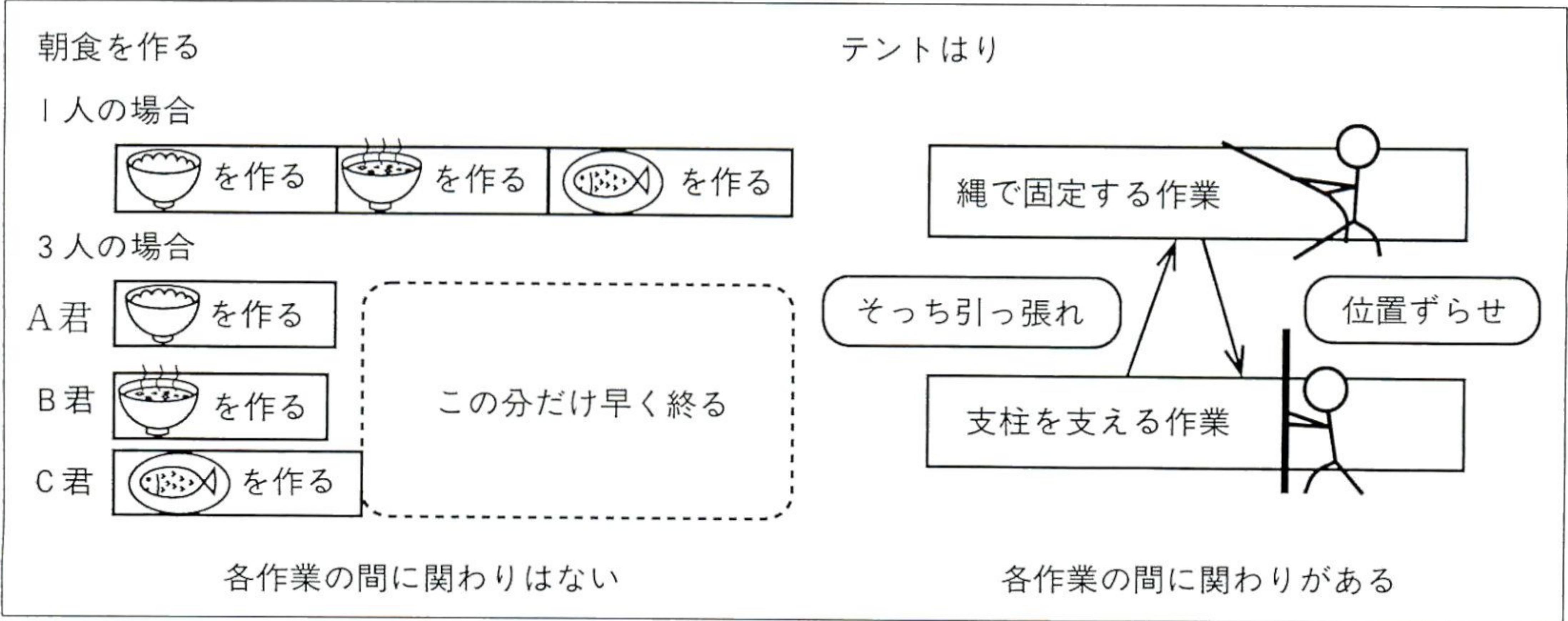
先の分散システムの話でもわかるように、これからは1人の人間が複数のコンピュータを利用するような状況が増えてくるでしょう。そのため、1台のコンピュータを利用していたときとは違うことを考える必要が出てきます。これは、いままで1人で行っていた仕事を複数の人で行うには、新しい仕事のやり方を考える必要があるのと同じことです。うまく仕事を割り振ることができれば、1人でやっていたときよりもかなり早く仕事を終えることができます。例えば、朝食を作る場合を考えてください。献立は、ごはんと味噌汁と焼き魚ということにします。これを1人で行う場合は1つずつ料理を作っていくことになるでしょう。しかし、もしこれを3人で行うならそれぞれの料理を分担して作ることで、早くこの仕事を終えることができます。

これと同じことが、コンピュータを動作させるためのプログラムを作る際にも言えます。いままでの1台のコンピュータを動作させるためのプログラムは「どのような作業をどのような手順で行うか」を指定していました。つまり「まずごはん、つぎに味噌汁、そしてそのつぎは焼き魚という順

序で料理を作りなさい」という具合にコンピュータに命令をしていたのです。このようなプログラムを手続き型のプログラムと言います。

しかし、このようなプログラムをそのまま複数のコンピュータで処理することは大変困難です。なぜなら、全部が1つにつながった形のプログラムを見ても、コンピュータにはそれがどういう作業に分割できるのか分からないからです。それでは、複数のコンピュータを利用するためのプログラムはどのように作ったら良いのでしょうか。そのためには「どの作業が同時に処理できるのか」をプログラム中で指定する必要があります。つまり「ごはんを作る、味噌汁を作る、焼き魚を作る、という3つの作業がある。これらは同時に処理することができます」とプログラム中で指定するのです。

ところが、実際の仕事はいま見たようにうまくいくわけではありません。なぜなら、複雑な要素がからみ合い、しかもそれぞれの作業が密接に関連しあっているからです。例えばキャンプでテントを張る場合を考えてください。この仕事を複数の人で行うとしたらどうなるのでしょうか。テント



いろいろな仕事

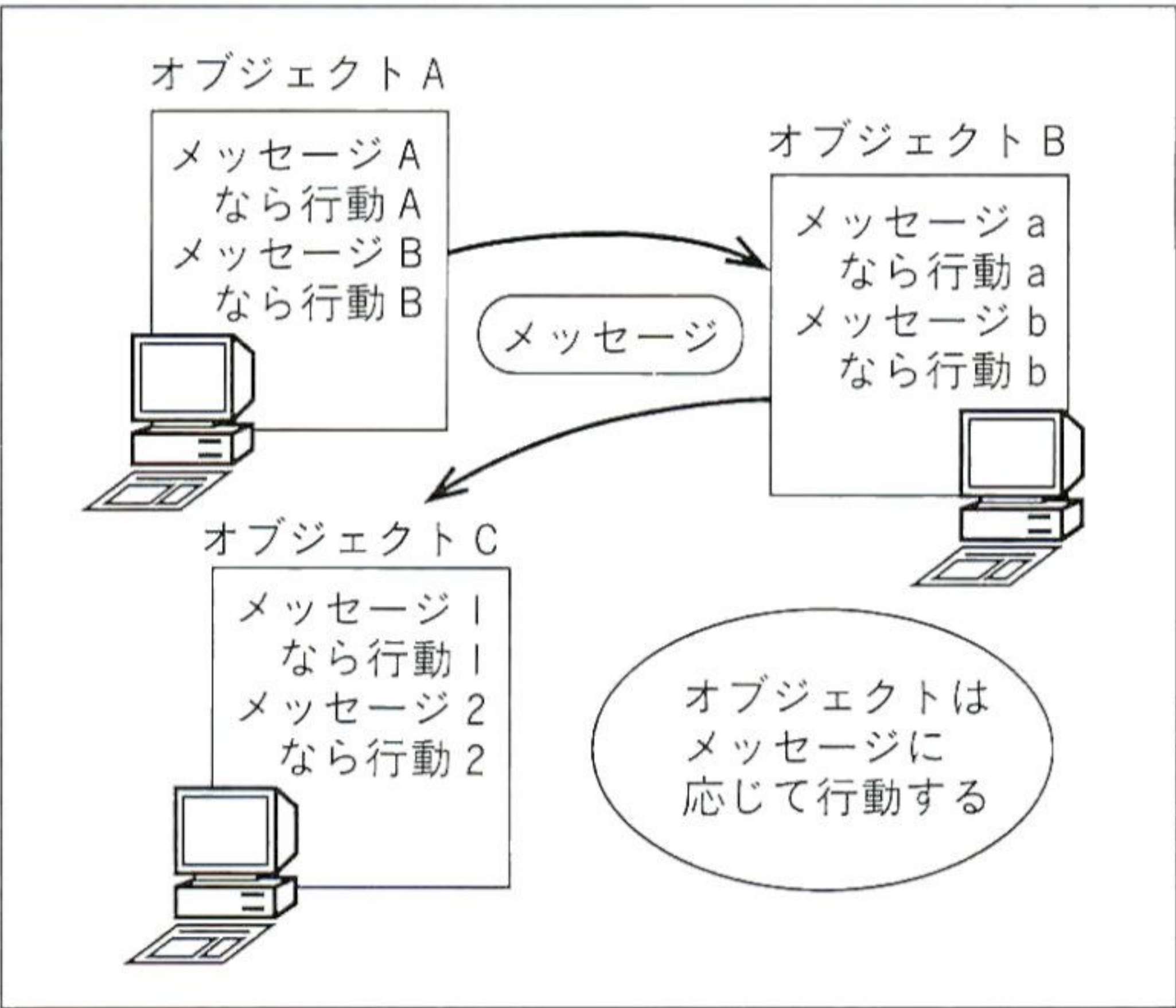
の支柱を固定することについてだけを考えてみても、縄を張って固定する作業、支柱を支える作業などがあります。この作業の間に何の関係もなければ、先ほどの朝食の話のように簡単に複数のコンピュータで処理できます。しかしながらこの作業の間には関わり合いがあります。例えば、どちらかの縄を引っ張り過ぎて支柱が傾いてしまった場合などには、支柱を支えている人に支柱の位置をずらしてもらう必要が出てきます。このように1つの作業が他の作業に影響を与えてしまうのです。そのため「それをもう少し引っ張って」とか「その支柱の位置ずらして」というような会話をを行い、各作業の間で調整をしながら仕事が進んでいくことになります。

それでは、このような仕事はうまく複数のコンピュータで処理できないのでは？と思う人もいるでしょう。実は、これを可能にするために「並列オブジェクト指向」という考え方があるのです。並列オブジェクト指向のプログラムでは、このような仕事をするとき、それぞれの作業をするオブジェクトがメッセージを送り合います。これがテント張りをする人たちの会話に相当するのです。つまり、オブジェクトとは自立して作業する「労働力」であり、それらが相互にメッセージを送り合いながら作業を進めていくのです。

このような協調作業をすべて1人の指示で行っていたらどうでしょう。「そこは支えて、そこは引っ張って…」というふうに。これでは指示する方も大変ですし、作業の効率も悪くなることがかえらるでしょう。ここに、いままでの「手続き型のプログラム」と「並列オブジェクト指向のプログラム」の差があるのです。このどちらが良いかは仕事の性質にもよりますが、一般的に複雑な

要素がからみあっている仕事をさせようとするときは、並列オブジェクトの考え方をした方が、よりプログラムを組みやすくなることがわかんと思います。

そしてなぜ柴山先生がこの並列オブジェクト指向を取り上げたかといえ、このように「協調作業するオブジェクト」を中心として組まれたプログラムこそ、分散システムで動かすのにふさわしいと考えられたからなのです。「分散システムでプログラムを動かす」というのは、仕事を複数のコンピュータに分担させて動かすことですから、「この作業とあの作業が同時にできる」というようにプログラム中で指示してあれば、コンピュータはそれを分散させて動かすことができます。つまり、1つのコンピュータに1つのオブジェクトを割り当てて動かしてもらえば良いわけです。この並列オブジェクト指向は、分散システムで動かすソフトウェアを構築するうえで大きな手助けとなることでしょう。



オブジェクト指向

最後に柴山先生に東工大の学生について伺ってみましたところ、次のように話されました。「最近の東工大の学生はおとなしくて、エンジニアにはなれるけど営業はできないなというタイプが多い。もし研究者になるのなら、自分の研究の売り込みをする必要があるから、やっぱり営業マン的センスはいるんです。そういう意味で言えば、おとなしいのは心配です」やはり、研究者を目指すなら、自分の主張をはっきり表現できる必要があるとい

うことでしょう。
今回の取材では、私達のコンピュータに関する知識が足りないために、先生にかなり苦勞をおかけしました。それでも熱心に説明してくださった柴山先生に感謝するとともに、先生の研究の成果が、皆さんに感じられるようになる日が早く来ることを期待しています。

(林崎 昌弘)