Preferably, the network would involve no payment between users and providers, with the network system (software, hardware, etc) providing an essentially equivalent usage of computing resources by both users and providers (since any network computer operated by either entity can potentially be both a user and provider of computing resources (even simultaneously, assuming multitasking), with potentially an override option by a user (exercised on the basis, for example, of user profile or user's credit line or through relatively instant payment).

Preferably, as shown in FIG. **4**, the priority and extent of use of PC and other users can be controlled on a default-to-standard-of-class-usage basis by the network (provider or other) and overridden by the user decision on a basis prescribed by the specific network provider (or by another level of network control). One obvious default basis would be to expend up to a PC's or other user's total credit balance with the provider described above and the network provider then to provide further prescribed service on an debt basis up to some set limit for the user; different users might have different limits based on resources and/or credit history.

A specific category of PC user based, for example, on specific microprocessor hardware owned or leased, might have access to a set maximum number of parallel PC's or microprocessors, with smaller or basic users generally having less access and vice versa. Specific categories of users might also have different priorities for the execution of their processing by the network. A very wide range of specific structural forms between user and provider are possible, both conventional and new, based on unique features of the new network computer system of shared processing resources.

For example, in the simplest case, in an initial system embodiment, as shown in FIG. **4**A, a standard PC **1** user request **11** for a use involving parallel processing might be defaulted by system software **13**, as shown in FIG. **4**E, to the use of only one other essentially identical PC 1_2 microprocessor for parallel processing or multitasking, as shown in FIG. **4**C; larger standard numbers of PC microprocessors, such as about three PC's at the next level, as shown in later FIG. **10**G (which could also illustrate a PC **1** user exercising an override option to use a level of services above the default standard of one PC microprocessor, presumably at extra cost), for a total of about four, then about 8, about 16, about 32, about 64 and so on, or virtually any number in between, would be made available as the network system is upgraded over time, as well as the addition of sophisticated override options. Eventually many more PC microprocessors would be made available to the standard PC user (virtually any number), preferably starting at about 128, then about 256, then about 512, then about 1024 and so on over time, as the network and all of its components are gradually upgraded to handle the increasing numbers. System scalability at even the standard user level is essentially unlimited over time.

Preferably, for most standard PC users (including present and future equivalents and successors), connection to the Internet (or present or future equivalents or successors like the MetaInternet) would be at no cost to PC users, since in exchange for such Internet access the PC users would generally make their PC, when idle, available to the network for shared processing. Preferably, then, competition between Internet Service Providers (including present and future equivalents and successors) for PC user customers would be over such factors as the convenience and quality of the access service provided and of shared processing provided at no addition cost to standard PC users, or on such factors as the level of shared processing in terms, for example of number of slave PC's assigned on a standard basis to a master PC. The ISP's would also compete for parallel processing operations, from inside or outside the ISP Networks, to conduct over their networks.

In addition, as shown in FIG. **5**, in another embodiment there would be a (hardware and/or software and/or firmware and/or other) controlling device to control access to the user's PC by the network. In its simplest form, such as a manually activated electromechanical switch, the PC user could set this controller device to make the PC available to the network when not in use by the PC user. Alternatively, the PC user could set the controller device to make the PC available to the network whenever in an idle state, however momentary, by making use of multitasking hardware and/or software and/or firmware and/or other component (broadcast or "push" applications from the Internet or other network could still run in the desktop background). Or, more simply, as shown in FIG. **5**A, whenever the state that all user applications are closed and the PC **1** is available to the network **14** (perhaps after a time delay set by the user, like that conventionally used on screensaver software) is detected by a software controller device **12** installed in the PC, the device **12** would signal **15** the network computer such as a server **2** that the PC available to the network, which could then control the PC 1 for parallel processing or multitasking by another PC. Such shared processing can continue until the device **12** detects the an application being opened **16** in the first PC (or at first use of keyboard, for quicker response, in a multitasking environment), when the device **12** would signal **17** the network computer such as a server **2** that the PC is no longer available to the network, as shown in FIG. **5**B, so the network would then terminate its use of the first PC.

In a preferred embodiment, as shown in FIG. **6**, there would be a (hardware and/or software and/or firmware and/or other component) signaling device **18** for the PC **1** to indicate or signal **15** to the network the user PC's availability **14** for network use (and whether full use or multitasking only) as well as its specific (hardware/software/firmware/other components) configuration **20** (from a status **19** provided by the PC) in sufficient detail for the network or network computer such as a server **2** to utilize its capability effectively. In one embodiment, the transponder device would be resident in the user PC and broadcast its idle state or other status (upon change or periodically, for example) or respond to a query signal from a network device.

Also, in another embodiment, as shown in FIG. **7**, there would be a (hardware/software and/or firmware and/or other component) transponder device **21** resident in a part of the network (such as network computer, switch, router, or another PC, for examples) that receives **22** the PC device status broadcast and/or queries **26** the PC for its status, as shown in FIG. **7**.

In one embodiment, as shown in FIG. **8**, the network would also have resident in a part of its hardware and/or software (and/or firmware and/or other components) a capacity such as to allow it to most effectively select and utilize the available user PC's to perform parallel processing initiated by PC users or the network providers or others. To do so, the network should have the (hardware and/or software and/or firmware and/or other component) capability of locating each PC accurately at the PC's position on the geographic grid lines/connection means **23** so that parallel processing occurs between PC's (PC **1** and PC 1_2) as close together as possible, which should not be difficult for PC's at fixed sites with a geographic location, customarily grouped together into cells **24**, as shown in FIG. **8**, but which