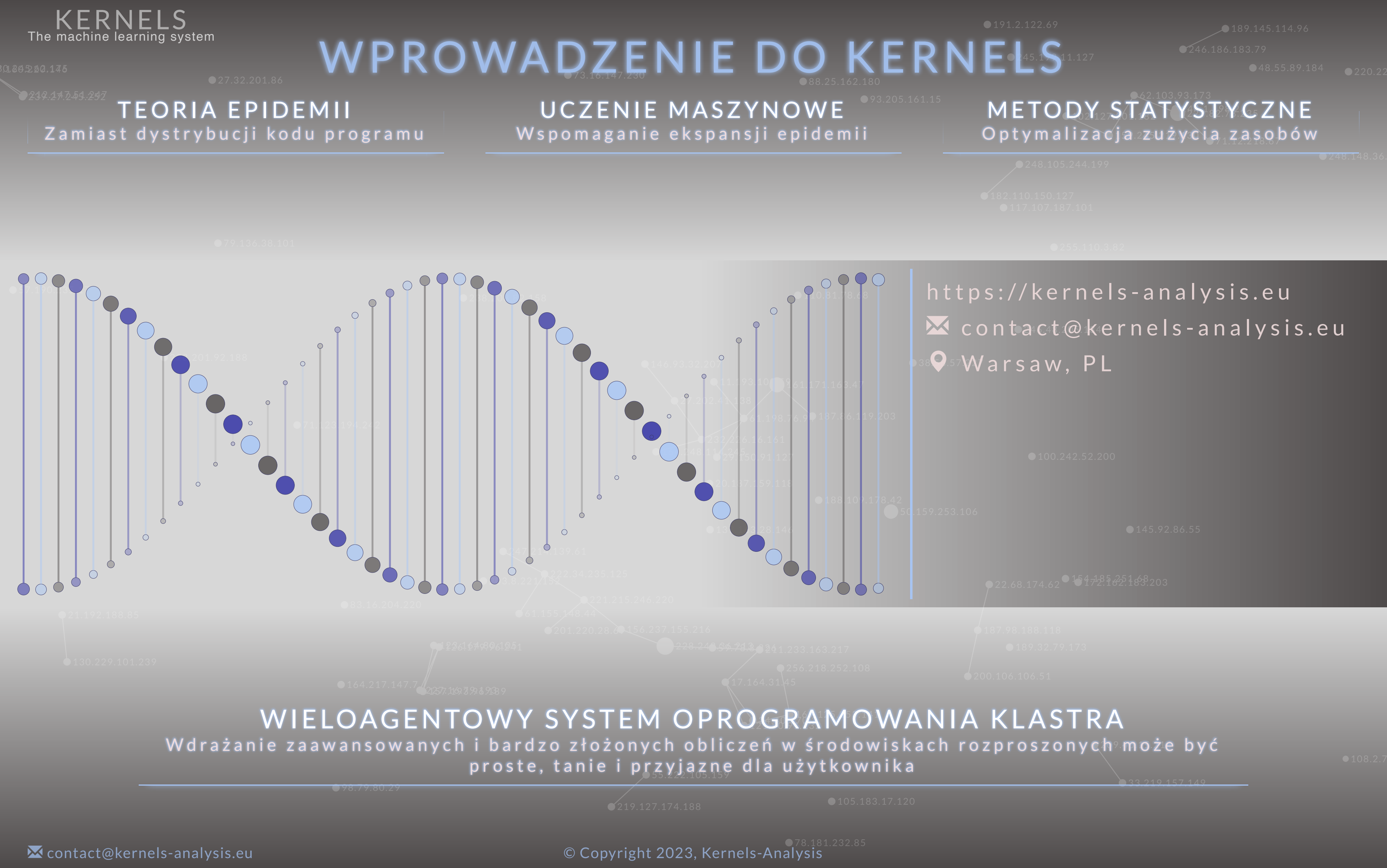




WPROWADZENIE DO KERNELS

TEORIA EPIDEMII
Zamiast dystrybucji kodu programu

UCZENIE MASZYNOWE
Wspomaganie ekspansji epidemii

METODY STATYSTYCZNE
Optymalizacja zużycia zasobów

<https://kernels-analysis.eu>
✉ contact@kernels-analysis.eu
📍 Warsaw, PL

WIELOAGENTOWY SYSTEM OPROGRAMOWANIA KLASTRA
Wdrażanie zaawansowanych i bardzo złożonych obliczeń w środowiskach rozproszonych może być proste, tanie i przyjazne dla użytkownika

WPROWADZENIE

- 1. Przyczyny powstania KERNELS
- 2. Wykorzystanie teorii epidemii

I

DLACZEGO KERNELS?

- 1. Zalety pracy z KERNELS
- 2. Teoria epidemii, a Twoje obliczenia
- 3. Bezpieczeństwo rozwiązywania
- 4. Architektura sprzętowa

II

CZYM SIĘ WYRÓŻNIAMY?

- 1. UNIWERSALNY model obsługi zrównoległe
- 2. KERNELS na rynku dostępnych rozwiązań

III

JAK DZIAŁA KERNELS?

WIELOAGENTOWY SYSTEM, WSPIERANY UCZENIEM MASZYNOWYM, czyli opis podprogramów i realacji pomiędzy nimi w KERNELS

IV

KERNELS W PRAKTYCE

- 1. Optymalizacja pracy klastra
- 2. Szukanie korelacji liniowej temperatury

V

WSPÓŁPRACA Z KERNELS

Możliwe formy współpracy oferowane przez KERNELS oraz sposób naliczania opłat

VI

PRZYKŁADOWE PAKIETY

Przedstawienie jak budujemy i estymujemy koszty naszych usług

VII

PRZYCZYNY POWSTANIA KERNELS

DROGA OD POMYSŁU DO PROGRAMU NA KLASTRZE, MOŻE BYĆ KRÓTKA

Najważniejsze powinna być **KONCEPCJA** algorytmu. Posiadanie jej często okazuje się być jednak dopiero pierwszym krokiem. Programista po długiej konfiguracji klastra dostaje zbiór realizujących się jednocześnie rozłożonych po maszynach instancji. Właściwy kod to przełożenie czegoś tak abstrakcyjnego jak pomysł na oprogramowanie każdej z nich oraz określenie jak mają się komunikować podczas obliczenia.

KERNELS DAJE GOTOWĄ OBSŁUGĘ praktycznie **KAŻDEGO ALGORYTMU** równoległego. Pisząc w nim programy implementujesz tylko i wyłącznie logikę ich działania, a nasz system przekłada ją za ciebie na pracę klastra.

PRZEGLĄD ZAGADNIEŃ Z ZAKRESU WDRAŻANIA ROZWIĄZAŃ TYPU MACHINE LEARNING I ARTIFICIAL INTELLIGENCE

PODSTAWOWE PROBLEMY przy algorytmach uczących przeznaczonych do złożonej analizy danych:

- Duża skala obliczeń, wymaga znaczących zasobów sprzętowych. Wymusza to rozproszoną infrastrukturę składającą się w wielu rozłącznych jednostek (KLASTER). Oznacza też konieczność zapewnienia ich wzajemnej komunikacji oraz implementacji procesów zarządzających w ramach każdego nowego programu.
- Utrzymanie stabilności. Cena mocy obliczeniowej implikuje, że awaria na jakimkolwiek etapie przebiegu algorytmu generuje znaczące koszty finansowe i opóźnienia.
- Probabilistyczny charakter przebiegu obliczeń, skutkuje koniecznością dynamicznego dostosowania liczby wykonywanych jednocześnie procesów do dostępnych zasobów sprzętowych.

OBECNIE DOSTĘPNE ROZWIĄZANIA realizujące algorytmy równoległe na klastrach:

- Gotowe implementacje zapewniające zazwyczaj obsługę jednego specyficznego rodzaju algorytmu równoległego. Wrzuca się do pudełka dane wejściowe i dostaje się wynik.
- Oprogramowanie integrujące ustaloną liczbę maszyn w centralnie zarządzany klaster oraz zestaw narzędzi umożliwiający samodzielne napisanie komunikacji i zarządzania. Właściwy realizujący obliczenie kod wymaga pracochłonnego oprogramowania pod niego tych procedur.

BUDOWA SWOJEGO ROZWIĄZANIA w standardowych klastrach:

- Opracowanie struktury jego przebiegu (KONCEPCJA).
- **INSTALACJA i INTEGRACJA** narzędzi dostępnych w ramach oferowanych na rynku oprogramowań klastrów.
- **IMPLEMNTACJA**. Konieczność opracowania zarówno logiki samego programu jak i procedur zarządzających obliczeniem w ramach zaplanowanych z zasobów klastra.



I. Wprowadzenie WYKORZYSTANIE TEORII EPIDEMII

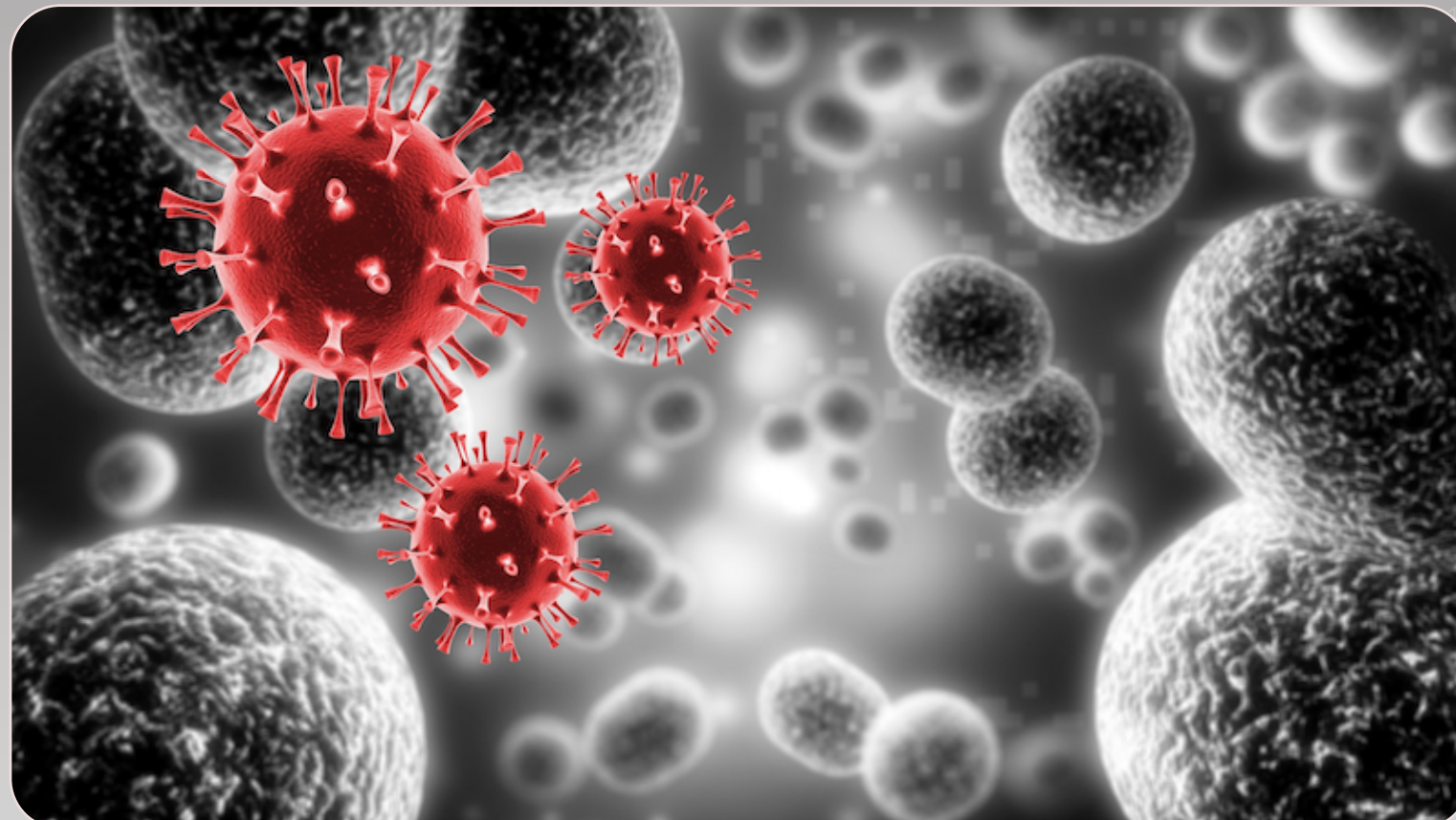
WIRUS NIE POTRZEBUJE ZARZĄDZANIA

Traktując klaster jako zmieniającą się dynamicznie, rozłożoną po jego rozłącznych jednostkach populację, a **KOD** użytkownika jako **WIRUS**, KERNELS przestaje potrzebować centralnego zarządzania. To zastosowane autorskie **UCZENIE MASZYNOWE**, wspomagając lokalne decyzje członków tego kolektywu, gwarantuje skuteczną i szybką ekspansję epidemii (obliczenia).

Pochodną podejścia jest naturalna zdecentralizowana obsługa dowolnego algorytmu równoległego. Realizuje ona za programistę **ROZPROSZENIE OBLICZEŃ** po klastrze. Zapewnia również **STABILNOŚĆ PRACY**, dzięki automatycznemu wykrywaniu uszkodzonych instancji i przenoszeniu prac na inne działające.

PROCES EPIDEMII W UJĘCIU OBLICZEŃ ROZPROSZONYCH

- **MATERIAŁ GENETYCZNY** (kwas nukleinowy) - Organiczny związek chemiczny zbudowany z nukleotydów. Znane są dwa rodzaje naturalnych kwasów nukleinowych: kwas deoksyrybonukleinowy (DNA) i rybonukleinowy (RNA). Przechowuje on zbiór instrukcji dla komórek zwany informacją genetyczną.
- **INFORMACJA GENETYCZNA** - Informacja zawarta w sekwencji DNA lub w przypadku niektórych wirusów RNA, która jest przekazywana organizmom potomnym. Można ją traktować jako zestaw zakodowanych instrukcji określających co komórki organizmu mają wykonać.
- **WIRUS** - Jedna z najbardziej pierwotnych struktur ewolucyjnych. Składa się z płaszcza białkowego kapsydu wewnątrz którego zawarty jest kwas nukleinowy (instrukcja dla komórki).
- **REPLIKACJA** - Białka kapsydu pośredniczą w przyłączaniu się do odpowiedniego receptora na powierzchni komórki swojego gospodarza (nosiciela wirusa), co pozwala im dostać się do jej środka. Następnie przy pomocy maszynerii komórkowej zmuszają ją do wykonania swoich kopii zgodnie z instrukcją zawartą w kwasie nukleinowym (polecenie budowy kopii wirusa).
- **EPIDEMIA** - Proces rozprzestrzenienia się wirusa pomiędzy jego potencjalnymi nosicielami. Po dostaniu się do organizmu kolejnego gospodarza wirus wymusza na jego zdrowych komórkach replikację swoich kopii, które zostają przekazane dalej, kolejnym członkom populacji, podczas ich wzajemnych kontaktów.
- **KOD, JAK WIRUS**, jest rozprowadzany przez zainfekowanie się nim członka modelowanej populacji. Zamiast budowy swojej kopii, żąda przeprowadzenia obliczenia użytkownika. Dostępna programiście komenda równoległego wykonania zadań oznacza, możliwość zarażenia się nimi od realizującej to polecenie instancji KERNELS. Odbywa się to podczas wspomaganych uczeniem maszynowym kontaktów potencjalnych nosicieli. (Szczegóły w rozdziale *Jak działa KERNELS?*)



II. Dlaczego KERNELS? ZALETY PRACY Z KERNELS

OSZCZĘDNOŚĆ CZASU, ZASOBÓW I FUNDUSZY

Dostarczamy produkt realnie **PRZYSPIESZAJĄCY PROCEDURY IMPLEMENTACJI** złożonych obliczeń równoległych, których skala wymaga użycia środowiska klastrowego.

Pozwala on programiście skupić się na najważniejszej rzeczy, jaką jest zaprojektowanie samej logiki działania algorytmu. Całą pracę związaną z **ZARZĄDZENIEM ZASOBAMI I KOMUNIKACJĄ** pomiędzy nimi w ramach obliczenia **WYKONA** za niego KERNELS.



CO WNIESIE KERNELS DO TWOJEJ FIRMY?

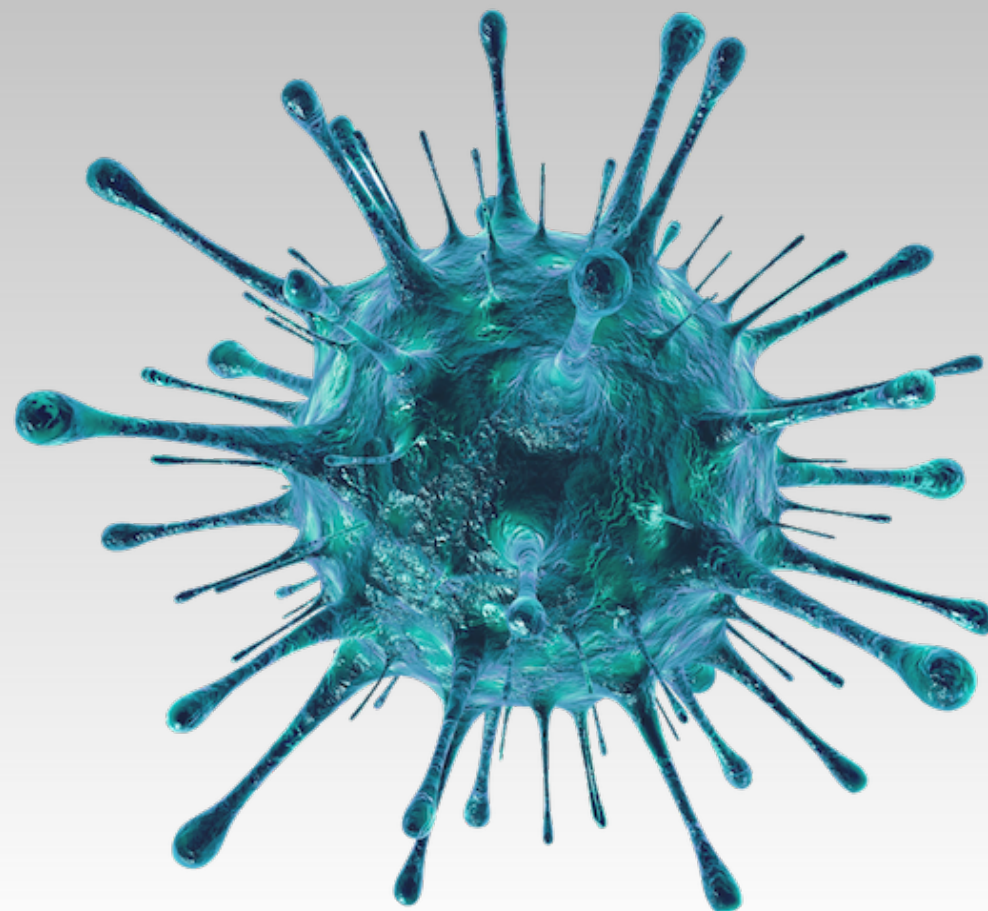
- **OBLICZENIA WYMAGAJĄ TYLKO NAPISANIA LOGIKI ICH DZIAŁANIA.** Dodatkowe i czasochłonne kodowania obsługi ich komunikacji w ramach klastra, ani sterowanie zasobami pod obliczenie, nie są potrzebne. Prace administracyjne wykona KERNELS. Znacząco skraca to czas uruchomienia rozwiązania i pozwala na uzyskanie pożądanego efektu angażując mniejszy zespół niż dotychczas.
- **ŁATWOŚĆ PISANIA NOWYCH ALGORYTMÓW.** Procedury zrównoleglenia kodu, reakcje na zwrot jego wyników, jak i współdzielenie danych są zaprojektowane tak, aby były jak najbardziej intuicyjne. Pozwala to na ich szerszą implementację, nie tylko przez personel programistyczny wyższego poziomu.
- **MOŻLIWOŚĆ WDROŻENIA NIEDOSTĘPNYCH WCZEŚNIEJ ROZWIĄZAŃ.** Złożone analizy danych wymagają potężnych zasobów obliczeniowych, co wymusza rozproszoną infrastrukturę sprzętową. Zdecydowana większość dostępnych na rynku rozwiązań ogranicza się tylko do specyficznych klas problemów. W przypadku KERNELS jest inaczej. Nasz system obsługuje praktycznie każde obliczenie, które wykorzystuje zrównoleglenia zadań.
- **OPTYMALIZACJA UŻYCIA ZASOBÓW SPRZĘTOWYCH.** Obecnie ich dzierżawienie lub utrzymywanie jest kosztowne. Dlatego tak ważnym jest, aby serwer był wykorzystany podczas obliczeń w jak największym stopniu. Bezczynna infrastruktura generuje niepotrzebne koszty. Wyposażyliśmy KERNELS w autorskie, oparte o maszynowe uczenie, oprogramowanie. Optymalizuje ono procedury administracyjne oraz sprawia, że właściwe algorytmy użytkownika maksymalnie wykorzystują klaster obliczeniowy. (Szczegóły w rozdziale *Jak działa KERNELS?*, LOCAL DECISIONS SUPPORT)
- **WIZUALIZACJE.** Zwykle końcowym odbiorcą, a także osobą weryfikującą postępy pracy jest człowiek. Aby podjąć racjonalne decyzje, bardziej potrzebuje interpretacji graficznych, niż surowych danych. KERNELS posiada bogatą gamę wykresów pozwalających prezentować w interfejsie zarówno końcowe jak i bieżące dane z pracy algorytmów.

II. Dlaczego KERNELS? TEORIA EPIDEMII, A TWOJE OBLICZENIA

WIĘCEJ, SZYBCIEJ, STABILNIEJ I TANIEJ

Użycie epidemii pozwala na **DECENTRALIZACJĘ** nie tylko zarządzania klastrem, ale także algorytmem użytkownika. Rozwiązuje to kluczowe problemy programowania rozproszonego, takie jak tworzenie się kolejek. Umożliwia **DYNAMICZNE DOŁĄCZANIE** do realizującego się już kodu dowolnych dodatkowych **ZASOBÓW SPRZĘTOWYCH**. Przede wszystkim gwarantuje **STABILNOŚĆ** wykonania Twoich prac.

Co najważniejsze stosowanie **KERNELS** **POSZERZY** **SPEKTRUM** możliwych do przeprowadzania obliczeń o te, których wdrożenie było wcześniej czasowo nierealne.



ZALETY ZASTOSOWANIA WSPOMAGANEJ UCZENIEM MASZYNOWYM EPIDEMII W PROGRAMOWANIU ROZPROSZONYM

- **BRAK CENTRALNEJ JEDNOSTKI STERUJĄCEJ** zarówno na poziomie zarządzania samym obliczeniem jak i całą realizującą je infrastrukturą serwerową. W istotny sposób zmniejsza to ryzyko tworzenia się kolejek, będących typowym problemem w modelu klient-serwer. Pozwala na przetwarzanie znacznie większych i wydajniejszych algorytmów, niż było to możliwe we wspomnianym klasycznym modelu. Niweluje również potencjalne problemy związane z dużą wielkością klastra obliczeniowego.
- **BARDZO SZYBKIE ROZPOCZĘCIE WYKONYWANIA WYSTAWIONYCH ZADAŃ**, pomimo braku centralnej jednostki zarządzającej. Efekt ten jest uzyskany dzięki zamianie zarządzania klient-serwer na statystyczne metody uczenia maszynowego wspomagające wynikającą z teorii epidemii ekspansję kodu programu po maszynach klastra.
- **STABILNOŚĆ I NIEZAWODNOŚĆ TWOICH OBLICZEŃ**. Uszkodzenie pojedynczego komponentu maszyny powoduje jedynie awarię operacji, które są na niej przetwarzane, a nie zatrzymanie całego systemu prowadzącego obliczenie. Co więcej **KERNELS** jest w stanie lokalnie wykryć przerwanie i przerzucić wykonywane prace na działające prawidłowo zasoby klastra. Taki problem nie będzie skutkować zatrzymaniem wykonywania Twojego algorytmu.
- **NIEOGRANICZONA SKALOWALNOŚĆ ORAZ PROSTE WDROŻENIE PRODUKCYJNE**. **KERNELS** pozwala na dynamiczne przydzielanie dodatkowej mocy obliczeniowej do realizowanych zadań, bez przerywania ich przetwarzania. Sam proces dodania nowych zasobów jest szybki i prosty. Przypomina czynności w super-komputerze, ale bez stromej krzywej uczenia się ich obsługi w początkowym etapie.
- **OBSŁUGA UNIWERSALNEGO MODELU OBLICZEŃ RÓWNOLEGŁYCH**. Z punktu widzenia implementacji, wykonujące się jednocześnie instancje kodu mogą powoływać kolejne zrównoleglenia, co pozwala na budowę algorytmów mających strukturę drzewa (rozdział *Uniwersalny model obsługi zrównoleglenia*).

II. Dlaczego KERNELS? BEZPIECZEŃSTWO ROZWIĄZYWANIA

• OCHRONA KAŻDEJ CZYNNOŚCI W KERNELS

Rozwój technologii ułatwia życie i otwiera niedostępne wcześniej możliwości. Wraz z nim pojawiają się jednak ogromne zagrożenia. Jednym z nich jest uzyskanie nieautoryzowanego dostępu do narzędzi, zasobów i wyników Twojej pracy.

To dzięki **NAJNOWOCZEŚNIEJSZYM STANDARDOM KRYPTOGRAFII**, a także **WYSOKIEMU POZIOMOWI BEZPIECZEŃSTWA** w trakcie **KOMUNIKACJI** pomiędzy komponentami, KERNELS pozwala Ci na przetwarzanie nawet bardzo wrażliwych i poufnych danych.



MECHANIZMY BEZPIECZEŃSTWA TELEINFORMATYCZNEGO KERNELS

- **BEZPIECZNA SZYFROWANA KOMUNIKACJA.** Wymiana informacji pomiędzy komponentami systemu wykorzystuje nowoczesną i potwierdzoną przez światową społeczność ekspercką kryptografię. Uwierzytelnienie komponentów systemu oparte jest o krzywe eliptyczne m.in Curve25519. Poufność przesyłanych danych zapewnia szyfrowanie ich uznanym i sprawdzonym standardem, algorytmem AES 256.
- **BEZPIECZEŃSTWO PRZETWARZANIA.** Poprawność bezpieczeństwa procesu wykonywania obliczenia jest zapewniona poprzez wykorzystanie polityk SELinux. W praktyce oznacza to, że każdy inicjowany przez KERNELS proces ma od samego początku ściśle ustalone uprawnienia dostępu, co oznacza, że nawet w przypadku uzyskania nieuprawnionego dostępu do niego możliwości uszkodzenia lub penetracji systemu jest infinitesimalna.
- **POUFNOŚĆ DANYCH W SPOCZYNKU.** Informacje w systemie są przechowywane na dyskach twardych zaszyfrowanych z wykorzystaniem bezpiecznego algorytmu AES 256.
- **ZAPOMNIJ O ZWYKŁYCH IDENTYFIKATORACH ORAZ DAJĄCYCH SIĘ ŁATWO WYKRAŚĆ HASŁACH,** jako podstawie potwierdzenia tożsamości użytkownika. Dostęp klientów do systemu wykorzystuje certyfikaty kryptograficzne X.509 uwierzytelnione podpisami cyfrowymi w oparciu o krzywą eliptyczną P-384 oraz funkcje skrótu SHA-384. W przeciwieństwie do haseł wprowadzanych przez użytkownika nie można ich łatwo przechwycić i wykorzystać.
- **MOŻLIWOŚĆ ROZBUDOWY POZIOMU ZABEZPIECZEŃ.** Dla wymagających jeszcze wyższego bezpieczeństwa klientów, istnieje możliwość przechowywania certyfikatów do systemu na zewnętrznych kluczach sprzętowych. Pozyskanie z nich certyfikatów przez atakującego jest praktycznie niemożliwe. Zalogowanie do KERNELS wymaga wtedy fizycznego posiadania klucza sprzętowego i wprowadzenia dodatkowego hasła. Zabezpieczenia takich kluczy są potwierdzone na poziomie EAL-6 Common Criteria.

II. Dlaczego KERNELS? ARCHITEKTURA SPRZĘTOWA

WYBIERZ NAJLEPSZE DLA SIEBIE ROZWIĄZANIE

Unikalna struktura programów składowych i relacji pomiędzy nimi powoduje, że KERNELS może pracować na szerokim spektrum konfiguracji sprzętowych. Instalacja jak również dodanie kolejnych maszyn są szybkie i bardzo proste.

NIE SĄ POTRZEBNE WŁASNE SERWERY. Uruchomienie odbywa się w ramach popularnych **CHMUR PUBLICZNYCH**. Jednak wdrożenie i późniejszą rozbudowę można również przeprowadzić na spełniającej minimalne wymagania prywatnej infrastrukturze wirtualizacyjnej (**ONPREMISE**).

INFRASTRUKTURA POZWALAJĄCA NA PRACĘ KERNELS

PODSTAWOWE KOMPONENTY, na których budowane jest KERNELS:

- **MASZYNA** (wirtualna lub fizyczna) - Pojedyncza samodzielna jednostka obliczeniowa odpowiadająca za prowadzenie operacji, a także komunikację ze swoimi odpowiednikami. Posiada własny **ODRĘBNY** system operacyjny i dostęp do przydzielonych zasobów.
- **WIRTUALIZACJA** - Dodanie warstwy abstrakcji pomiędzy platformą sprzętową, a systemem operacyjnym. Służy optymalizacji, umożliwiając współdzielenie **JEDNYCH** zasobów sprzętowych przez wiele maszyn (wtedy nazwanych **WIRTUALNYMI**).
- **KLASTER** (obliczeniowy) - grupa połączonych jednostek komputerowych, które współpracują ze sobą w celu udostępnienia zintegrowanego środowiska pracy.
- **CHMURA PUBLICZNA** - Zbiór maszyn wirtualnych pracujących na infrastrukturze sprzętowej operatora. Klient ma do niej dostęp poprzez dedykowane ku temu interfejsy. W ramach jej zasobów istnieje możliwość uruchomienia **KLASTRA**.
- **SIEĆ MIĘDZYINSTANCYJNA** - Warstwa wymiany informacji w ramach dynamicznego klastra obliczeniowego. Ma postać dedykowanej podsieci pomiędzy komponentami systemu.

JAKIE ROZWIĄZANIE WYBRAĆ? Sugerujemy zwrócić uwagę na dwa czynniki:

- **ELASTYCZNOŚĆ** - Przewaga Chmury - Swoboda modyfikacji zasobów, płatność tylko za rzeczywiste wykorzystanie, bezpieczeństwo kontraktowe danych.
- **KONTROLA** - Przewaga rozwiązania OnPremise (Instalacja i praca na prywatnych serwerach) - Własność, ale i konieczność utrzymania całego stosu technologicznego.

WYMAGANIA INSTALACYJNE. KERNELS potrzebuje **TYLKO** spełnienia prostych warunków:

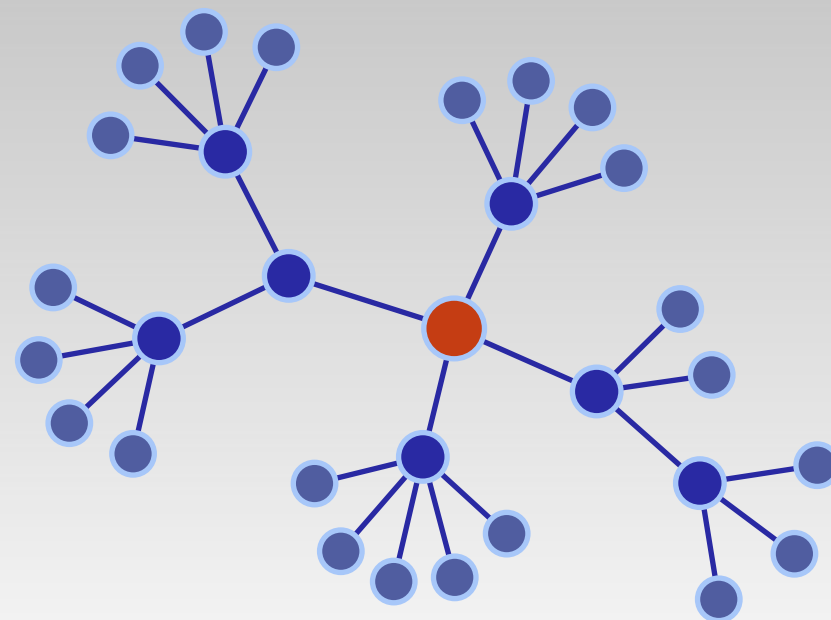
- **ŚRODOWISKO WIRTUALIZACJI** - Dysponowanie ustandaryzowanym środowiskiem uruchomienia maszyn wirtualnych (OnPremise lub Chmura).
- **ARCHITEKTURA** - Powszechnie używana platforma sprzętowa zgodna z x86_64.



MODEL, KTÓRY MOŻE PRAWIE WSZYSTKO

Obliczenia klastrowe nie kończą się na podzieleniu zagadnienia na podproblemy i złożeniu wyniku. Często zachodzi potrzeba, żeby dowolna z wykonujących się jednocześnie instancji kodu posiadała możliwość zrównoleglenia swoich prac. Takie algorytmy mają strukturę dynamicznie zmieniającego się drzewa. KERNELS pozwoli Ci w prosty sposób zrealizować DOWOLNE OBLICZENIE tego rodzaju.

Stan przejściowy obliczenia. Pokazuje węzły czekające na zwrot wyników.



DYNAMICZNE DRZEWO OBLICZEŃ I JEGO ZALETY

ZAŁOŻENIA MODELU

Podstawowym elementem jest węzeł, czyli instancja kodu mogąca powołać kolejne węzły i zareagować na zwrot ich wyników. Węzeł w trakcie swojej realizacji może

- Zażądać powołań dowolnej liczby węzłów (dzieci),
- Udostępnić dowolne dane wszystkim swoim realizującym się dzieciom.

Rodzic, czyli węzeł, który powołał dzieci, w momencie zwrotu dowolnego z nich może

- Zakończyć czekanie na zwrot pozostałych dzieci i przejść dalej,
- Wystawić żądanie powołania kolejnych dzieci i przejść w tryb czekania na zwrot
- Przejść w tryb czekania na zwrot kolejnych dzieci,
- Jeżeli był to zwrot ostatniego dziecka, a nie było decyzji co dalej system uznaje, że rodzic zakończył realizację zrównoleglenia i przechodzi dalej.

Zakończenie całego obliczania jest równoważne z zakończeniem kodu pierwszego węzła.

DLACZEGO TEN MODEL JEST TAK WAŻNY?

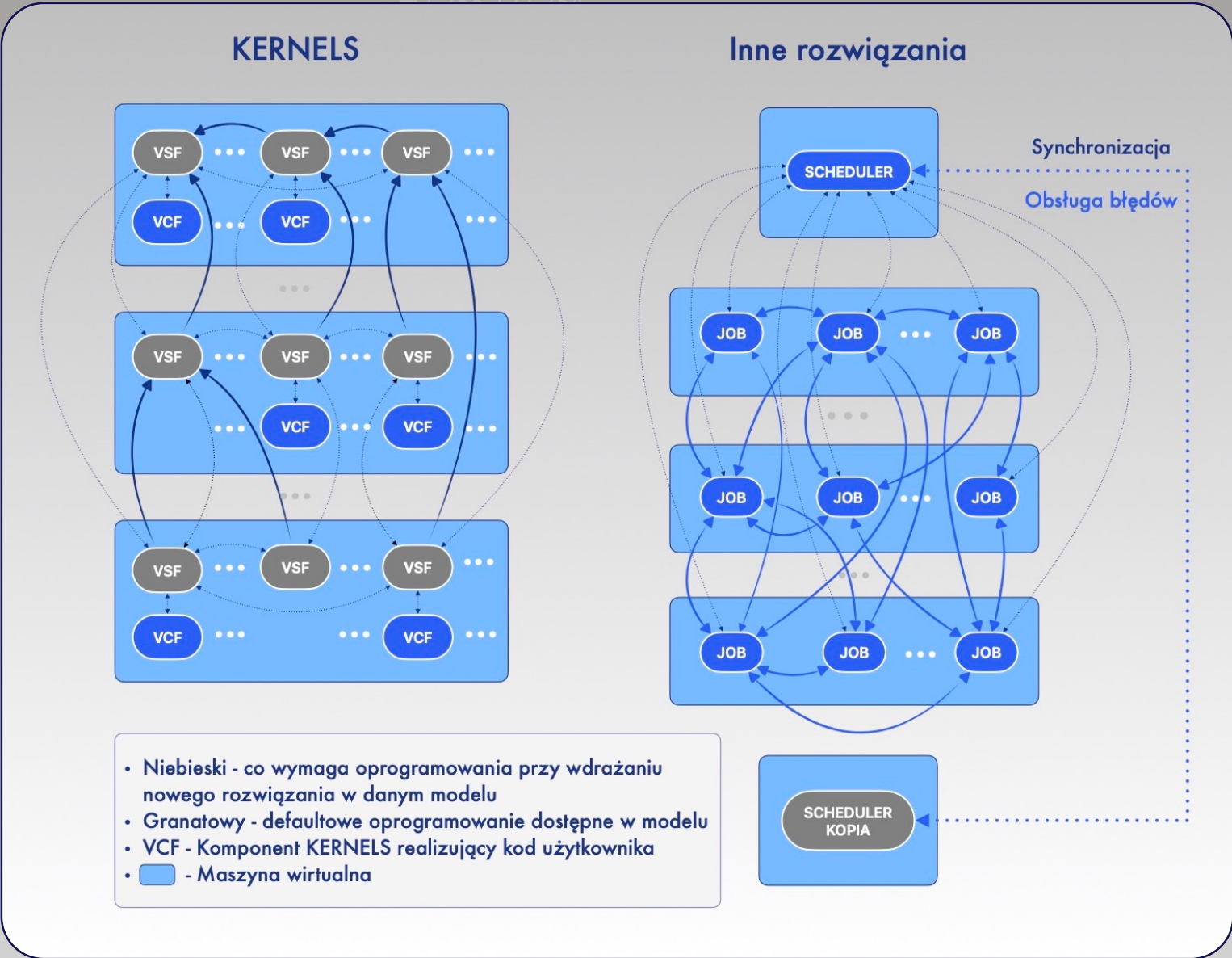
- Jest odpowiedzią na bieżące problemy związane z kierunkiem rozwoju technologii. Gigantyczne rozmiary danych, z których mają być wyciągane wnioski, uniemożliwiają analizę na podstawie całej posiadanej wiedzy. Stąd rozwiązania idą w kierunku inteligentnego próbkowania. Dynamiczne drzewo obliczeń opiera się o rekurencyjne powoływanie kolejnych węzłów. Umożliwia to efektywniejszą eksplorację danych, poprzez uzależnienie liczby zrównoleglenia (np. wspomniane próbkowanie) od efektów jakie uzyskuje się na danym obszarze badanych zasobów.
- W innych modelach, takich jak sieci neuronowe, widać tylko efekt końcowy nauki, a w modelu KERNELS sama struktura jego przebiegu pokazuje jak program po kolei dochodził do finalnych wniosków.
- Jest zupełny. Oznacza to, że realizuje każdy model obliczeń korzystających ze zrównoleglenia, a co za tym idzie da się w nim zaimplementować praktycznie wszystko.

III. Czym się wyróżniamy?

KERNELS NA RYNKU DOSTĘPNYCH ROZWIĄZAŃ

OBLICZENIA NA KLASTRACH MOGĄ BYĆ PROSTE I EFEKTYWNE

Programowanie rozproszone nie musi różnić się od pisania obliczeń na pojedynczej maszynie. Znajomość obsługi sprzętowej klastra i konieczność pisania warstwy komunikacyjnej dla każdego nowego algorytmu przestaje być przeszkodą. Z PUNKTU WIDZENIA PROGRAMISTY KERNELS JEST JEDNYM URZĄDZENIEM. Jedyne co musisz zrobić, to nie dotykając części serwerowej powoływać węzły i określić co zrobić po zwrocie ich wyników.



KERNELS	Inne rozwiązania
Zbiór zdecentralizowanych przemieszczających się po maszynach procesów zamiast zarządzania. Ich ruch i właściwości rozładowują typowe przy obliczeniach rozproszonych kolejki, niwelują konieczność utrzymania drzewa obliczenia, rozprawdzają kod użytkownika po klastrze jak wirusa oraz zapewniają ochronę przed jego awarią.	Po konfiguracji dostaje się maszyny (Nodes) oraz Scheduler, który może zainicjować na każdej z nich z góry ustaloną liczbę działających równolegle instancji. Implementacja komunikacji między nimi i reakcji na nią, jest właściwym obliczeniem. Jednak ochrona i optymalizacje tyczą utrzymania struktury klastra, a NIE kodu użytkownika.
Wdrożenie programu obliczeniowego sprowadza się do implementacji węzłów. Posiadają one obsługę powołania dowolnej liczby dzieci, przekazania wyniku rodzicowi oraz przyjęcia zwrotu dziecka, a po nim opcję zakończenia całego węzła wraz z podgałęziami.	Konieczne jest niskopoziomowe oprogramowanie, mogącej się wykonywać jednocześnie, statycznej liczby instancji. NIE MA obsługi węzłów, czyli całej abstrakcyjnej struktury utrzymania dynamicznego drzewa obliczenia, powołania, zwrotu oraz zakończenia węzłów.
Liczba wykonywanych równolegle prac jest dynamicznie dostosowywana do wolnych zasobów obliczeniowych na danej maszynie, a nie ustalona z góry. Pozwala to KERNELS efektywniej wykorzystać potencjał klastra, co za tym idzie znacząco zredukować koszty prowadzenia na nim prac.	Jest założenie, że jeden rdzeń Nodesa może wykonywać jedną zleconą przez Scheduler instancję. Nie ma znaczenia, że w danym momencie realizacji programów użytkownika są wolne zasoby pozwalające na wykonywanie równolegle większej liczby podzadań. Część mocy klastra pozostaje potencjalnie bezczynna.
Dzięki decentralizacji nie jest potrzebna żadna konfiguracja pod nowe obliczenie. Nieważne czy KERNELS nie robi nic, czy wykonuje programy użytkownika, uruchomienie kolejnego sprowadza się do wprowadzenia jego kodu przez interfejs.	Przed uruchomieniem kodu obliczenia trzeba określić wiele parametrów konfiguracyjnych środowiska pracy. Przy zmianach prac klastra, czy dodaniu nowego zadania, zwykle konieczne jest przerwanie wszystkiego i ponowna konfiguracja.
Unikalna obsługa zrównoleglenia prac pozwala dynamicznie zwiększyć moc obliczeniową DOWOLNEGO programu już w trakcie jego realizacji. Po dołączeniu nowych maszyn, zaczną one po prostu wykonywać zlecone przez działające już programy podzadania.	Dynamiczne zwiększanie mocy obliczeniowej KONKRETNEGO programu wymaga podczas jego pisania zaplanowania i wprowadzenia procedur określających sposób w jaki nowe maszyny mają się włączyć do jego realizacji, co często okazuje się być nietrywialnym zagadnieniem.

LOCAL DECISIONS SUPPORT (LDS)

Nasze AUTORSKIE OPROGRAMOWANIE (MACHINE LEARNING). Dzięki niemu możliwa jest błyskawiczna dystrybucja kodu programów po całym klastrze. Rozmieszczone na wszystkich rozłącznych maszynach LDSy stanowią rodzaj zdecentralizowanej statystycznej wiedzy klastra. Każdy LDS:

- Uczy się aktualnie wykonywanych przez swoją maszynę czynności oraz rozpoznawania pojawiających się w jej pracy anomalii,
- Na podstawie informacji dostarczonych przez przemieszczające się po całym klastrze VSFy analizuje stan zewnętrznych maszyn.

Celem powyższych czynności są:

- Estymowanie skąd VSFy mogą pobrać zadania obliczeniowe,
- Dynamiczne dostosowanie rozmiaru populacji VSFów (nosicieli kodu) oraz liczby wykonywanych jednocześnie VCF (węzłów obliczeń),
- Reagowanie na możliwe pojawienie się przeciążenia maszyny, poprzez zbiór zaprojektowanych ku temu procedur odciążenia RAMu i CPU,
- Wykrycie awarii zewnętrznych maszyn i wyłączenie ich z prac klastra.

VIRTUAL SEARCH FORK (VSF)

KERNELS jest WIELOAGENTOWYM OPROGRAMOWANIEM KLASTRA. Rolę agentów pełnią w nim podprogramy VSF. Ich populacja, rozprawdza jak wirusa, napisany w języku węzłów kod programu użytkownika. W zależności od cyklu życia VSF:

- Przemieszcza się po własnej (INNER) oraz rozłącznych (OUTER) maszynach klastra, gdzie komunikując się z innymi VSFami, próbuje zarazić się od nich zadaniem obliczeniowym,
- Po zainfekowaniu się zadaniem, VSF zgłasza do LDS żądanie budowy węzła obliczeniowego i wykonania kodu programu użytkownika przez VCF.
- Kiedy realizacji kodu węzła obliczania zawiera polecenie powołania dzieci, wystawia zadania, którymi mogą się zarazić inne VSFy,
- Pełni rolę przekaźnika między Interfejsem użytkownika, a KERNELS.

VIRTUAL COMPUTATION FORK (VCF)

Program realnie wykonujący kod węzła (napisany przez użytkownika). Liczba VCFów jest dynamicznie określana na podstawie uczenia maszynowego i metod statystycznych przez LDS. Metodyka ta pozwala w pełni wykorzystać zasoby obliczeniowe pojedynczej maszyny.

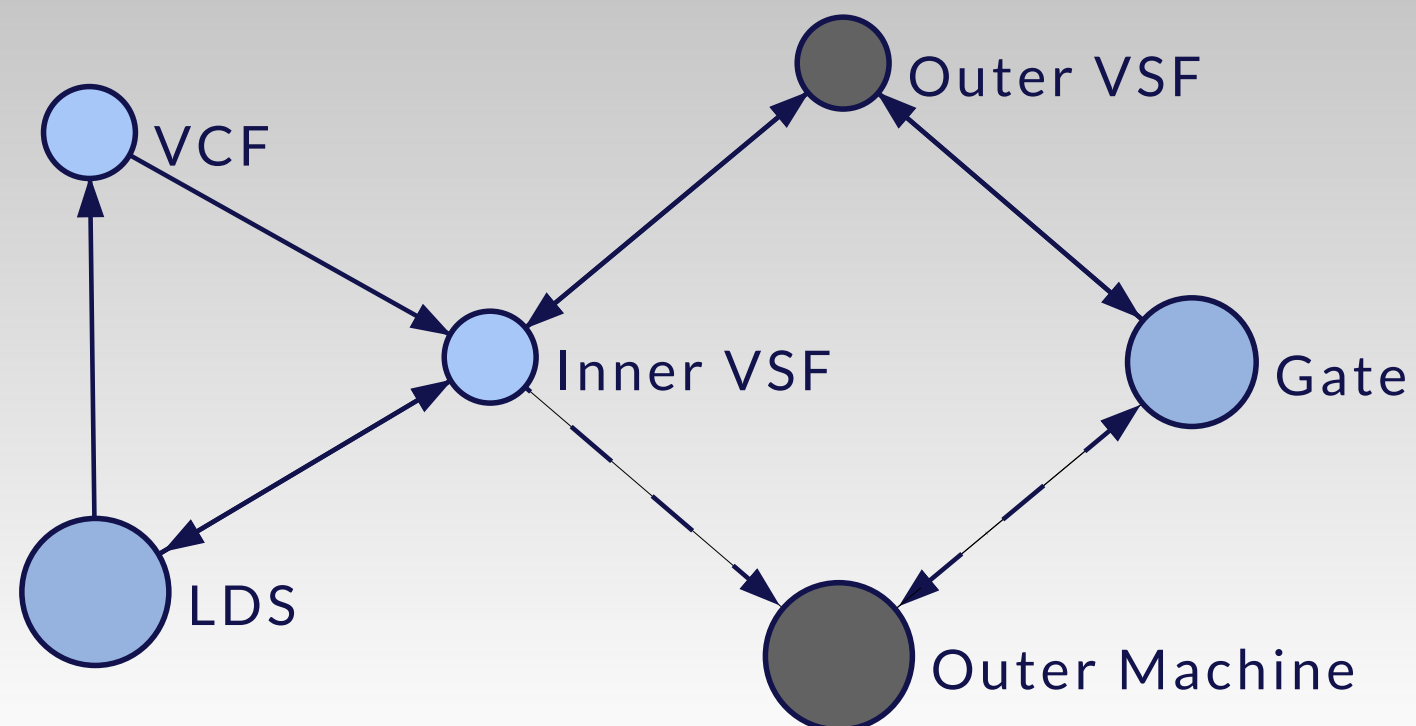
GATE

Program odpowiadający za transfer przychodzących z zewnętrznych jednostek VSFów oraz obsługę tych, które wracają do siebie. Ponadto prowadząc te prace zbiera dla LDS informacje o innych maszynach klastra.

OUTER MACHINE

Inna jednostka obliczeniowa pracująca w klastrze będąca kopią struktury przedstawionej na schemacie maszyny: LDS, VSF, VCF oraz Gate.

Relacje pomiędzy programami w KERNELS



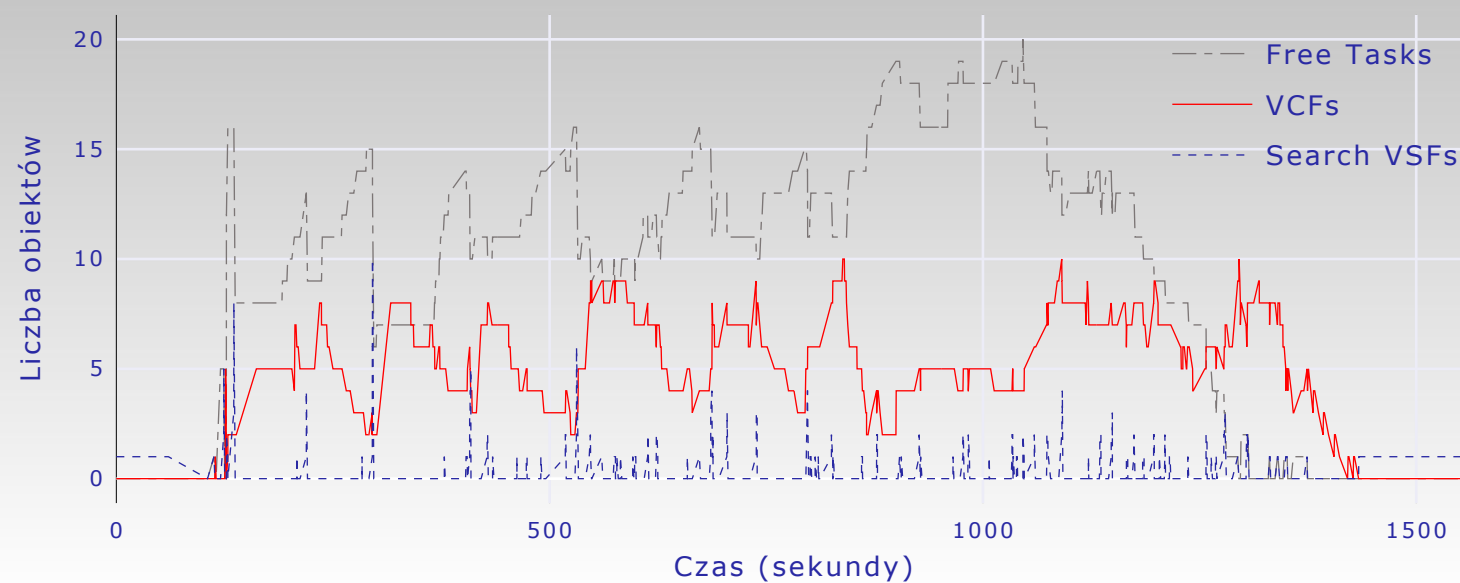
V. KERNELS w praktyce

OPTIMALIZACJA PRACY KLASTRA

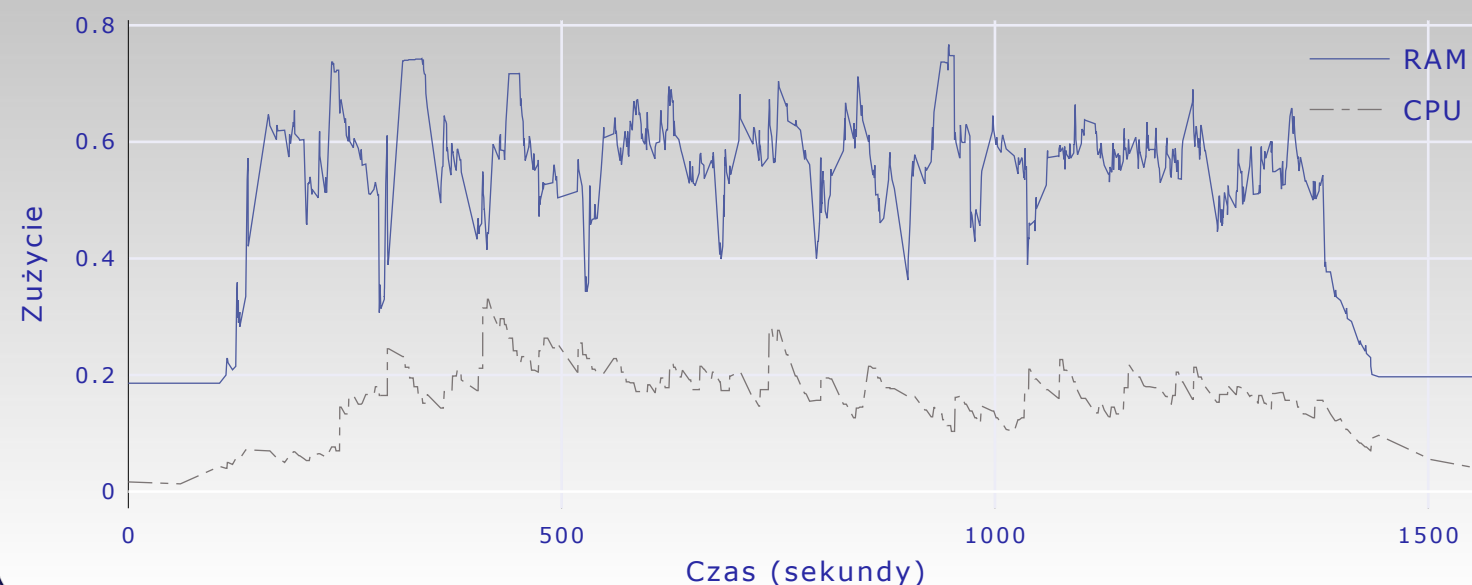
OD TEORII DO PRAKTYKI

Przykład ilustruje na wykresach zalety wspomagania prac klastra UCZENIEM MASZYNOWYM (ML). Demonstracja odbywa się na prostym kodzie, generującym duże fluktuacje zużycia RAMu. Ponadto wyjaśnia czemu ten pozornie banalny algorytm, bez zastosowania KERNELS może stać się prawdziwym wyzwaniem dla programisty.

UCZENIE MASZYNOWE (RYS. 1)



ZUŻYCIE RAM & CPU (RYS. 2)



WYDAJNE AI, CZYLI OBSŁUGA NIEDERMINISTYCZNEGO KODU

OPIS ALGORYTMU: Kod powołuje 5 węzłów. Każdy z nich tworzy losowo pomiędzy 20, a 30 kolejnych. Końcowe węzły zajmują RAM w jednym z rozmiarów 50MB, 120MB, 140MB, 460MB i utrzymują zajętość pamięci przez losowy trwający od 30 do 60 sekund czas, odpowiednio z prawdopodobieństwami 0.3, 0.3, 0.3, 0.1. Aby groziło to przeciążeniem, klastro składa się z pojedynczej maszyny, mającej 3GB RAM, a 1.5GB zajmuje system operacyjny.

PROBLEMY niedeterministycznych algorytmów występujące w przykładzie:

- Mogące przeciążyć klastro anomalie (nagłe drastyczne wzrosty RAMu).
- Przez większość przebiegu jednoczesne uruchomienie żądanych w danym momencie zadań (Free Tasks) przekracza założone możliwości sprzętowe.
- Przy tego typu kodzie złe zarządzanie pracą klastra wymusi przeszacowanie mocy obliczeniowej, zwiększając niepotrzebnie koszty finansowe.

Oparte o ML zdecentralizowane procedury działania KERNELS rozwiązują te trudności. Umożliwia to unikatowy proces adaptacji obsługi do charakterystyki bieżących prac.

UCZENIE MASZYNOWE (RYS. 1) dostosowuje ilości przetwarzanych równolegle węzłów oraz tempo dystrybucji kodu do aktualnie wolnych zasobów.

- Free Tasks - Obecnie wystawione przez węzły zadania, którymi nie zaraził się jeszcze żaden VSF.
- VCFs - liczba wykonywanych jednocześnie prac obliczeniowych. ML opiera jej zmiany o wykrywającą anomalie analizę statystyczną zużycia zasobów.
- Search VSFs - liczba agentów szukających Free Tasksów. Dzięki ML populacja VSFów dopasowuje się do możliwości bieżącej równoległej realizacji węzłów.

ZUŻYCIE RAM & CPU (RYS. 2), mimo przypadkowej alokacji pamięci i wystąpień outlierów (skoki RAM o 460MB), są utrzymane na bezpiecznym poziomie.

- RAM (Random-Access Memory) - Pamięć operacyjna. Jej przekroczenie może spowodować awarię przetwarzania zadania.
- CPU (Central Processing Unit) - Procesor obliczeniowy. W przykładzie ma 5 rdzeni. Przeciążony zwolni lub praktycznie zatrzyma wykonywanie rozkazów.

Wartości zmiennych są podane po ich standaryzacji do przedziału [0, 1].

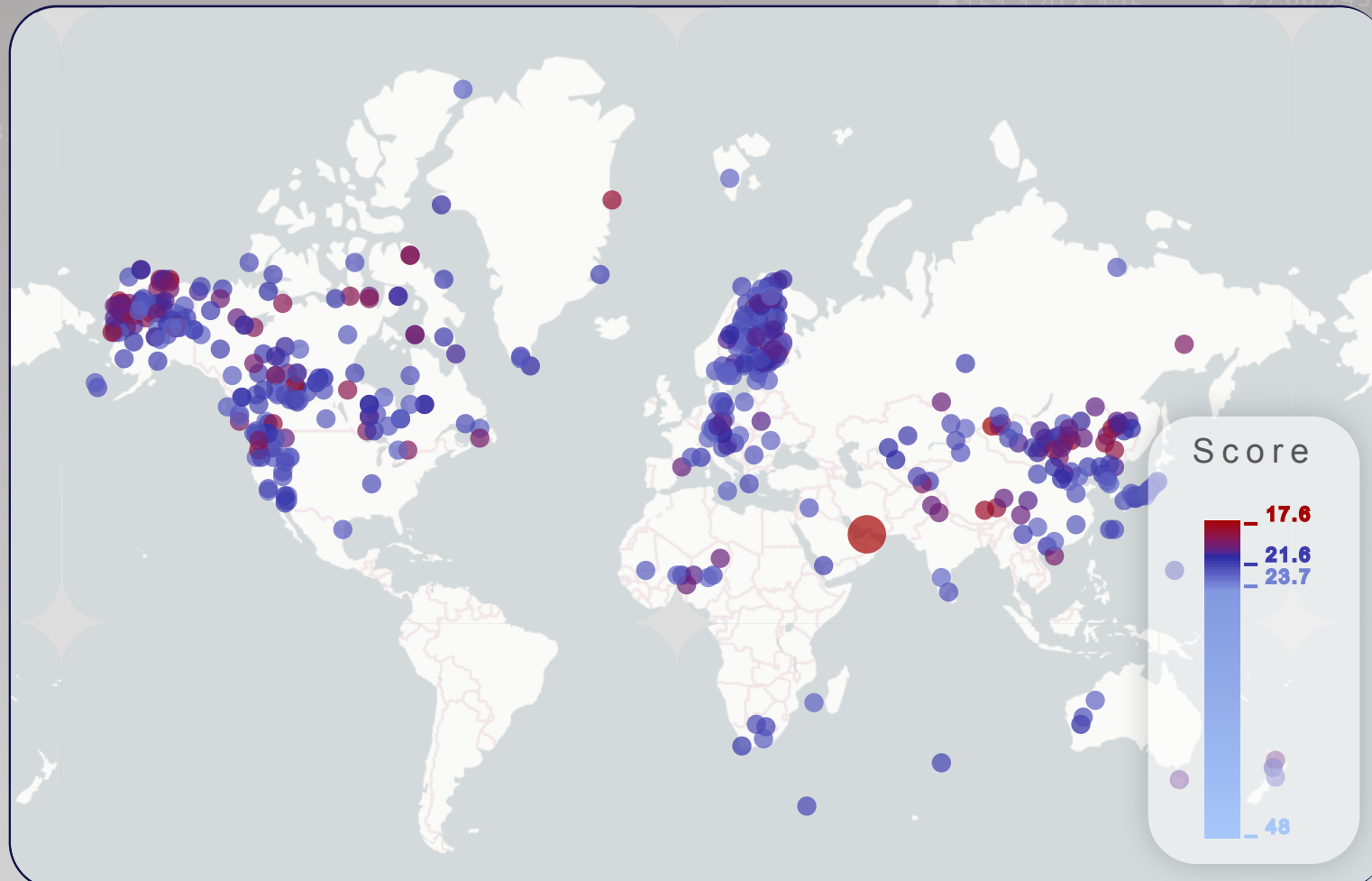
ZASTOSOWANIE KERNELS DO SZUKANIA POWIĄZAŃ POGODY

NOWE MOŻLIWOŚCI PROGRAMOWANIA

Istotą KERNELS jest dostarczenie programiście **OBSŁUGI** rekurencyjnego powoływania przez węzły jego obliczenia kolejnych podwęzłów oraz przyjmowania ich zwrotów.

Możliwość modelu przedstawiamy na podstawie algorytmu **INTELIGENTNEGO** przeszukania abstrakcyjnej przestrzeni korelacji liniowych pogody w celu znalezienia najlepszej estymacji temperatury dla wybranej stacji pomiarowej.

Mapa poniżej pokazuje obszary, które są powiązane regresją liniową z miejscem zaznaczonym na czerwono.



DYNAMICZNE DRZEWO OBLICZEŃ - PRZYKŁAD ZASTOSOWANIA

OPIS ALGORYTMU - poszukuje najlepszej funkcji liniowej zbudowanej z temperatur w różnych częściach świata, która przybliży temperaturę w zaznaczonej na czerwono stacji. Istotna jest metodyka jego działania. W przestrzeni wszystkich możliwych kombinacji, gdy któryś z testujących je węzłów dostaje obiecujące wyniki, powołuje podwęzły które intensyfikują przeszukanie tego obszaru (**REKURENCJA**). Oznacza to, że głębokość dynamicznego drzewa nie jest określona i może zmieniać w niedający się przewidzieć sposób.

SZCZEGÓŁY DZIAŁANIA - Program sprowadza się do dwóch rodzajów węzłów:

- Pierwszy węzeł obliczenia - Żąda wystawienia N. Celem jest rozproszenie obszaru przeszukania. Istotnie zwiększa to prawdopodobieństwo znalezienia się względnie blisko właściwego rozwiązania, co ułatwi następne kroki programu.
- Pozostałe węzły - Pobierają od rodzica parametry do metody pozwalającej na próbkowanie w okolicach najlepszego znanego swojemu bezpośredniemu przodkowi obszaru. Następnie przez kolejne przeszukania próbuje je poprawić.

W momencie zwrotu wyniku od dziecka, każdy węzeł, łącznie z pierwszym:

- Jeżeli wyniki znacząco się poprawiają, powołuje proporcjonalną do ich jakości liczbę dzieci. To jest właśnie **REKURENCJA**
- W przeciwnym przypadku kończy zagnieżdżenie.

Analizowane pomiary temperatury są pobierane z zewnętrznej bazy danych poprzez zapytanie SQL. Należy podkreślić, że KERNELS posiada obsługę, takich połączeń. Program trwa do momentu, kiedy nie wszystkie gałęzie się same nie skończą.

NOWE MOŻLIWOŚCI WYKORZYSTANIA KLASTRA - Obsługa węzłów jest elementem KERNELS. Wcześniej wymagała ogromnych nakładów pracy własnej. Pisząc oprogramowanie po prostu **TRZEBA SIĘ DO NIEJ ODWOŁAĆ**. Efektywność, jak i czas działania takich obliczeń jest wprost proporcjonalny do zasobów obliczeniowych. Dlatego w zależności od potrzeb można dostosować rozmiary klastra.

VI. Współpraca z KERNELS

MOŻLIWE FORMY WSPÓŁPRACY OFEROWANE PRZEZ KERNELS

REGUŁY TWORZENIA ABONAMENTÓW

Oferujemy Ci dostęp zawnasowanej platformy obliczeniowej, jaką jest KERNELS, przez interfejs. Standardowa oferta opiera się na pakietach. Poniżej przedstawiamy zasady ich budowy. W celu lepszej presonalizacji abonamtu do Twoich potrzeb, prosimy o kontakat.

OPŁATA STAŁA

Miesięczna opłata stała zależna od liczby maszyn zintegrowanych przez KERNELS w Twoim pakiecie.

ZASOBY OBLICZENIOWE

Miesięczna opłata za zużyte zasoby obliczeniowe i ich utrzymanie

KONSULTACJE

W razie problemów ze składnią lub w celu wyjaśnienia komplikacji związanych z asynchronicznością realizacji kodu, oferujemy konsultacje. Pakiety mogą zawierać pewną liczbę godzin takich konsutlacji. Po jej przekroczeniu pobierana jest stała opłata od każdej kolejnej godziny.

IMPLEMENTACJA (DEBUG) KODU

Oferujemy możliwość napisania części kodu Twojego algorytmu lub jeżeli masz gotowy kod, ale nie działa, możemy pomóc w jego debugowaniu. Przedstawiasz swój problem i w zależności od jego skali ustalamy godzinową stawkę oraz szacowany czas rozwiązania.

INDYWIDUALNE ROZWIĄZANIA

Określasz nam zadanie. My opracowujemy i implementujemy jego spersonalizowane pod Ciebie rozwiązanie w KERNELS.

1. Przedstawiasz problem i oczekiwania dotyczące rozwiązania,
2. Określamy kosztorys i plan wdrożenia,
3. Dostarczamy Ci gotowy program wraz z wizualizacjami, dostępny z poziomu interfejs Krenels.

Istnieje możliwość optymalizacji KERNELS pod Twój kod. Abonament tego nie oferuje. Jeżeli jesteś zainteresowany, prosimy o kontakat.

GŁÓWNE AKTYWNOŚCI

Przedstawienie celów działalności KERNELS.

- Udostępnienie interfejsu pozwalającego tworzyć algorytmy i wizualizacje na paltofrmie KERNELS.
- Wdrażaniem algorytmów dostarczonych przez naszych klientów.
- Budową zaawansowanych interaktywnych raportów z prac algorytmów.
- Projektowanie i implemantacja alorytmów ML i AI na potrzeby klienta.
- Pomocą naszym klientom w programowaniu ich rozwiązań.
- Prowadzenie szkoleń z pracy na KERNELS.



PRZEDSTAWIENIE JAK BUDUJEMY USŁUGI

DOBIERZ NAJLEPSZE ROZWIĄZANIE DLA SIEBIE

Przedstawione poniżej przykładowe pakiety mają na celu przybliżenie kosztów użytkowania KERNELS oraz pokazanie, że jest to narzędzie, którego cenę możesz dostosować do swoich potrzeb. Zaproponowane pakiety pozwalają Ci implementować znacznie szersze spektrum rozwiązań, niż było to możliwe do tej pory. Nasz zespół oferuje również opcję pełnego wdrożenia. Dostarczymy Ci gotowy program działający na naszej platformie.

PAKIET STARTOWY

Pozwala programować w KERNELS. Trzeba pamiętać, że oferowana w nim moc obliczeniowa jest mała, przez co czas wykonania bardziej złożonych algorytmów może być długi. Zalecamy ten pakiet podmiotom, które chcą zapoznać się z KERNELS i zobaczyć jak się na nim pracuje.

ABONAMENT:	Cena/Mies.
Procesor:	Nazwa
RAM:	Ilość RAM
Dysk:	Ilość pamięci
HARDWARE:	Cena
KONSULTACJE:	Cena/1h
IMPLEMENTACJA:	Cena-Cena/1h
SUMA:	Cena

PAKIET EKONOMICZNY

Jest wyposażony w potężniejszą, ale pojedynczą maszynę. Twoje algorytmy równoległe będą się w nim liczyły w zadowalającym tempie. Jeżeli Twoim celem nie jest prowadzenie skomplikowanych analiz, jest on dobrym balansem pomiędzy tym co chcesz uzyskać, a atrakcyjną ceną.

ABONAMENT:	Cena/Mies.
Procesor:	Nazwa
RAM:	Ilość RAM
Dysk:	Ilość pamięci
HARDWARE:	Cena
KONSULTACJE:	5h Free. Cena/1h
IMPLEMENTACJA:	Cena-Cena/1h
SUMA:	Cena

PAKIET KOMBINACJA MASZYN

Ten pakiet stanowi docelowe zastosowanie KERNELS, czyli systemu integrującego zbiór maszyn w jedno urządzenie wykonujące obliczenia AI i ML, jak jedna maszyna. W pakiecie przedstawiamy konfigurację złożoną z dwóch jednostek obliczeniowych, ale możliwości są dowolne. Taki rodzaj pakietów zalecamy do zaawansowanej analizy danych. Przedstaw nam swoje cele, a my pomożemy Ci dobrać właściwy zestaw maszyn.

ABONAMENT:	Cena/Mies.
Procesor:	Nazwa
RAM:	Ilość RAM
Dysk:	Ilość pamięci
HARDWARE:	Cena
KONSULTACJE:	20h za free, Cena/1h
IMPLEMENTACJA:	Cena-Cena/1h
SUMA:	Cena